



CONEXIDAD DIFUSA Y SEGMENTACIÓN DE IMÁGENES DIGITALES A PARTIR DE PUNTOS SEMILLA

TESIS:

PARA OBTENER EL TÍTULO DE
LICENCIADA EN MATEMÁTICAS APLICADAS

PRESENTA:

ROSA MARÍA GUTIÉRREZ APOLONIO

DIRECTOR DE TESIS:

M.C. ADOLFO MACEDA MÉNDEZ

HUAJUAPAN DE LEÓN, OAX., DICIEMBRE, 2016

*Dedicado a
mis padres queridos
Enrique Gutiérrez Morales
Rosa Apolonio Martínez
que me han dado su apoyo
incondicional.*

Agradecimientos

Quiero agradecer profundamente a Dios porque a pesar de los momentos difíciles, he encontrado la fortaleza en él, y me ha ayudado a culminar una etapa más en mi vida. También por darme la bendición de tener unos padres maravillosos, Enrique Gutiérrez Molaes y Rosa Apolonio Martínez, que sin su apoyo incondicional hubiera sido difícil terminar mis estudios. Porque han dado su vida entera, su amor, y han procurado la felicidad de sus hijos antes que la de ellos.

Mis más sinceros agradecimientos al M.C. Adolfo Maceda Méndez por su apoyo, motivación, orientación, supervisión y sobre todo su paciencia que me ha brindado. Así como a mis sinodales, al M.C. Juan Luis Hernández López, Dra. Silvia Reyes Mora, y al Dr. Jesús Fernando Tenorio Arvide, por su disposición en la revisión de este trabajo, y por sus observaciones para la mejora del mismo.

Por último agradezco a la Universidad Tecnológica de la Mixteca por haberme dado la oportunidad de realizar mis estudios en sus aulas y a mis profesores por el conocimiento y motivación brindada.

Índice general

Agradecimientos	II
Lista de figuras	V
1. Conjuntos difusos	3
1.1. Conjuntos difusos: conceptos básicos	3
1.2. Operaciones básicas de los conjuntos difusos	7
1.3. Clasificación de los conjuntos difusos	13
1.4. Relaciones difusas	14
2. Imágenes digitales	17
2.1. Tipos de imágenes	17
2.1.1. Imágenes analógicas	18
2.1.2. Imágenes digitales	18
2.2. Píxeles	22
2.3. Segmentación de imágenes digitales	28
2.3.1. Procesamiento de imágenes digitales	29
2.3.2. Concepto y aplicaciones de la segmentación	31
2.3.3. Clasificación de las técnicas de segmentación	32
2.3.4. Algoritmos de segmentación	33
3. Conexidad difusa y puntos semilla	38
3.1. K -conexidad y afinidad difusa	38
3.2. M -segmentación y M -semisegmentación	44
3.3. Grafo M -difuso	47
4. Algoritmo de segmentación con puntos semilla	56
4.1. Algoritmos voraces	56
4.2. Colas de prioridad	58
4.3. Algoritmo de Dijkstra	59
4.4. Algoritmo de segmentación	63
4.5. Resultados	69
4.5.1. Características de la máquina y ejecución del algoritmo	69
4.5.2. Figura geométrica	70
4.5.3. Fotografía	71
4.5.4. Imagen por satélite	74

4.5.5. Resonancia magnética del cerebro humano	78
--	----

Índice de figuras

1.1.	Gráfica de un subconjunto difuso asociado a la variable lingüística “números reales considerablemente mayor a 10”.	4
1.2.	Conjuntos difusos que expresan el concepto general de una clase de números reales que son cercanos a 2. Las curvas son continuas, simétricas y monótonas.	6
1.3.	Conjuntos difusos que representan al conjunto de las personas jóvenes, maduras y ancianas, respectivamente.	7
1.4.	Intersección estándar de los conjuntos difusos del Ejemplo 3.	10
1.5.	Complemento e intersección de los conjuntos difusos A_1 y A_3 del Ejemplo 3.	11
2.1.	Ejemplo de una imagen analógica.	19
2.2.	El ojo humano captura parte del paisaje. Imagen obtenida de http://www.el-nacional.com/ciencia_y_ambiente/ .	19
2.3.	Imagen digital en blanco y negro.	20
2.4.	Matriz correspondiente a la imagen de la Figura 2.3.	20
2.5.	Ejemplo de una imagen en tonos de gris con su respectiva representación matricial.	21
2.6.	Imagen de huevos, cuando se cuantifica a 8, 4, 2 y 1 bit(s). Imagen obtenida de [7].	22
2.7.	Imagen en el sistema RGB, donde la primera imagen es la original y las otras corresponden a la forma en que se ve con cada canal de color: verde, rojo y azul, respectivamente. Imagen obtenida de [7].	23
2.8.	Canales de color del sistema CMYK. Imagen obtenida de [18].	23
2.9.	Espacio de color HSV. Imagen obtenida de [18].	23
2.10.	Representación de $N_4(p)$, donde $p = (2, 2)$.	24
2.11.	Representación de $N_D(p)$, donde $p = (2, 2)$.	25
2.12.	Representación de $N_8(p)$, donde $p = (2, 2)$.	25
2.13.	Ejemplo de dos conjuntos U y V que 8-adyacentes.	26
2.14.	Representación de dos 8-caminos del píxel r al s .	27
2.15.	Representación de un 4-camino del píxel u al píxel v .	27
2.16.	Representación gráfica de los píxeles p y q tal que $D_4(p, q) \leq 2$, donde $p = (2, 2)$.	29
2.17.	Etapas del procesamiento de imágenes. Se puede observar que una de ellas es la segmentación de imágenes. Figura adaptada de [3].	30

2.18. Ejemplo del proceso de segmentación de imágenes.	32
3.1. Gráfica que representa el marco $V(b, d)$, donde $b = (1, 1)$ y $d = (3, 3)$	40
3.2. Representación de la imagen a) en el plano \mathbb{Z}^2	41
3.3. Representación gráfica de un escenario difuso sobre V con su respectivo subconjunto difuso f	41
3.4. Ejemplo de una cadena que comienza en el píxel c y termina con el píxel d	43
3.5. Gráfica que muestra algunas cadenas que se pueden formar desde el píxel c al d	43
3.6. Conjunto de puntos V para construir una M -segmentación.	46
3.7. Representación de un grafo M -difuso.	47
3.8. El conjunto V consta de 5 subconjuntos: T, L, B, R, O . Los tres puntos de la parte superior representan al conjunto T . El conjunto L consta de los 5 puntos horizontales que están a la izquierda en la gráfica. Los 3 puntos horizontales en la derecha representan el conjunto R . B consta de los tres puntos verticales que están debajo, y el punto que se encuentra aislado forma el subconjunto O	49
3.9. Representación de una σm -cadena.	50
3.10. Cada línea gris representa una σm -cadena que va de un píxel en $d \in S_m$ al píxel $c \in V$, para $1 \leq m \leq M$	50
3.11. Representación de $\mu_{\sigma, m, V, S_m}(c)$ en una imagen.	51
3.12. Representación gráfica del Teorema 4. Cada conexión a d representa una σm -cadena que comienza en un elemento de S_n , para $1 \leq n \leq M$. La conexión de d a c representa a $\psi_n(c, d)$	54
3.13. Conjunto $V = \{u, v, w\}$, donde $V_1 = \{v\}$ y ψ_1 está definida por la función dada en (3.28)	54
4.1. Representación gráfica de una cola de máxima prioridad a la que se le aplica la operación $\text{INSERT}(H, x)$	59
4.2. Representación gráfica de una cola de máxima prioridad a la que se le aplica la operación $\text{EXTRACT-MAX}(H)$	60
4.3. Representación de la técnica de relajación.	61
4.4. Aplicación del método de Dijkstra al grafo a) e inicialización de los vectores d y π . En b) el algoritmo se aplica al primer nodo, A . De c) a g) se muestran algunas ejecuciones del algoritmo (Ver la continuación en la Figura 4.5).	64
4.5. Aplicación del método de Dijkstra. En h) se puede observar el camino con costo mínimo de A a cualquier nodo (Ver la primera parte en la Figura 4.4).	65
4.6. Segmentación que resulta de la imagen (a). Las imágenes (b) y (c) son el resultado de la segmentación al utilizar una 4-adyacencia.	71
4.7. Las imágenes (b) y (c) son el resultado de la segmentación aplicado a la imagen (a) utilizando una 8-adyacencia.	72
4.8. Como resultado de la segmentación aplicado a la imagen (a) utilizando una 8-adyacencia se obtiene las imágenes (b) y (c).	73

4.9. Segmentación de la imagen (a) utilizando una 4-adyacencia. Se seleccionaron 3 objetos y el resultado se puede observar en las imágenes (b), (c) y (d). El tiempo de ejecución es de 5.032 segundos.	74
4.10. Se segmenta en 3 objetos a la imagen (a) obteniendo como resultado las imágenes de (b) a (d), en un tiempo de 7.292 segundos.	75
4.11. Imagen por satélite de la región de Hong Kong, la imagen se divide en 3 objetos y las imágenes (b), (c) y (d) son los resultados de la segmentación para una 4-adyacencia.	76
4.12. Segmentación que resulta de la imagen (a) al seleccionar 3 objetos con una 8-adyacencia. Se observa una mejor segmentación en comparación a una 4-adyacencia.	77
4.13. Segmentación de una imagen de resonancia magnética del cerebro tipo coronal utilizando una 4-adyacencia. La imagen se divide en 3 objetos, el cerebro, el cráneo y el fondo.	79
4.14. Imagen de resonancia magnética del cerebro tipo coronal, la imagen se segmentó en 3 objetos utilizando una 8-adyacencia.	80
4.15. Imagen de resonancia magnética corte axial que se segmentó en 3 objetos utilizando una 4-adyacencia.	81
4.16. Segmentación de una MRI corte axial utilizando una 8-adyacencia y seleccionando 3 objetos. Lo que se desea es distinguir el área del encéfalo.	82

Introducción

La segmentación de imágenes digitales es una parte crucial del Procesamiento Digital de Imágenes (PDI) debido a las dificultades que conlleva como la importancia de sus resultados. La segmentación de imágenes asigna una etiqueta a cada píxel de forma que los píxeles con la misma etiqueta comparten características similares. La segmentación ayuda a localizar objetos o sus límites dentro de la imagen. La dificultad está en cómo asignar la etiqueta a los píxeles y agruparlos por sus características ([3]).

La segmentación de imágenes es muy utilizada en diferentes áreas como en medicina, reconocimiento de patrones, en imágenes por satélite, entre otras. En medicina se aplica para identificar áreas anatómicas y poder realizar algún diagnóstico o tratamiento, como en vasos sanguíneos ([6]). También, se puede segmentar la materia blanca, la materia gris y el fluido cerebro-espinal de una imagen de resonancia magnética, que es de suma importancia en el análisis cuantitativo del volumen de diferentes partes del cerebro, con el propósito de evaluar el progreso o condición de diversas enfermedades como el Alzheimer y la epilepsia ([20]). En el reconocimiento de patrones es importante en la detección de huellas o rostros. La detección de huellas dactilares es muy utilizada en criminalística para identificar a una persona, también es usada por un usuario que desee acceder a su cuenta en alguna empresa ([19], [24]).

Los algoritmos de segmentación toman en cuenta similitudes como el color, la intensidad o la textura. Con el concepto de conexidad difusa, el algoritmo presentado en esta tesis segmenta a la imagen en *componentes conexas difusas*, debido a que las imágenes se pueden interpretar utilizando conceptos de matemática difusa ([16]).

Existen diferentes enfoques para atacar el problema de la segmentación. Se puede utilizar la detección de bordes, el crecimiento por regiones y algunos métodos bayesianos, entre otros ([17], [20]). Algo importante a destacar, es que no existe una segmentación universal para utilizar en cualquier aplicación, todo depende de lo que se desee reconocer en una imagen, es decir, de la aplicación ([12]).

Para etiquetar cada píxel se necesita analizar diferentes caminos entre ellos utilizando otros elementos de la imagen. Para lograr esto, tiene un papel importante el concepto de *afinidad*. Una afinidad valora la adyacencia entre dos píxeles y la similitud de sus intensidades. Con la función de afinidad se puede medir la fuerza de los caminos entre dos píxeles para determinar el grado de pertenencia de un píxel a un objeto. El problema principal es que se requiere analizar una gran cantidad de caminos para determinar dicho grado de pertenencia.

El objetivo general de esta tesis es presentar un algoritmo de segmentación de imágenes en escala de gris a partir de puntos semilla utilizando la teoría de conexidad difusa y el algoritmo de Dijkstra. Para esto se presenta el concepto de conjunto difuso y se discute la conveniencia de representar una imagen en tonos de gris por medio de un conjunto difuso. Esta relación se debe a que por naturaleza estas imágenes son de aspecto difuso, por la pérdida de información en la transformación de una imagen de color a una imagen en tonos de gris y de una escena en 3-D a una imagen en 2-D. El algoritmo de segmentación que se presenta en esta tesis se basa en el trabajo [16]. El aporte de esta tesis es exponer de forma más comprensible la teoría de dicho trabajo, y se dan algunos elementos que facilitan comprender la implementación de dicho algoritmo.

Las imágenes analizadas en la presente tesis son digitales en tono de gris, donde la tonalidad se representa en el rango $[0, 255]$. Es difícil que una persona logre distinguir los 256 tonos de gris, y el segmentar una imagen en 2 o 3 objetos ocasiona un problema al querer dividir el rango de tonalidades en esos objetos. Por esto, es importante el contexto difuso para facilitar el manejo de las inexactitudes en los datos, en lugar de una simple visualización. El algoritmo presentado requiere la intervención del usuario, el cual tiene que indicar algunos puntos de los objetos en los que desea segmentar la imagen. A los píxeles marcados que se encuentran en los objetos se les denomina *puntos semilla*.

La estructura del documento es la siguiente: En el Capítulo 1 se presenta el concepto de conjunto difuso, así como algunas de sus propiedades y operaciones básicas como unión e intersección. También se proporciona el concepto de una relación difusa y las propiedades de una relación de equivalencia difusa. En el Capítulo 2 se discute el tema de las imágenes digitales, haciendo énfasis en las imágenes en tonos de gris. También se describen brevemente las imágenes binarias y a color. Se mencionan las diferentes representaciones para una imagen digital y conceptos matemáticos de conexidad de píxeles. Por último se dedica una sección para el tema de segmentación, los diferentes métodos que existen y algunas aplicaciones en las que se ha utilizado.

En el Capítulo 3 se mencionan los conceptos de K -conexidad y de afinidad, para definir una M -segmentación, donde M es el número de objetos en los que se desea segmentar la imagen. Para lograr la M -segmentación se introduce el concepto de grafo M -difuso. El resultado principal de este capítulo es el Teorema 4 que garantiza la unicidad de una M -segmentación de una imagen en tono de gris a partir de afinidades construidas utilizando puntos semilla para cada uno de los objetos en los que se desea segmentar a la imagen. Para construir la M -segmentación mencionada en este teorema, en el Capítulo 4 se estudian los métodos voraces que permiten analizar de manera óptima los diferentes caminos entre dos píxeles de una imagen digital. Utilizando el algoritmo de Dijkstra se determina a partir de los puntos semilla el grado de pertenencia de cada píxel a los objetos en los que se desea segmentar a la imagen. El resultado es una M -segmentación que satisface las condiciones del Teorema 4. Además, se presentan algunas pruebas del empleo de este algoritmo en imágenes sintéticas, médicas y satelitales. Se muestran los conjuntos difusos en los que quedan segmentadas las imágenes y se proporciona el tiempo de ejecución en cada caso.

Capítulo 1

Conjuntos difusos

Existen diferentes problemas que contienen cierta ambigüedad en sus datos por lo que es difícil modelarlos con la matemática tradicional que conocemos, y cuando se hace de esta manera muchas veces no se obtienen buenos resultados. Por eso, han surgido diferentes metodologías para tratar de modelar estos problemas. En este capítulo se muestra una manera que se ha conseguido para modelar la ambigüedad de algunos problemas: los conjuntos difusos. Se da una introducción sobre las operaciones básicas que se pueden hacer con ellos. Estos conceptos son básicos para introducir la idea de conexidad difusa.

1.1. Conjuntos difusos: conceptos básicos

Hay términos que representan conceptos lingüísticos como frío, caliente y muy caliente, que generalmente son empleados para referirse a propiedades de objetos. A estas propiedades se les denomina *variables lingüísticas*. Las variables lingüísticas facilitan transiciones graduales entre estados, por lo que poseen una capacidad natural para expresar y tratar con observaciones y medidas de incertidumbre, a diferencia de las variables rígidas (variables tradicionales), que no tienen esta capacidad. La definición de estados mediante conjuntos rígidos es matemáticamente correcto, sin embargo no siempre es realista por los inevitables errores de medición. Por el contrario, las variables lingüísticas se apegan más a la realidad, ya que capturan la medida de la incertidumbre como parte de los datos experimentales ([13]). A estas variables se les puede asociar un conjunto difuso, de acuerdo a la siguiente definición.

Definición 1. Sea X un conjunto rígido. Un subconjunto difuso A de X es una función $A : X \rightarrow [0, 1]$. Dado $x \in X$, el valor $A(x)$ es el grado de pertenencia del elemento x al subconjunto difuso A . Si $A(x) = 0$, se dice que x no pertenece al subconjunto difuso A , mientras que si $A(x) = 1$, se dice a que x pertenece totalmente a A .

En la definición anterior entre mayor es el valor de $A(x)$ mayor es su grado de pertenencia. Para distinguir entre un conjunto difuso y un conjunto en el sentido normal,

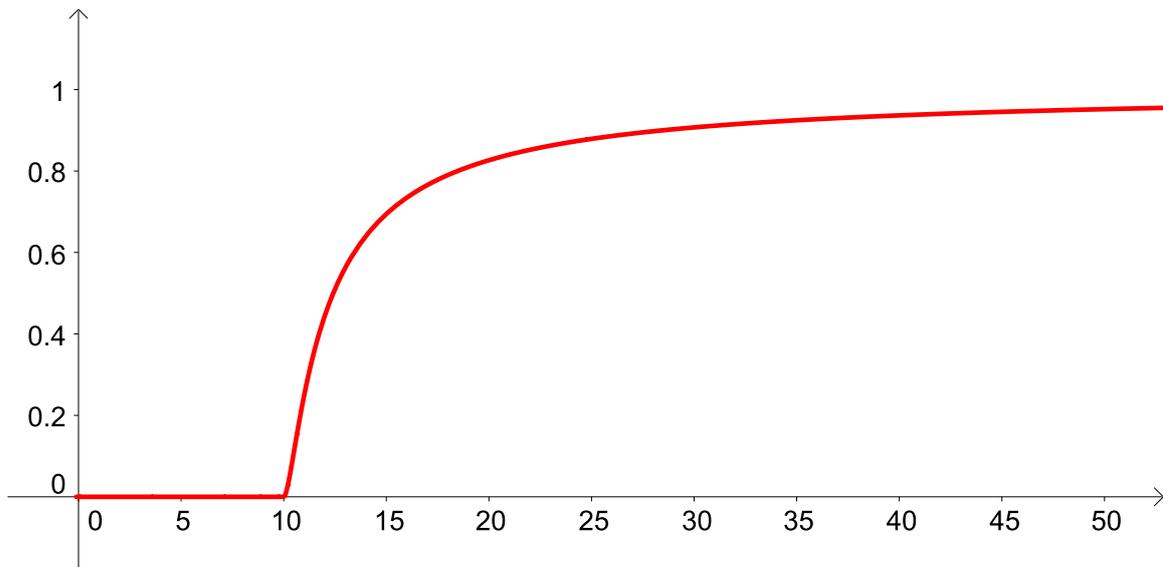


Figura 1.1: Gráfica de un subconjunto difuso asociado a la variable lingüística “números reales considerablemente mayor a 10”.

a estos últimos se les denomina en la literatura como conjuntos rígidos. Un subconjunto rígido E de X se puede identificar con un subconjunto difuso mediante su función característica. La función característica de E está definida como sigue:

$$\chi_E(x) = \begin{cases} 1, & \text{si } x \in E; \\ 0, & \text{si } x \notin E. \end{cases} \quad (1.1)$$

Como se puede observar, la función $\chi_E(x)$ es un caso particular de un conjunto difuso.

Ejemplo 1. Consideremos la variable lingüística "números reales considerablemente mayores que 10". A esta variable se le puede asignar el subconjunto difuso \bar{A} definido como:

$$\bar{A}(x) = \begin{cases} 0, & \text{si } x \leq 10; \\ \left(\frac{x-10}{x-9}\right)^2, & \text{si } x > 10. \end{cases} \quad (1.2)$$

La Figura 1.1 es la representación gráfica del conjunto difuso \bar{A} . Los números menores o iguales que 10 tienen grado de pertenencia igual a 0 porque no son “considerablemente más grandes que 10”. En cambio, entre más grande sea el valor de x , $\bar{A}(x)$ se acerca a 1. A la variable lingüística considerada se le pueden asociar otros conjuntos difusos, basta que para $x \leq 10$, cumplan que $A(x) = 0$ y para $x > 10$, $\bar{A}(x)$ se acerque a 1.

Una de las ventajas de un conjunto difuso es que permite representar conceptos ambiguos expresados en el lenguaje natural, pero esta representación no solo depende del concepto, sino también del contexto en el cual se esté mencionando. Por ejemplo, si

se habla de temperaturas altas se pueden tener varios contextos, ya sea del clima o de algún objeto, que pueden ser representados por conjuntos difusos totalmente diferentes. Incluso en contextos similares, los conjuntos difusos pueden variar considerablemente representando el mismo concepto ([4]). Sin embargo, pueden compartir atributos característicos como se puede observar en el Ejemplo 2.

Ejemplo 2. Sean \tilde{A}_i , para $i = 1, \dots, 4$, subconjuntos difusos de \mathbb{R} que representan el concepto de “números reales cercanos a 2”, definidos por las siguientes funciones:

$$\tilde{A}_1(x) = \begin{cases} (x - 2) + 1, & \text{si } x \in [1, 2]; \\ (2 - x) + 1, & \text{si } x \in [2, 3]; \\ 0, & \text{en otro caso.} \end{cases}$$

$$\tilde{A}_2(x) = \frac{1}{1 + 10(x - 2)^2},$$

$$\tilde{A}_3(x) = \exp(-|5(x - 2)|),$$

$$\tilde{A}_4(x) = \begin{cases} \frac{1 + \cos(2\pi(x - 2))}{2}, & \text{si } x \in [2 - \frac{1}{2}, 2 + \frac{1}{2}]; \\ 0, & \text{en otro caso.} \end{cases}$$

Estos conjuntos corresponden al mismo concepto lingüístico, pero su definición es muy distinta, como se puede observar en la Figura 1.2. Sin embargo, comparten ciertas características, porque para $i = 1, \dots, 4$:

- \tilde{A}_i es continua.
- $\tilde{A}_i(2) = 1$ y $\tilde{A}_i(x) < 1$ para todo $x \neq 2$.
- Entre mayor sea $|2 - x|$, menor es el valor de $\tilde{A}_i(x)$, es decir decrece monótonamente.
- $\tilde{A}_i(2 + x) = \tilde{A}_i(2 - x)$ para todo $x \in \mathbb{R}$, esto es, \tilde{A}_i es simétrico con respecto a $x = 2$.
- $\tilde{A}_i(x) \approx 0$ para todo $x \notin [1, 3]$.

En las aplicaciones, depende del contexto del problema en particular la forma adecuada del conjunto difuso. Algunas aplicaciones no son muy sensibles a las variaciones en la forma, por lo que conviene usar un conjunto más simple como \tilde{A}_1 . Sin embargo, en muchas aplicaciones se usan conjuntos más elaborados como \tilde{A}_4 .

La utilidad de los conjuntos difusos depende mucho de la capacidad para construir funciones apropiadas para los diferentes conceptos dados en distintos contextos, y en muchos casos no es una tarea fácil. Los conjuntos difusos también pueden definirse mediante funciones estadísticas.

Definición 2. Sea X un conjunto rígido y A un subconjunto difuso de X . Se dice que A es normal si existe $x \in A$, tal que $A(x) = 1$. En otro caso, si para todo $x \in A$,

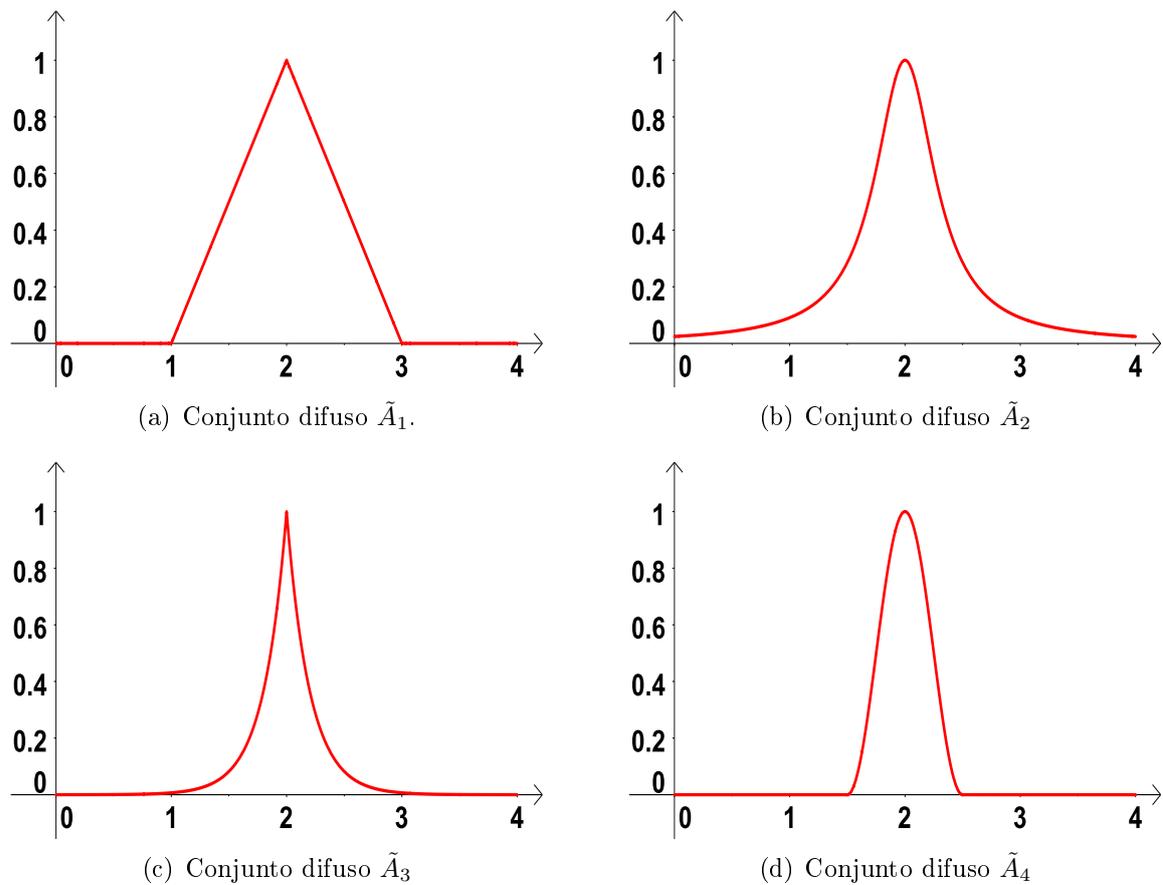


Figura 1.2: Conjuntos difusos que expresan el concepto general de una clase de números reales que son cercanos a 2. Las curvas son continuas, simétricas y monótonas.

$A(x) < 1$, entonces A se le denomina subnormal.

Es fácil ver que los conjuntos difusos del Ejemplo 2 son normales. Los conjuntos difusos definidos en los Ejemplos 2 y 1 se definen en los números reales. Ahora, consideremos el ejemplo 3 cuyo contexto es empleado cotidianamente.

Ejemplo 3. Sean A_1 , A_2 y A_3 conjuntos difusos que intentan capturar el concepto de personas jóvenes, adultas y ancianas, respectivamente. Una forma de definir estos conjuntos es:

$$A_1(x) = \begin{cases} 1, & \text{si } x \leq 20; \\ \frac{35-x}{15}, & \text{si } 20 < x < 35; \\ 0, & \text{si } x \geq 35. \end{cases}$$

$$A_2(x) = \begin{cases} 0, & \text{si } x \leq 20 \text{ o } x \geq 60; \\ \frac{x-20}{15}, & \text{si } 20 < x < 35; \\ \frac{60-x}{15}, & \text{si } 45 < x < 60; \\ 1, & \text{si } 35 \leq x \leq 45. \end{cases}$$

$$A_3(x) = \begin{cases} 0, & \text{si } x \leq 45; \\ \frac{x-45}{15}, & \text{si } 45 < x < 60; \\ 1, & \text{si } x \geq 60, \end{cases}$$

donde el valor de x representa la edad de una persona. En la Figura 1.3 se puede observar que, por ejemplo, una persona con 25 años de edad se considera una persona joven con un grado de pertenencia de 0.6, mientras que se considera como persona adulta con un grado de 0.3 y como una persona anciana con un grado de 0.

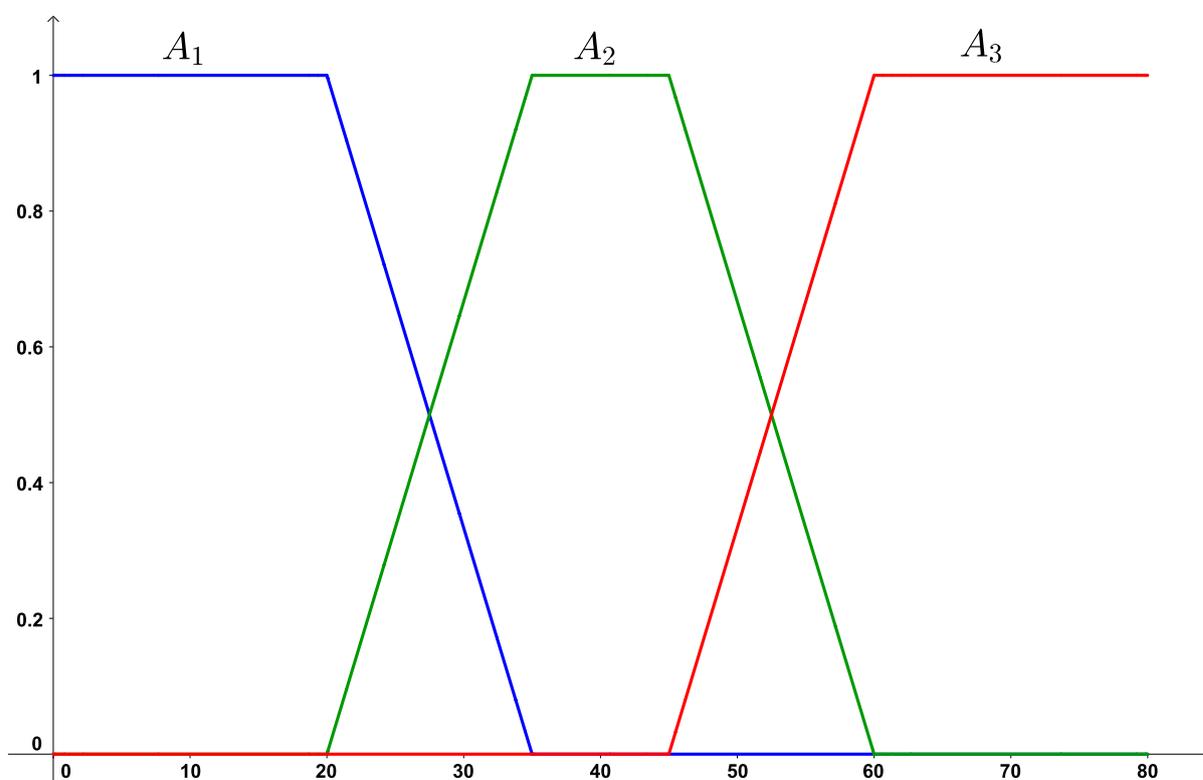


Figura 1.3: Conjuntos difusos que representan al conjunto de las personas jóvenes, maduras y ancianas, respectivamente.

1.2. Operaciones básicas de los conjuntos difusos

En la teoría de los conjunto rígidos se cumplen tres operaciones básicas que son: el complemento, la intersección y unión. Además, se plantean los conceptos de subconjunto e igualdad de conjuntos. Estos conceptos se mencionan en la Definición 1.

Teorema 1. Sea X un conjunto rígido. Si A, B son subconjuntos rígidos de X entonces para todo $x \in X$ se cumple que:

1. $\chi_{A \cup B}(x) = \text{máx} \{\chi_A(x), \chi_B(x)\}$.
2. $\chi_{A \cap B}(x) = \text{mín} \{\chi_A(x), \chi_B(x)\}$.
3. $\chi_{A^c}(x) = 1 - \chi_A(x)$.
4. $A = B$, si y solo si $\chi_A(x) = \chi_B(x)$.
5. $A \subseteq B$, si y solo si $\chi_A(x) \leq \chi_B(x)$.

Para la demostración del Teorema 1 solo se probará el primer inciso ya que los otros casos se prueban de forma análoga

Demostración. Por demostrar que $\chi_{A \cup B}(x) = \text{máx} \{\chi_A(x), \chi_B(x)\}$. La función característica $\chi_{A \cup B}(x)$ está dada por:

$$\chi_{A \cup B}(x) = \begin{cases} 1, & \text{si } x \in A \cup B; \\ 0, & \text{en otro caso.} \end{cases} \quad (1.3)$$

Para esta prueba, se deben considerar los siguientes casos:

- $x \in A$ y $x \notin B$, entonces $\chi_A(x) = 1$ y $\chi_B(x) = 0$, luego $\text{máx} \{\chi_A(x), \chi_B(x)\} = 1$.
- $x \notin A$ y $x \in B$, entonces $\chi_A(x) = 0$ y $\chi_B(x) = 1$, luego $\text{máx} \{\chi_A(x), \chi_B(x)\} = 1$.
- $x \notin A$ y $x \notin B$, entonces $\chi_A(x) = 0$ y $\chi_B(x) = 0$, luego $\text{máx} \{\chi_A(x), \chi_B(x)\} = 0$.
- $x \in A$ y $x \in B$, entonces $\chi_A(x) = 1$ y $\chi_B(x) = 1$, luego $\text{máx} \{\chi_A(x), \chi_B(x)\} = 1$.

Es decir, $\text{máx} \{\chi_A(x), \chi_B(x)\} = 1$ si $x \in A \cup B$ y $\text{máx} \{\chi_A(x), \chi_B(x)\} = 0$, en otro caso. Por lo tanto, $\chi_{A \cup B}(x) = \text{máx} \{\chi_A(x), \chi_B(x)\}$. \square

Estos conceptos pueden ser extendidos a la teoría de conjuntos difusos como se observa en la Definición 3.

Definición 3. Sea X un conjunto rígido. Si A, B son subconjuntos difusos de X se define, para todo $x \in X$:

- $(A \cup B)(x) = \text{máx} \{A(x), B(x)\}$,
- $(A \cap B)(x) = \text{mín} \{A(x), B(x)\}$,
- $(A^c)(x) = 1 - A(x)$.

Definición 4. Sea X un conjunto rígido. Si A, B son subconjuntos difusos de X , se dice que $A \subseteq B$, si y solo si $A(x) \leq B(x)$, para todo $x \in X$.

Al complemento, unión e intersección de conjuntos difusos se les conoce como *operaciones estándar de conjuntos difusos*.

Ejemplo 4. Considere el Ejemplo 3. La operación $A_1 \cap A_2$ está dada por la función dada en (1.4) y en la Figura 1.4(a) se muestra su gráfica. Es decir, en la intersección del conjunto de las personas jóvenes (A_1) con el de las personas adultas (A_2) se encuentran las personas que tienen entre 20 y 35 años. Por otro lado, la intersección del conjunto de las personas adultas con el de las personas ancianas ($A_2 \cap A_3$) corresponde a las personas entre 45 y 60 años (ver función dada en (1.5) y Figura 1.4(b)).

$$(A_1 \cap A_2)(x) = \begin{cases} 0, & \text{si } x \leq 20 \text{ o } x \geq 35; \\ \frac{x-20}{15}, & \text{si } 20 < x < 27.5; \\ \frac{35-x}{15}, & \text{si } 27.5 \leq x < 35. \end{cases} \quad (1.4)$$

$$(A_2 \cap A_3)(x) = \begin{cases} 0, & \text{si } x \leq 45 \text{ o } x \geq 60; \\ \frac{x-45}{15}, & \text{si } 45 < x < 52.5; \\ \frac{60-x}{15}, & \text{si } 52.5 \leq x < 60. \end{cases} \quad (1.5)$$

Con estos dos casos ($A_1 \cap A_2$ y $A_2 \cap A_3$) se puede observar que a pesar de que algunos conjuntos difusos sean normales, no necesariamente su intersección lo es.

Siguiendo con el Ejemplo 3, el complemento de los conjuntos A_1 y A_3 están dados en (1.6) y (1.7), respectivamente.

$$A_1^c(x) = \begin{cases} 0, & \text{si } x \leq 20; \\ \frac{x-20}{15}, & \text{si } 20 < x < 35; \\ 1, & \text{si } x \geq 35. \end{cases} \quad (1.6)$$

$$A_3^c(x) = \begin{cases} 1, & \text{si } x \leq 45; \\ \frac{60-x}{15}, & \text{si } 45 < x < 60; \\ 0, & \text{si } x \geq 60. \end{cases} \quad (1.7)$$

Claramente se puede corroborar que para este ejemplo $A_1^c \cap A_3^c = A_2$. La Figura 1.5(a) corresponde al conjunto difuso A_1^c mientras que la Figura 1.5(b) corresponde al conjunto difuso A_3^c . Se puede observar que la Figura 1.5(c), que representa la operación $A_1^c \cap A_3^c$ es igual al conjunto difuso A_2 (ver Figura 1.3).

El ejemplo más sencillo de un subconjunto difuso es el conjunto vacío, denotado por \emptyset , y cumple que para todo $x \in X$, $\emptyset(x) = 0$. También se cumple que $\emptyset \subseteq A$, para cualquier conjunto difuso A .

La propiedad transitiva de la inclusión de conjuntos rígidos también se cumple para el caso de conjuntos difusos como se puede ver en el Teorema 2.

Teorema 2. Sean A , B y C conjuntos difusos definidos sobre un conjunto rígido X . Si $A \subseteq B$ y $B \subseteq C$ entonces $A \subseteq C$.

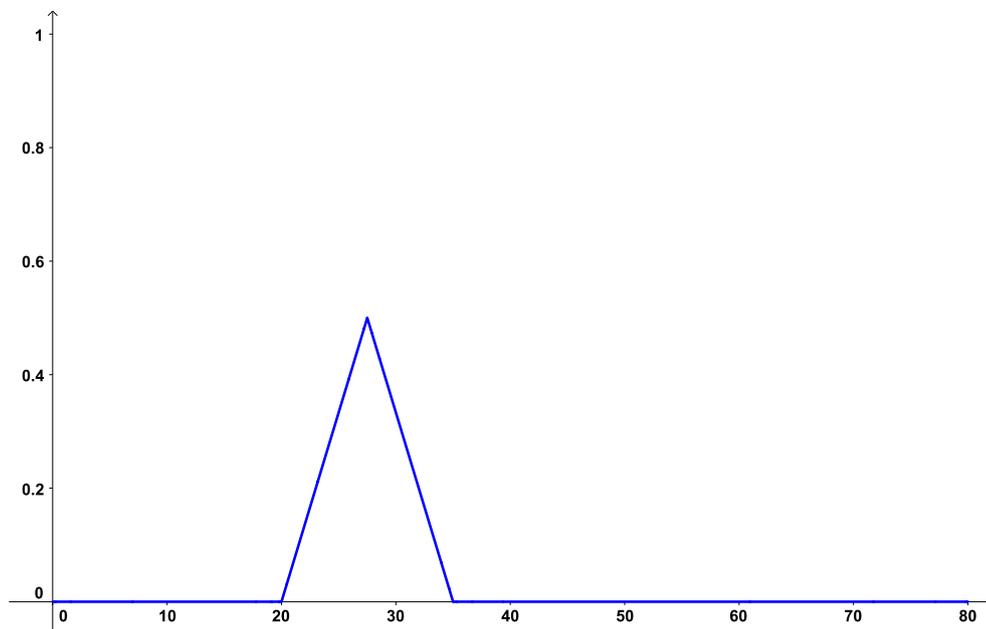
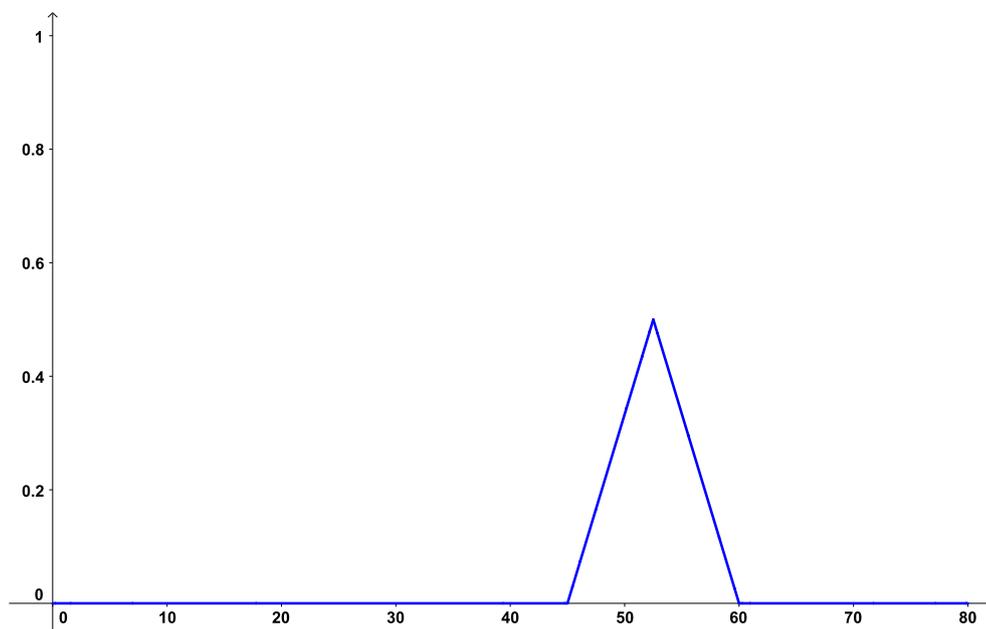
(a) Gráfica del conjunto difuso $A_1 \cap A_2$.(b) Gráfica del conjunto difuso $A_2 \cap A_3$.

Figura 1.4: Intersección estándar de los conjuntos difusos del Ejemplo 3.

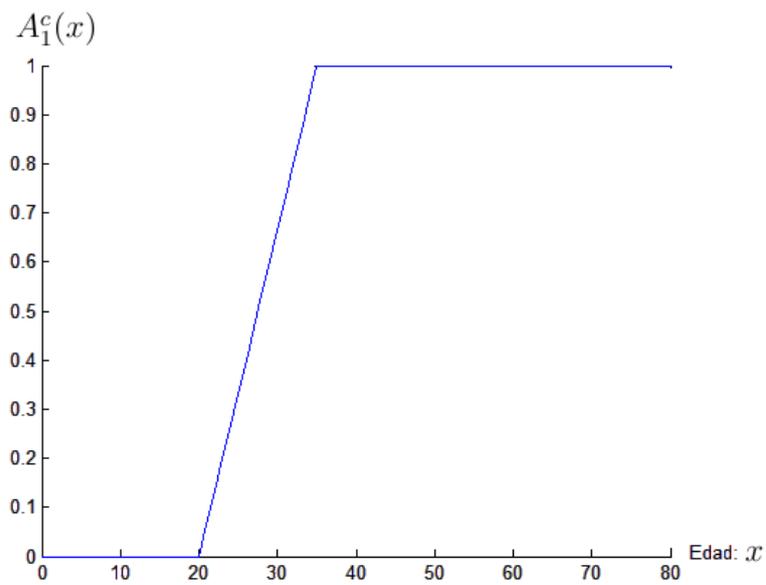
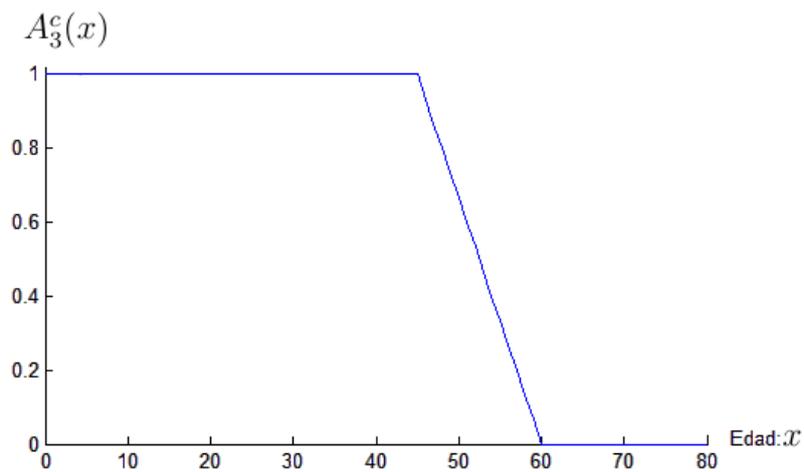
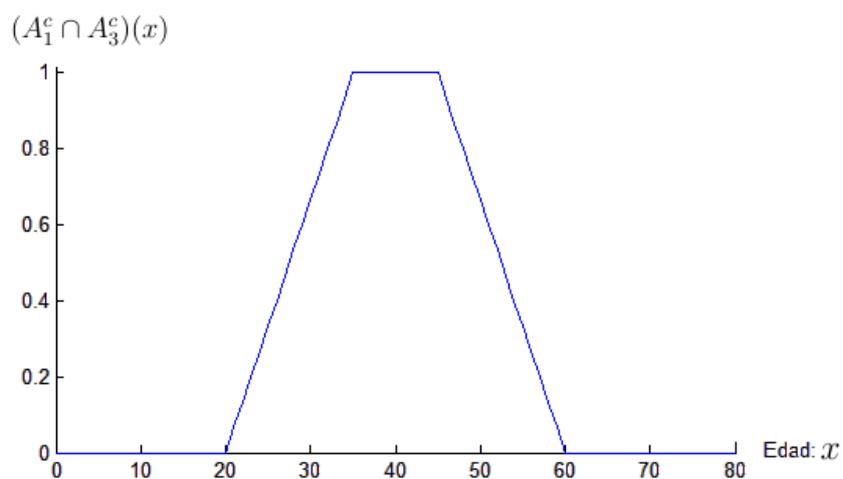
(a) Gráfica del conjunto difuso A_1^c .(b) Gráfica del conjunto difuso A_3^c .(c) Gráfica del conjunto difuso $A_1^c \cap A_3^c$.

Figura 1.5: Complemento e intersección de los conjuntos difusos A_1 y A_3 del Ejemplo 3.

Demostración. Si $A \subseteq B$ y $B \subseteq C$, por la Definición 3 se cumple que:

$$A(x) \leq B(x) \text{ y } B(x) \leq C(x), \text{ para todo } x \in X.$$

entonces $A(x) \leq C(x)$, para todo $x \in X$. Por lo tanto $A \subseteq C$. \square

A continuación se presentan propiedades de las operaciones de conjuntos difusos, que también se cumplen en los conjuntos rígidos.

Teorema 3. Sean X un conjunto rígido y A, B, C subconjuntos difusos de X . Se cumple:

1. *Ley conmutativa:*

$$A \cup B = B \cup A \text{ y } A \cap B = B \cap A.$$

2. *Ley asociativa:*

$$(A \cup B) \cup C = A \cup (B \cup C) \text{ y } (A \cap B) \cap C = A \cap (B \cap C).$$

3. *Ley idempotencia:*

$$A \cup A = A \text{ y } A \cap A = A.$$

4. *Ley monótona:*

$$\text{Si } A \subseteq B \text{ entonces } A \cup C \subseteq B \cup C.$$

$$\text{Si } A \subseteq B \text{ entonces } A \cap C \subseteq B \cap C.$$

5. *Ley distributiva*

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C).$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$$

6. *Ley de identidad*

$$A \cup \emptyset = A.$$

$$X \cap A = A.$$

7. *Ley de absorción por X y \emptyset*

$$A \cap \emptyset = \emptyset.$$

$$A \cup X = X.$$

8. *Ley de involución*

$$A = (A^c)^c.$$

$$A \subseteq B \text{ si y sólo si } B^c \subseteq A^c.$$

9. *Leyes de De Morgan:*

$$(A \cup B)^c = A^c \cap B^c.$$

$$(A \cap B)^c = A^c \cup B^c.$$

Para la demostración de cada ley del Teorema 3 se puede consultar ([4]) o ([8]).

A pesar de que muchos conceptos se pueden generalizar para conjuntos difusos, existen propiedades del complemento que no se cumplen para ellos. Por ejemplo, la unión de un conjunto con su complemento no necesariamente es el conjunto universal y su intersección no es el conjunto vacío. Esto es:

$$A \cup A^c \neq X \text{ y } A \cap A^c \neq \emptyset.$$

En efecto, si $x \in A$ con un grado de pertenencia $A(x)$, también $x \in A^c$ con $A^c(x) = 1 - A(x)$. Para el caso de la intersección de A con A^c , el grado de pertenencia está dado por:

$$(A \cap A^c)(x) = \text{mín} \{A(x), A^c(x)\}, \text{ para todo } x \in X.$$

Si $0 < A(x) < 1$ entonces $0 < (A \cap A^c)(x) < 1$, es decir, $x \in A \cap A^c$ por lo que la intersección no es vacía. En el caso de la unión, no se cumple que $A \cup A^c$ sea igual a X , ya que en esa unión solo están los elementos de A pero con un grado de pertenencia distinto.

Para una familia de subconjuntos difusos $(A_i)_{i \in I}$ de un conjunto X , donde I es un conjunto de índices, se definen:

$$A = \bigcup_{i \in I} A_i, \text{ con } A(x) = \sup_{i \in I} \{A_i(x)\} \text{ para todo } x \in X.$$

$$A = \bigcap_{i \in I} A_i, \text{ con } A(x) = \inf_{i \in I} \{A_i(x)\} \text{ para todo } x \in X.$$

1.3. Clasificación de los conjuntos difusos

Los conjuntos difusos que se han analizado hasta el momento, es decir, aquellos conjuntos que a cada elemento $x \in X$, le asigna un valor particular en el intervalo $[0, 1]$ (ver Ejemplos 1, 2 y 3), también se les conoce como *conjuntos difusos ordinarios* o de *tipo 1*. Sin embargo, en algunas aplicaciones el valor de $A(x)$ solo puede ser aproximado. Por ejemplo, se puede aproximar una cota inferior o superior para $A(x)$ y tomar alguna de las dos aproximaciones o el promedio de ellas. En estos casos solo se le asigna un valor a x . Pero los conjuntos difusos ordinarios se pueden generalizar de dos maneras:

- Conjuntos difusos valuados en un intervalo: Asignan a cada elemento $x \in X$ un intervalo cerrado de números reales entre las cotas inferior y superior mencionadas anteriormente. De manera formal se definen como:

$$A : X \rightarrow \epsilon([0, 1])$$

donde $\epsilon([0, 1])$ denota la familia de todos los intervalos cerrados contenidos en $[0, 1]$.

- Conjuntos difusos de tipo 2: asignan al elemento $x \in X$ un conjunto difuso, es decir:

$$A : X \rightarrow F([0, 1])$$

donde $F([0, 1])$ denota el conjunto de todos los conjuntos difusos ordinarios definidos en el conjunto universal $[0, 1]$. A $F([0, 1])$ se le conoce como conjunto difuso potencia de $[0, 1]$.

Ejemplo 5. Consideré el conjunto de personas $Y = \{ \text{Juan, Pedro, María} \}$ y los conjuntos difusos ordinarios A_i , para $i = 1, 2, 3$, definidos en el Ejemplo 3. Un ejemplo de un conjunto difuso tipo 2 está dado por la función:

$$A : Y \rightarrow F([0, 1]), \quad (1.8)$$

tal que,

$$A(\text{Juan}) = A_2, \quad A(\text{Pedro}) = A_3 \quad \text{y} \quad A(\text{María}) = A_1.$$

Los conjuntos difusos valuados en un intervalo tienen la ventaja de ser más realistas en algunas aplicaciones. La desventaja de estos conjuntos, es que computacionalmente hablando son muy costosos. Los conjuntos difusos del tipo 2 poseen una gran potencia expresiva, y por tanto son conceptualmente más interesantes. Sin embargo, son mucho más costosos computacionalmente hablando que los conjuntos difusos valuados en un intervalo, razón por la que hasta el momento no han sido usados en alguna aplicación ([13]). En la presente tesis solo se utilizarán conjuntos difusos ordinarios.

Como a los elementos de un conjunto difuso se les asigna un valor entre $[0, 1]$, a veces se puede cometer el error de confundir la probabilidad con la matemática difusa, pero se debe tener en cuenta que la probabilidad mide la frecuencia con la que ocurre un evento, mientras que la matemática difusa describe objetos que se caracterizan por el grado de imprecisión de algunas de sus propiedades ([22]).

1.4. Relaciones difusas

Una *relación rígida* representa la presencia o ausencia de interconexiones entre los elementos de dos o más conjuntos.

Definición 5. Sean X, Y conjuntos rígidos. Una relación rígida R de X en Y es un subconjunto del producto cartesiano $X \times Y$.

A esta relación se le conoce como relación binaria. Cada relación rígida R , al igual que los conjuntos rígidos, puede ser definida por una función característica, la cual asigna el valor de 1 si la pareja del conjunto universal pertenece a la relación y 0 en caso contrario. Formalmente se define como:

$$\chi_R(x, y) = \begin{cases} 1, & \text{si y sólo si } (x, y) \in R; \\ 0, & \text{en otro caso.} \end{cases}$$

El valor de $\chi_R(x, y)$ en una relación binaria indica que los elementos de la pareja están o no relacionados entre ellos.

Ejemplo 6. Sea X el conjunto de todos los hombres, Y el conjunto de todas las mujeres y R la relación de hombres que están casados con mujeres. Si Juan está casado con María entonces,

$$\chi_R(\text{Juan}, \text{María}) = 1,$$

lo que indica que el par (Juan, María) pertenece a la relación; por otro lado, si Pedro no está casado con Leticia, entonces

$$\chi_R(\text{Pedro}, \text{Leticia}) = 0,$$

que indica que el par (Pedro, Leticia) no pertenece a la relación.

El concepto de una relación rígida se puede generalizar para introducir diferentes grados de relación entre los elementos representados, lo que origina una relación difusa.

Definición 6. Sean X, Y conjuntos rígidos. Una relación difusa R de X en Y es un subconjunto difuso del producto cartesiano $X \times Y$.

El valor de $R(x, y)$ indica la fuerza de la relación que hay entre x e y . Es posible definir una relación difusa entre los elementos de un mismo conjunto, es decir, con $X = Y$.

Ejemplo 7. Sea $X = \{1, 2, 3, 4, 5\} = Y$ y considérese la relación difusa \bar{R} definida por:

$$\bar{R} : \text{“}x \text{ es aproximadamente igual a } y\text{”}.$$

\bar{R} debe de cumplir que para $x = y$, $\bar{R}(x, y) = 1$ y entre mayor sea $|x - y|$, menor debe ser $\bar{R}(x, y)$. Una forma de representar estas condiciones es:

$$\bar{R}(x, y) = \begin{cases} 1, & \text{si } x = y; \\ 0.8, & \text{si } |x - y| = 1; \\ 0.5, & \text{si } |x - y| = 2; \\ 0.2, & \text{si } |x - y| = 3; \\ 0, & \text{si } |x - y| = 4. \end{cases} \quad (1.9)$$

La Tabla 1.1 muestra los resultados obtenidos a partir de la función dada en (1.9). La primera fila corresponde a los elementos de Y , mientras que la primera columna contiene los elementos del conjunto X . Por ejemplo, $\bar{R}(2, 1) = 0.8$ y $\bar{R}(3, 1) = 0.5$ ya que 2 es más cercano a 1 que 3.

$X \setminus Y$	1	2	3	4	5
1	1	0.8	0.5	0.2	0
2	0.8	1	0.8	0.5	0.2
3	0.5	0.8	1	0.8	0.5
4	0.2	0.5	0.8	1	0.8
5	0	0.2	0.5	0.8	1

Tabla 1.1: Relación difusa definida por $\bar{R} : x$ aproximadamente igual a y .

En las relaciones rígidas existen los conceptos de reflexividad, simetría y transitividad, los cuales también pueden ser extendidas para relaciones difusas.

Definición 7. Sea X un conjunto rígido y R una relación difusa en X . Se dice que R es

1. Reflexiva, si para todo $x \in X$, $R(x, x) = 1$.
2. Simétrica, si para todo $x, y \in X$, $R(x, y) = R(y, x)$.
3. Transitiva, si para todo $x, z \in X$,

$$\sup_y \{ \min \{ (R(x, y), R(y, z)) \} \} \leq R(x, z).$$

Definición 8. Sea X un conjunto rígido. Una relación difusa R en X se dice que es una *relación difusa de equivalencia* o *relación de similitud*, si R es reflexiva, simétrica y transitiva.

Ejemplo 8. Claramente, la relación del ejemplo 7 es reflexiva y simétrica, pero no transitiva, ya que $\bar{R}(1, 3) = 0.5$ y $\sup_y \{ \min \{ (\bar{R}(1, y), \bar{R}(y, 3)) \} \} = 0.8$, porque:

$$\begin{aligned} \min \{ \bar{R}(1, 1), \bar{R}(1, 3) \} &= \min \{ 1, 0.5 \} = 0.5, \\ \min \{ \bar{R}(1, 2), \bar{R}(2, 3) \} &= \min \{ 0.8, 0.8 \} = 0.8, \\ \min \{ \bar{R}(1, 3), \bar{R}(3, 3) \} &= \min \{ 0.5, 0.1 \} = 0.5, \\ \min \{ \bar{R}(1, 4), \bar{R}(4, 3) \} &= \min \{ 0.2, 0.8 \} = 0.2, \\ \min \{ \bar{R}(1, 5), \bar{R}(5, 3) \} &= \min \{ 0, 0.5 \} = 0. \end{aligned}$$

Una relación de similitud puede aplicarse de dos formas. Primero, agrupando a los elementos en conjuntos rígidos cuyos grados son similares. Cuando este grado es igual a 1, el agrupamiento es una clase de equivalencia. Segundo, considerando los grados de similitud que los elementos de X tienen con algún elemento específico $x \in X$.

Definición 9. Dada una relación de similitud R en X , para cada $x \in X$ una clase de similitud es el un conjunto difuso A_x , dado por:

$$A_x(y) = R(x, y) \text{ para todo } y \in X.$$

Capítulo 2

Imágenes digitales

En la actualidad, con el avance de la tecnología y la aparición del internet, las imágenes digitales han invadido la vida de las personas, hasta el grado que las cámaras fotográficas digitales están desplazando a las analógicas. En cualquier área donde se utilizan imágenes es más fácil procesar imágenes digitales para obtener información como: reconocer objetos, medir su tamaño, su forma o posición [21]. El objetivo de este capítulo es describir la forma en que se representan las imágenes digitales en tono de gris mediante algunos conceptos matemáticos. También se dan algunas propiedades de los llamados *píxeles* que se utilizan para procesar esta clase de imágenes. Por último, se describe brevemente en que consiste el proceso de segmentación de una imagen digital y se mencionan algunos de los algoritmos que actualmente se utilizan para realizar esta tarea.

2.1. Tipos de imágenes

Todo lo que pasa por la mirada o frente al ojo interior puede entenderse como una imagen, o transformarse en una imagen ([5]). Una *imagen* puede verse plasmada en una fotografía, en una pintura, en una estatua, etc. Es decir, una imagen es fija y no cambia con el tiempo, en comparación con un video, que evoluciona con el tiempo y generalmente contiene movimientos de objetos [7].

Las imágenes han sido y son muy importantes en la sociedad, muestra de esto es que se han utilizado desde hace mucho tiempo; por ejemplo los cavernícolas realizaban pinturas que representaban sus cacerías. También, se conocen diferentes culturas principalmente en México que han dejado una gran variedad de imágenes grabadas en piedra. Es hasta los 80's del siglo *XIX* cuando se inventó el primer procedimiento fotográfico conocido como daguerrotipo, el cual solo tomaba fotos en blanco y negro, eran de baja calidad y tardaban alrededor de 3 días en revelarse. Durante ese siglo se inventó la fotografía a color pero no tuvo muchas aplicaciones comerciales debido a dificultades e imperfecciones. A finales del siglo *XX* apareció una nueva tecnología que permite la creación inmediata de imágenes: la cámara digital. Al principio, las cámaras

digitales eran muy costosas y pesadas (llegaban a pesar hasta 4 kg.) lo que hacía que muchas personas no pudieran tener acceso a ellas. Con el tiempo se han vuelto más accesibles, solo pesan algunos gramos y la resolución de las imágenes es mejor ([25]).

Existen diferentes áreas donde la fotografía ha sido de utilidad, como por ejemplo la astronomía, la fotogrametría y la física de partículas. En la astronomía se logró medir las posiciones y magnitudes de las estrellas. La fotogrametría ha producido mapas topográficos a partir de imágenes aéreas. Los físicos descubrieron varias partículas elementales a partir de una gran cantidad de imágenes de cámaras de burbujas de hidrógeno. Sin embargo, todo este proceso se llevaba a cabo de manera manual, lo cual consumía mucho tiempo al revisar la gran cantidad de imágenes disponibles. Algunos dispositivos fueron diseñados para procesar dichas imágenes, sin embargo solo eran adaptados a propósitos específicos, por lo cual en ese momento no se encontró una amplia aplicación al procesamiento de imágenes. En general, las imágenes sólo se utilizaron para la documentación, descripción cualitativa y la ilustración de los fenómenos observados ([3]).

Ahora, debido al rápido progreso de la tecnología de video y la computadora, las computadoras personales tienen la capacidad de procesar los datos de una imagen, incluyendo secuencias de imágenes de visualización 3-D. El procesamiento de imágenes se está expandiendo tan rápidamente que ahora se aplica a prácticamente todas las ciencias naturales y disciplinas técnicas ([12]).

2.1.1. Imágenes analógicas

Una imagen analógica es aquella que es capturada con una cámara, un microscopio, un telescopio o cualquier instrumento óptico que presenta una variación de sombras y tonos continuos. Una imagen analógica puede ser en blanco y negro, en escala de grises o a color. En la actualidad, estas imágenes no son muy utilizadas en la investigación, debido a que para procesarlas o manipularlas por una computadora primero deben ser transformadas a una imagen discreta. Esta transformación se le conoce como digitalización y lo que se obtiene es una imagen digital ([23]). En la Figura 2.1 se puede observar una imagen analógica.

2.1.2. Imágenes digitales

Las imágenes digitales son imágenes que se han convertido en un formato binario legible por la computadora, en el que utilizando únicamente los símbolos 0 y 1 se puede almacenar la información contenida en la imagen después de aplicarle un proceso de discretización. Las imágenes digitales se obtienen generalmente mediante la conversión de señales continuas en formato digital. Por digital se entiende que las señales se definen en un dominio discreto, es decir, toma valores en un conjunto discreto ([23]).

Hasta el momento, aún no se ha podido plasmar totalmente una imagen real en una imagen digital, de hecho, la manera en que alguna persona ve los objetos no ne-



Figura 2.1: Ejemplo de una imagen analógica.

cesariamente es como son en la realidad. El ser humano no puede ver todo lo que está a su alrededor con observar en una sola dirección, como se aprecia en la Figura 2.2. Más aún, en una fotografía lo que se hace es convertir una imagen en 3D a 2D, y hacerla más pequeña. Además, de una cámara a otra el brillo y la luminosidad pueden cambiar. También, una foto que es tomada en un cuarto donde se ilumina con una luz roja, no se ve igual que si se ilumina con una luz blanca. Aún con estos inconvenientes, en los últimos tiempos, el uso de imágenes digitales ha ido en aumento ([13]).



Figura 2.2: El ojo humano captura parte del paisaje. Imagen obtenida de http://www.el-nacional.com/ciencia_y_ambiente/.

Una imagen digital está compuesta por un número finito de elementos, los cuales se les conoce como *elementos de la imagen*, *pels* o *píxeles*. El término más utilizado es píxel ([3]). Una imagen digital se puede representar en una matriz, tal que cada elemento de la matriz representa un punto o píxel de la imagen. A cada píxel se le asigna un objeto matemático que contiene información relacionada con características ópticas de la imagen. Dicho objeto indica la tonalidad de ese punto ([7]). Dada la gran utilidad de las imágenes digitales, se ha tenido la necesidad de darle una representación matemática para poder estudiar sus propiedades. Así, matemáticamente el término imagen se refiere a una función bidimensional f , donde el valor de f en (x, y) está

relacionado con propiedades ópticas de la imagen, como brillo, color, luminosidad, etc., y se puede representar por un escalar o por un vector ([13]).

Imágenes binarias

Las imágenes digitales en blanco y negro son conocidas como *imágenes digitales binarias*. A cada píxel le corresponde un valor 1 o 0, donde 0 representa el tono negro, y 1 el tono blanco. La Figura 2.3 representa una imagen en blanco y negro. A esta imagen se le puede asociar una matriz, como se muestra en la Figura 2.4. En forma matemática, la función f correspondiente solo toma valores en $\{0, 1\}$ ([7]).

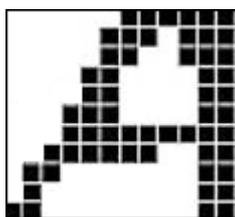


Figura 2.3: Imagen digital en blanco y negro.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Figura 2.4: Matriz correspondiente a la imagen de la Figura 2.3.

Imágenes en tonos de gris

Si la imagen está representada visualmente en tonos de gris (como una fotografía en blanco y negro), los valores de los píxeles corresponden a diferentes valores de gris. Los valores de la imagen deben ser cuantificados para el procesamiento digital. Una vez que se decidió el conjunto discreto de valores que definen el rango de nivel de gris, un método simple y eficiente de cuantificación debe redondear los valores de los píxeles de la imagen a los respectivos elementos cercanos de la gama de intensidad. Estos valores redondeados pueden ser cualquier número, pero por conveniencia conceptual y facilidad de formato digital, usualmente son mapeados por una transformación lineal

en un conjunto finito de números enteros no negativos $\{0, \dots, K - 1\}$, donde K es una potencia de dos: $K = 2^B$. Por lo tanto, el número de niveles de gris admisibles es K , y el número de bits asignados al nivel de gris de cada píxel es B . Por lo general para imágenes binarias se utiliza $B = 1$ mientras que para imágenes en tono de gris se utiliza $B = 8$, donde cada nivel de gris ocupa un byte, estas son las profundidades de bits más comunes ([7]). Con la cuantificación simple (el redondeo de nivel de gris) la principal preocupación es que las intensidades de los píxeles o niveles de gris deben ser cuantificados con precisión suficiente para que no se pierda información. Por esta razón, usualmente los niveles de gris de una imagen digital se encuentran en $[0, 255]$. Dicho intervalo se conoce como *escala de grises*. Los valores extremos del intervalo, 0 y 255, representan los tonos negro y blanco respectivamente, y los valores intermedios son tonos de gris que van degradándose continuamente entre el negro y el blanco ([12]). En la Figura 2.5 se puede observar una imagen en tonos de gris con su respectiva representación matricial.

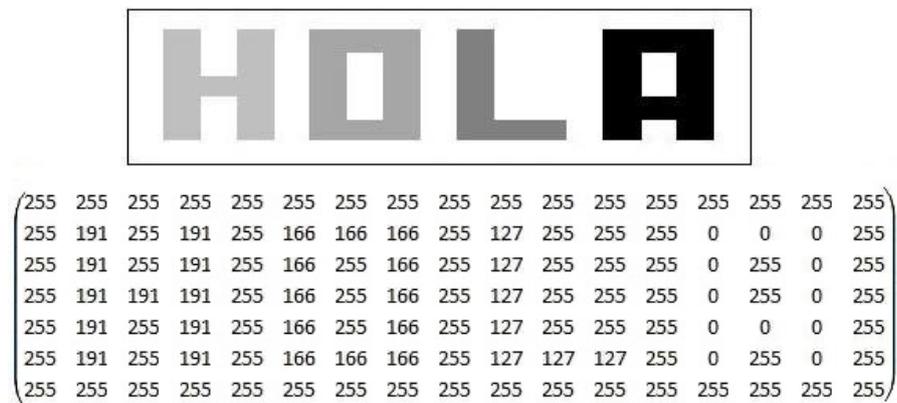


Figura 2.5: Ejemplo de una imagen en tonos de gris con su respectiva representación matricial.

Si se hacen cuantificaciones diferentes para una imagen, se obtienen diferentes resultados. Por ejemplo, en la Figura 2.6, se muestran cuantificaciones en 8, 4, 2 y 1 bit(s) de la resolución de nivel de gris. A 8 bits, la imagen es muy nítida y se puede distinguir que corresponde a cascarones de huevos. A 4 bits, los huevos toman la apariencia de estar rayados o pintados como los huevos de Pascua. Este efecto se conoce como "falso contorno", y resulta cuando se utiliza una resolución inadecuada de escala de grises para representar diferentes regiones de una imagen. En esos lugares, los efectos del nivel de gris pueden ser visualmente exagerados, y conducen a una apariencia de estructuras falsas. A 2 bits y 1 bit, la información importante se ha perdido de la imagen, por lo que es difícil reconocer los objetos en ella ([7]).

Imágenes a color

Para representar una imagen a color, el valor de f en (x, y) está dado por un vector, el cual se construye de diversas formas, dependiendo de las características que se utilicen

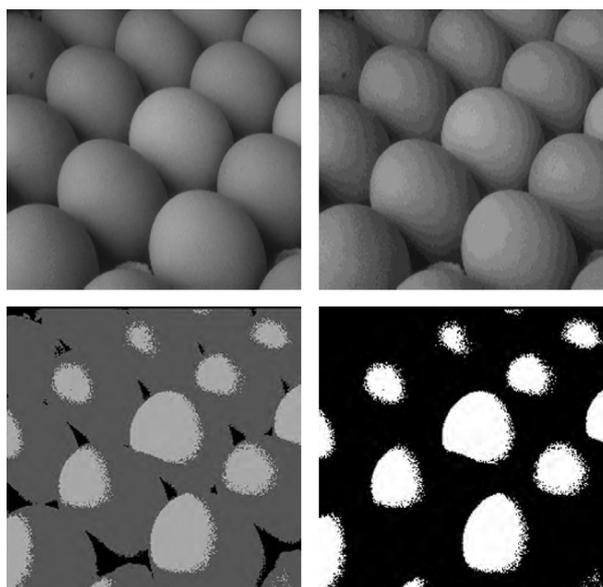


Figura 2.6: Imagen de huevos, cuando se cuantifica a 8, 4, 2 y 1 bit(s). Imagen obtenida de [7].

para describir un color. Con este fin se han creado varios sistemas. A continuación se describen tres de los más importantes ([18]).

RGB: Este sistema usa tres canales de tonos de color: rojo, verde y azul. Estos colores se derivan de los receptores de color que percibe el ojo humano (ver Figura 2.7).

CMYK: Tiene cuatro canales: cian, magenta, amarillo y negro. Este sistema es el estándar que se utiliza en la imprenta (ver Figura 2.8).

HSV o Hue Saturation Value: Este sistema almacena la información de color en tres canales, al igual que RGB. Los dos primeros canales se refieren a la información de color y el tercer canal conocido como “value” corresponde al brillo. El sistema HSV es especialmente útil en compresión de video con pérdida, donde la pérdida de información del color es menos perceptible al ojo humano (ver Figura 2.9).

2.2. Píxeles

Los píxeles de una imagen digital, ya sea en escala de grises o, en blanco y negro, se pueden representar en el plano \mathbb{Z}^2 , esto es, un píxel p se asocia con un punto (x, y) , con $x, y \in \mathbb{Z}$ ([13]). Existen diferentes relaciones importantes entre los píxeles de una imagen, por ejemplo, la conexidad que permite establecer los límites de los objetos y los componentes de áreas en una imagen. Dichas relaciones ayudarán a realizar la segmentación que se propone en esta tesis.

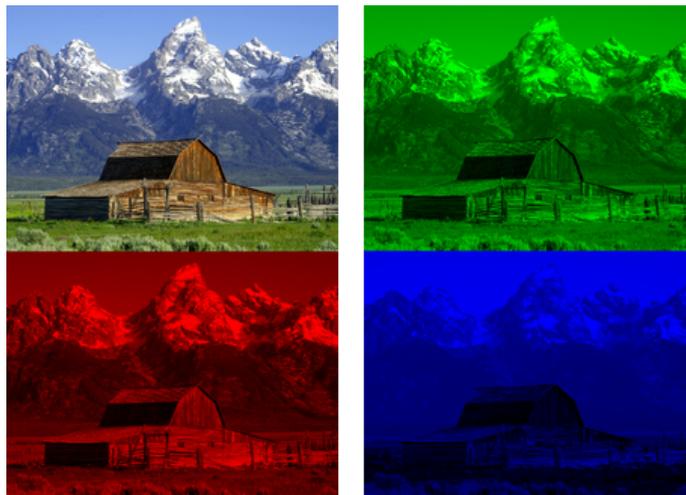


Figura 2.7: Imagen en el sistema RGB, donde la primera imagen es la original y las otras corresponden a la forma en que se ve con cada canal de color: verde, rojo y azul, respectivamente. Imagen obtenida de [7].

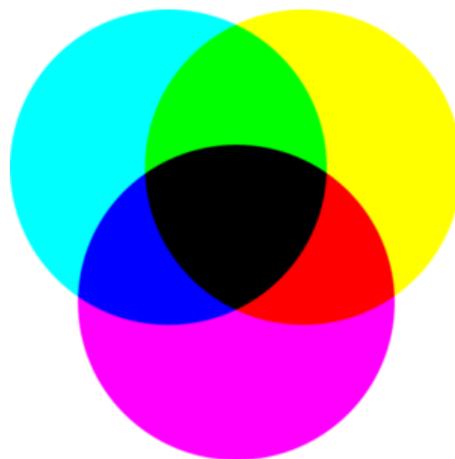


Figura 2.8: Canales de color del sistema CMYK. Imagen obtenida de [18].

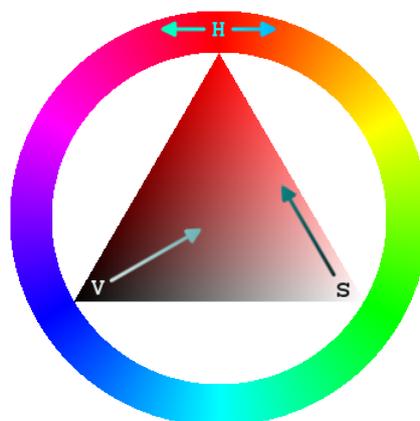


Figura 2.9: Espacio de color HSV. Imagen obtenida de [18].

Definición 10. A los elementos de \mathbb{Z}^2 se les conoce como píxeles.

En la segmentación de imágenes digitales es importante definir la conexidad entre píxeles que nos permite saber si dos píxeles están conectados, para esto es importante dar las definiciones de m -adyacencia, m -vecino y m -componente, para $m = 4, 8$.

Definición 11. Sea $p = (x, y)$ un píxel. El conjunto 4-vecinos de p , denotado por $N_4(p)$, se define por:

$$N_4(p) = \{(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)\}.$$

Ejemplo 9. Considere la gráfica de la Figura 2.10. El conjunto $N_4(p)$, donde $p = (2, 2)$, es:

$$N_4(p) = \{(3, 2), (1, 2), (2, 3), (2, 1)\}.$$

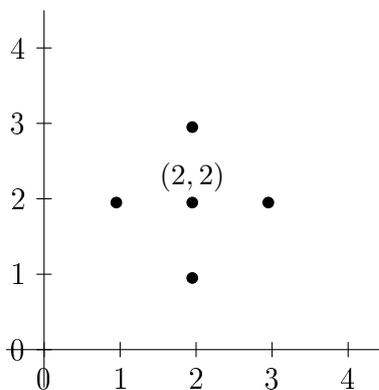


Figura 2.10: Representación de $N_4(p)$, donde $p = (2, 2)$.

Definición 12. Sea $p = (x, y)$ un píxel. El conjunto de vecinos diagonales de p , denotado por $N_D(p)$, se define como:

$$N_D(p) = \{(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)\}.$$

Ejemplo 10. En la Figura 2.11 se puede ver que:

$$N_D(p) = \{(3, 3), (1, 3), (3, 1), (1, 1)\},$$

donde $p = (2, 2)$.

Definición 13. Dado un píxel p , el conjunto 8-vecinos de p se define como:

$$N_8(p) = N_4(p) \cup N_D(p).$$

Ejemplo 11. Si $p = (2, 2)$ entonces

$$N_8(p) = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2), (3, 3)\},$$

(ver Figura 2.12).

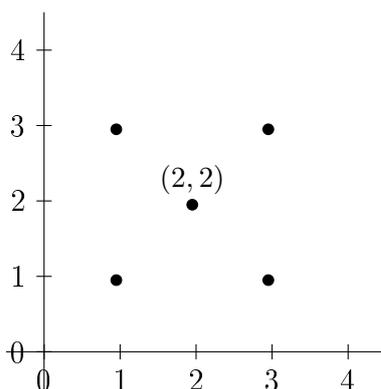


Figura 2.11: Representación de $N_D(p)$, donde $p = (2, 2)$.

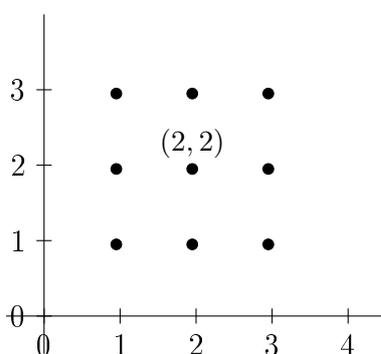


Figura 2.12: Representación de $N_8(p)$, donde $p = (2, 2)$.

En la aplicación se utiliza más el conjunto de 4 u 8-vecinos pero es importante saber que existe el conjunto de vecinos diagonales de un píxel. Utilizando los conceptos anteriores se puede definir dos tipos de adyacencia entre píxeles:

Definición 14. Sean p y q dos píxeles y $m \in \{4, 8\}$. Se dice que p y q son m -adyacentes si $q \in N_m(p)$.

En el Ejemplo 9, el píxel con coordenadas $(2, 3)$ es 4-adyacente al píxel $(2, 2)$, mientras que en el Ejemplo 11, los píxeles con coordenadas $(3, 3)$ y $(2, 2)$ son 8-adyacentes. Este concepto también se puede extender a conjuntos como se muestra en la siguiente definición.

Definición 15. Se dice que dos conjuntos de píxeles U y V son m -adyacentes, para $m \in \{4, 8\}$, si existen $p \in V$ y $q \in U$ tales que p y q son m -adyacentes.

Ejemplo 12. Considere a los conjuntos de píxeles V y U definidos por:

$$V = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\},$$

$$U = \{(4, 4), (4, 5), (5, 5)\}.$$

Su representación gráfica se muestra en la Figura 2.13. Los píxeles con coordenadas

(3,3) y (4,4) son 8-adyacentes, por tanto los conjuntos U y V son 8-adyacentes. Cabe notar que los conjuntos U y V no son 4-adyacentes, dado que no existen $p \in V$ y $q \in U$ tales que p y q sean 4-adyacentes. Sin embargo, los siguientes pares de conjuntos son 4-adyacentes:

- $\bar{V} = V \cup \{(4, 3)\}$ y U ,
- $\hat{V} = V \cup \{(3, 4)\}$ y U ,
- V y $U^* = U \cup \{(4, 3)\}$.

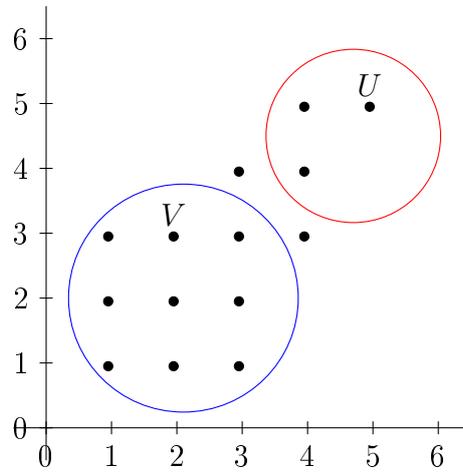


Figura 2.13: Ejemplo de dos conjuntos U y V que 8-adyacentes.

El concepto de m -adyacencia permite definir un m -camino entre dos píxeles cualesquiera en un conjunto, este m -camino ayudara a definir una conexión entre dos píxeles.

Definición 16. Sean V un conjunto de píxeles y $p, q \in V$. Un m -camino de p a q en V es una sucesión ordenada $\{p_i\}_{i=1}^n$ en V , tal que $p_1 = p$, $p_n = q$ y p_i es m -adyacente a p_{i+1} , para $1 \leq i \leq n - 1$.

Ejemplo 13. Considere los píxeles $r = (2, 0)$ y $s = (4, 3)$. Un ejemplo de un 8-camino de r a s se puede formar pasando por los píxeles $(2,1)$ y $(3,2)$. Esto es, un 8-camino de r a s está dado por la siguiente sucesión de píxeles:

$$\{(2, 0), (2, 1), (3, 2), (4, 3)\}.$$

Otro 8-camino de r a s se puede formar por la siguiente sucesión de píxeles:

$$\{(2, 0), (1, 1), (1, 2), (0, 3), (1, 3), (1, 2), (2, 3), (3, 2), (4, 3)\}.$$

En la Figura 2.14 se observan los dos 8-caminos de r a s mencionados anteriormente. Se puede ver claramente que dichos 8-caminos no pueden ser 4-caminos, debido a que existen píxeles en la sucesión que no son 4-adyacentes, por ejemplo los píxeles $(2,1)$ con $(3,2)$ y $(1,2)$ con $(0,3)$.

Por otro lado, en la Figura 2.15 se puede observar un 4-camino del píxel $u = (5, 0)$ al $v = (0, 3)$ dado por la siguiente sucesión de píxeles:

$$\{(5, 0), (4, 0), (3, 0), (2, 0), (2, 1), (1, 1), (1, 2), (2, 2), (2, 3), (1, 3), (1, 2), (1, 3), (0, 3)\}.$$

Con este ejemplo se puede ver que dados dos píxeles p y q , se pueden formar varios m -caminos de p a q . Además, un m -camino puede pasar varias veces por un píxel, observe el 4-camino de este ejemplo, pasa dos veces por el píxel $(1, 2)$.

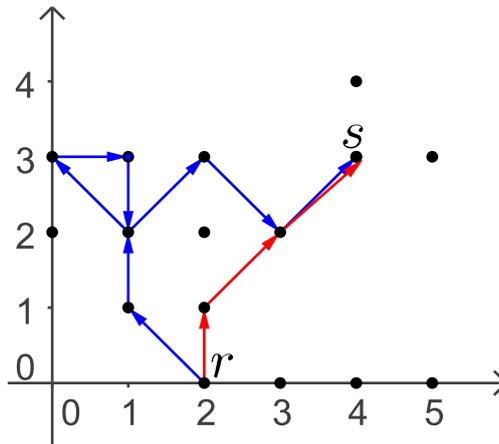


Figura 2.14: Representación de dos 8-caminos del píxel r al s .

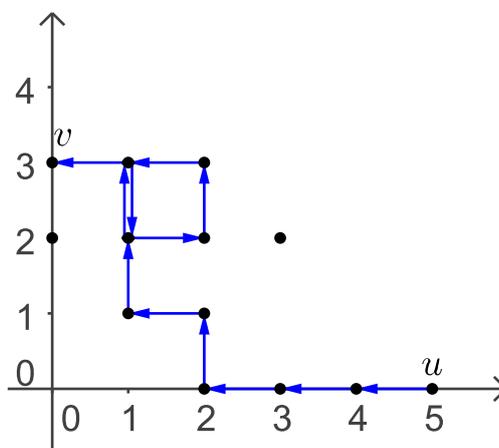


Figura 2.15: Representación de un 4-camino del píxel u al píxel v .

La definición de un m -camino permite dar las siguientes definiciones ([13]).

Definición 17. Sea V un conjunto de píxeles, $p, q \in V$ y $m \in \{4, 8\}$. Se dice que p y q están en relación de m -conexidad R_m , si y sólo si, existe un m -camino en V de p a q .

Se puede ver que R_m es una relación de equivalencia en V .

Definición 18. Sea V un conjunto de píxeles y $m \in \{4, 8\}$. Se dice que V es m -conexo, si y sólo si, $pR_m q$, para todo $p, q \in V$.

Definición 19. Dado $m \in \{4, 8\}$, una m -componente de un conjunto de píxeles V es una clase de equivalencia de V respecto a R_m .

Las m -componentes distintas de un conjunto de píxeles son disjuntas y cualquier par de píxeles de una misma m -componente están conectados por un m -camino en V .

A partir de la representación de coordenadas de un píxel, se puede definir una distancia entre dos píxeles.

Definición 20. Sean $p = (x, y)$ y $q = (s, t)$ dos píxeles. Se definen las siguientes distancias entre p y q ([12]):

- Distancia euclidiana: $D_c(p, q) = [(x - s)^2 + (y - t)^2]^{\frac{1}{2}}$.
- Distancia D_4 : $D_4(p, q) = |x - s| + |y - t|$.
- Distancia D_8 : $D_8(p, q) = \max(|x - s|, |y - t|)$.

Ejemplo 14. Considere el píxel $p = (2, 2)$. Al calcular la distancia $D_4(p, q)$, para $q \in N_4(p)$, se obtienen los siguientes resultados:

$$\begin{aligned} D_4((2, 2), (2, 3)) &= |2 - 2| + |2 - 3| = 1, & D_4((2, 2), (2, 1)) &= |2 - 2| + |2 - 1| = 1, \\ D_4((2, 2), (1, 2)) &= |2 - 1| + |2 - 2| = 1, & D_4((2, 2), (3, 2)) &= |2 - 3| + |2 - 2| = 1. \end{aligned}$$

Otros ejemplos del cálculo de la distancia D_4 son:

$$\begin{aligned} D_4((2, 2), (2, 4)) &= |2 - 2| + |2 - 4| = 2, & D_4((2, 2), (2, 0)) &= |2 - 2| + |2 - 0| = 2, \\ D_4((2, 2), (0, 2)) &= |2 - 0| + |2 - 2| = 2, & D_4((2, 2), (4, 2)) &= |2 - 4| + |2 - 2| = 2. \end{aligned}$$

De forma similar, se pueden calcular las distancias D_c , D_4 y D_8 para cualesquiera $p, q \in \mathbb{Z}_+^2$. La Figura 2.16 es una representación gráfica de los píxeles $p = (2, 2)$ y q , tal que $D_4(p, q) \leq 2$. El número en cada píxel q es la distancia $D_4(p, q)$.

Haciendo el mismo análisis del ejemplo 14 se obtiene que $D_8(p, q) = 1$, para $q \in N_8(p)$. Se puede concluir que dado un píxel q , los 4-vecinos de p son aquellos píxeles q tales que $D_4(p, q) = 1$, mientras que los 8-vecinos de p cumplen con $D_8(p, q) = 1$.

2.3. Segmentación de imágenes digitales

El objetivo de esta tesis es dar un algoritmo de segmentación de imágenes digitales por lo que en esta sección se explica brevemente en que consiste la segmentación, algunas aplicaciones, así como los diferentes algoritmos con los que se ha atacado este problema. La segmentación es una etapa muy importante del procesamiento de imágenes digitales, por esto, es importante primero describir este proceso.

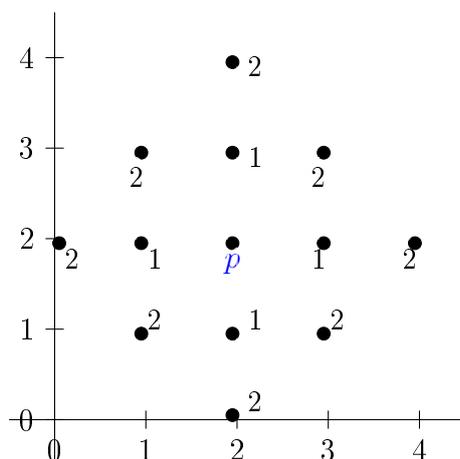


Figura 2.16: Representación gráfica de los píxeles p y q tal que $D_4(p, q) \leq 2$, donde $p = (2, 2)$.

2.3.1. Procesamiento de imágenes digitales

Gracias al avance vertiginoso que ha sufrido el hardware y la importancia que ha adquirido en la vida moderna, el Procesamiento Digital de Imágenes (PDI) ha tenido gran desarrollo, por lo que se ha utilizado en diferentes áreas de la investigación. El PDI es el conjunto de procedimientos que se pueden realizar sobre una imagen como ([12]):

- El mejoramiento (aumento del contraste de una imagen, la eliminación del ruido, entre otras),
- La restauración (tipos de degradaciones en el momento de la captura de la imagen),
- Reconstrucción,
- Segmentación,
- Compresión.

Algunos de los resultados del PDI se aplican en la clasificación automática del terreno a partir de imágenes satelitales, en el mejoramiento y análisis de imágenes biomédicas, en el uso de la robótica en la industria, en el estudio de las condiciones ambientales y meteorológicas ([3]).

En la Figura 2.17 se observan las etapas del PDI y los niveles en que se dividen los algoritmos del PDI. La primera etapa del PDI es la *captura de la imagen*, la cual se puede llevar a cabo a través de diferentes dispositivos, por ejemplo una cámara (analógica o digital). Si el dispositivo es analógico se debe aplicar el proceso de digitalización para obtener una imagen analógica. Una vez que se ha obtenido la imagen digital, la etapa que sigue es la del *pre-tratamiento* de la imagen, el cual no siempre es necesario llevarlo a cabo. Esta etapa consiste en disminuir el ruido, mejoramiento del contraste, realzar los bordes, entre otros ([3]).

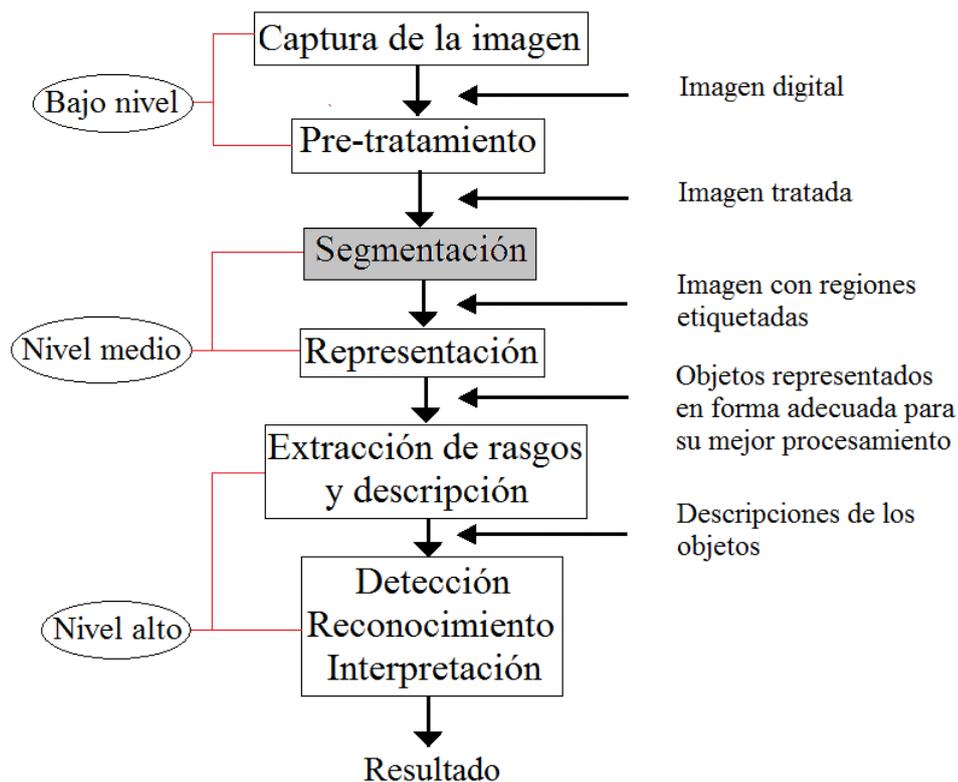


Figura 2.17: Etapas del procesamiento de imágenes. Se puede observar que una de ellas es la segmentación de imágenes. Figura adaptada de [3].

De acuerdo a la Figura 2.17, la tercera etapa es la *segmentación* de la imagen. La segmentación es la etapa crucial del PDI debido a las dificultades que conlleva y a la importancia de sus resultados. Como salida del proceso de segmentación se obtiene una imagen etiquetada. Este proceso se describe con más detalle en la siguiente sección ([3]).

Después de tener el resultado de la segmentación sigue la etapa de *representación*. El objetivo de la representación es, llevar los datos de cada uno de los objetos segmentados a formas en que la computadora pueda trabajar con ellos de manera más apropiada. La quinta etapa es la *Extracción de rasgos y descripción*, la cual se aplica a la imagen segmentada y consiste en capturar las diferencias esenciales entre objetos pertenecientes a diferentes clases ([12]).

Finalmente, se tiene la etapa que consiste en la *detección*, el *reconocimiento* y la *interpretación* de la imagen. La detección consiste en decidir si una instancia de un objeto dado se encuentra presente en una imagen. Por otro lado, el reconocimiento asigna una determinada etiqueta a un objeto a partir de la información proporcionada, por ejemplo, su color, tamaño, ancho, entre otros. Por último, la interpretación determina el significado de un conjunto de objetos previamente conocidos ([3]).

Además, los algoritmos del PDI se pueden dividir en tres niveles ([12]):

1. Bajo nivel: son los que tratan directamente a nivel píxel y abarca las dos primeras

etapas del PDI (la captura y el pre-tratamiento de la imagen).

2. Nivel medio: son aquellos algoritmos que hacen uso de los resultados de bajo nivel para otros medios y abarca la etapa de segmentación y representación.
3. Alto nivel: son los algoritmos que tratan de extraer el significado semántico de la información proporcionada por los niveles más bajos y realizar la toma de decisiones respecto al contenido de las imágenes. Este nivel abarca las etapas de extracción de rasgos, descripción, detección, reconocimiento e interpretación.

2.3.2. Concepto y aplicaciones de la segmentación

En la segmentación de imágenes se desea localizar objetos o encontrar los límites de estos dentro de una imagen. La segmentación es la partición de la imagen digital en grupos de píxeles u objetos (componentes conexas) tal que no se solapan y son homogéneas con respecto a algún criterio, cuya unión cubre a toda la imagen ([3]). La forma de dividir en grupos los píxeles es asignar una etiqueta a cada píxel de la imagen de forma que píxeles con la misma etiqueta tengan características visuales similares, como el color, la intensidad o la textura. Las regiones formadas que son adyacentes resultan diferentes con respecto a la misma característica. Esto da como resultado un conjunto de segmentos que cubre a toda la imagen ([12]). El objetivo de la segmentación es simplificar o cambiar la representación de una imagen en otra más significativa y fácil de analizar ([7]).

Una representación gráfica del proceso de segmentación se muestra la Figura 2.18. El proceso recibe como entrada una imagen en escala de grises que contiene tres objetos, un rombo, un ovalo y el fondo. Como salida se obtiene una imagen etiquetada y dividida en tres objetos que cubren a toda la imagen. Una forma de etiquetar los objetos es considerando su intensidad de gris:

- El objeto con la etiqueta “1” corresponde a los píxeles que conforman al rombo, estos píxeles comparten la característica de tener el mismo tono de gris (negro).
- El segundo objeto, etiquetado con “2”, es el subconjunto de píxeles que forman el ovalo.
- El objeto con la etiqueta “3” corresponden a los píxeles que forman el fondo de la imagen.

Los píxeles de cada objeto comparten la característica de tener la misma intensidad gris, y esta característica es la que los hace diferentes de los píxeles de otro objeto. Este es un ejemplo muy simple, sin embargo, existen aplicaciones donde se encuentran casos en los que no es tan sencillo distinguir los límites de los objetos.

La segmentación es utilizada en diferentes áreas, algunos ejemplos de aplicación son:

- Medicina: La segmentación de vasos sanguíneos en una angiografía retinal¹ permi-

¹angiografía retinal: examen ocular en el que se usa un tinte y una cámara especial para examinar el flujo sanguíneo en la retina

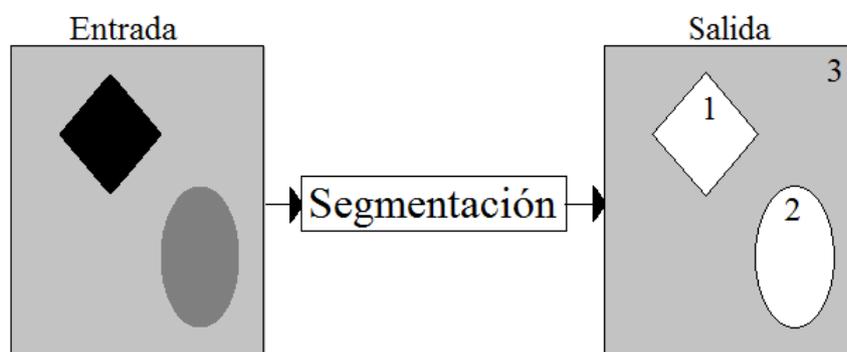


Figura 2.18: Ejemplo del proceso de segmentación de imágenes.

te mejorar la visualización de las ramificaciones y así, los médicos pueden realizar un diagnóstico correcto, como la detección de problemas oculares, la determinación de la circulación adecuada o decidir si un tratamiento está funcionando ([6]).

- Huellas dactilares: Esta es una forma muy común de reconocer a una persona, pero muchas veces las imágenes que se obtienen de las huellas son de muy baja calidad, con daños en la piel o con algo de suciedad. En este caso se requiere segmentar la imagen para obtener las características que proporciona un patrón biométrico formado por las coordenadas espaciales de los puntos característicos de la imagen (minucias) ([19]), ([24]). Esto es muy útil en criminalística, y varias empresas ya están haciendo uso de esto para que solo el cliente pueda ingresar a su cuenta.
- Reconocimiento de caras humanas tanto en imágenes como en vídeos. Actualmente es muy común encontrarse con una cámara con detección de caras o sonrisas ([10]), ([17]).
- En imágenes por satélite, la teledetección de aerosoles es importante en aspectos de contaminación marina, productividad, salud humana o estudios del clima. La segmentación en este caso permite calcular el área ocupada por los aerosoles en las imágenes ([11]).

2.3.3. Clasificación de las técnicas de segmentación

Existen herramientas de segmentación que pueden resultar muy útiles para diversas aplicaciones, sin embargo la segmentación de imágenes no es un problema sencillo, ya que una técnica puede dar buenos resultados para una aplicación específica pero no necesariamente para otra. Las técnicas de segmentación se dividen en tres categorías ([12]):

- Segmentación manual: Técnica que requiere gran intervención del usuario, lo cual resulta lento y expuesto a mayores errores. El usuario selecciona manualmente el contorno del objeto a segmentar mediante herramientas de dibujo.
- Segmentación semiautomática: En esta técnica aún es necesaria la interacción

del usuario. El usuario define las regiones de interés señalando algunos puntos en cada uno de los objetos de la imagen, y los algoritmos se aplican de manera que se elija el camino que mejor se ajusta al borde de cada uno de dichos objetos. La mayoría de las técnicas de segmentación se encuentran en esta categoría, debido a que la técnica manual es muy tediosa, y la automática es muy difícil de lograr. Algunas de las técnicas existentes son: contornos activos, tijeras inteligentes y Grabcut ([15]).

- La segmentación automática es una técnica donde la participación del usuario es casi nula, lo que hace que sea fácil de utilizar, pero utiliza parámetros que muchas veces son difíciles de calcular. En la segmentación automática se pueden encontrar: algoritmos de crecimiento por regiones y watershed ([2]).

2.3.4. Algoritmos de segmentación

Los algoritmos de segmentación de imágenes, generalmente se basan en dos propiedades de intensidad: *discontinuidad y similitud*. Los algoritmos que se basan en la discontinuidad de las intensidades dividen la imagen con base en cambios bruscos de niveles de intensidad; detección de puntos, líneas y bordes son ejemplos de esta categoría. Por otro lado, los algoritmos que se basan en similitud de las intensidades, dividen la imagen en regiones bajo un criterio de similitud, es decir, los píxeles de una misma región deben tener propiedades en común. Algunos ejemplos son: umbralización, región por crecimiento o región por división y fusión. Estos algoritmos hacen uso de una sub-imagen denominada máscara de tamaño 3×3 , definida por la siguiente matriz ([7]):

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} \quad (2.1)$$

donde $w_i \in \mathbb{Z}$, para $i = 1, \dots, 9$. El proceso consiste en realizar una convolución, es decir, una multiplicación elemento con elemento de la máscara (2.1) y la imagen. Esto es, sea V una imagen y la máscara (2.1), la respuesta de la máscara en un píxel $p \in V$, con $p = (x, y)$, está dada por ([12]):

$$\begin{aligned} T &= w_1 \cdot f(x-1, y-1) + w_2 \cdot f(x-1, y) + w_3 \cdot f(x-1, y+1) \\ &+ w_4 \cdot f(x, y-1) + w_5 \cdot f(x, y) + w_6 \cdot f(x, y+1) \\ &+ w_7 \cdot f(x+1, y-1) + w_8 \cdot f(x+1, y) + w_9 \cdot f(x+1, y+1). \end{aligned} \quad (2.2)$$

donde f es la intensidad del píxel. Como se puede ver en la ecuación (2.2), se utiliza la intensidad de los píxeles del conjunto $N_8(p)$. La salida de estos algoritmos es una imagen M del mismo tamaño de V , donde el valor de T es asignado al píxel central de la región de la imagen M . Cuando se realiza una operación de convolución, el resultado no se puede almacenar en la imagen original, ya que esto puede alterar las intensidades originales ocasionando un resultado no deseado. Por eso, se utiliza una imagen distinta donde se almacena la nueva imagen.

Los algoritmos de segmentación que se basan en la propiedad de discontinuidad permiten encontrar puntos aislados, líneas y bordes ([12]):

Detección de puntos aislados: Los puntos aislados son píxeles que tienen una intensidad que difiere drásticamente de la intensidad de sus vecinos. Para detectar puntos aislados se utiliza el Laplaciano que genera las máscaras que se muestran a continuación ([1]):

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

La técnica consiste en aplicar la máscara sobre un píxel q , si el resultado es mayor a un cierto valor u entonces se trata de un punto aislado, esto se describe en la ecuación (2.3).

$$f(q) = \begin{cases} 1, & \text{si } |T| \geq u; \\ 0, & \text{en otro caso.} \end{cases} \quad (2.3)$$

donde $u \in \mathbb{R}^+$ y T está definido por (2.2). A u se le conoce como umbral y su valor depende de la aplicación.

Detección de líneas: Una línea es una secuencia de píxeles en la que dos píxeles consecutivos están conectados, es decir, cada píxel se puede conectar con alguno de sus 8-vecinos, lo que proporciona cuatro direcciones posibles ya sea vertical, horizontal, de 45° o de -45° . Las máscaras utilizadas son ([1]):

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}, \quad \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}, \quad \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}, \quad \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}. \quad (2.4)$$

a) b) c) d)

Los píxeles que forman parte de una de las cuatro líneas (horizontal, vertical o diagonal), tienen mayor respuesta ante su respectiva máscara. Por ejemplo, si los píxeles forman parte de una línea vertical, entonces el valor de T será mayor para la máscara 2.4 b) que para las otras máscaras ([1]).

Detección de bordes: Esta técnica es muy útil cuando las regiones son suficientemente homogéneas y nos sirve para detectar los bordes en una imagen, es decir, la frontera entre dos regiones cuyos tonos de gris difieren significativamente o tienen propiedades diferentes (cambios bruscos de tonos de gris). Los bordes nos proporcionan información sobre dónde están los objetos, su forma, tamaño, y su textura ([3]). Para este método se ocupan los operadores de primera y segunda derivada. La derivada de primer orden produce bordes gruesos, mientras que la

segunda derivada produce bordes más finos y su signo permite determinar si un píxel está del lado oscuro o claro del borde ([12, 3, 7]).

Otros métodos que se basan en buscar fronteras entre regiones, basados en discontinuidades de tonos de gris son: *procesamiento global vía la transformada de Hough* y *vía teoría de Grafos* ([7]).

Procesamiento global vía la transformada de Hough: En esta técnica se usa información global, la cual se puede introducir a través de modelos. El objetivo de este método es enlazar los bordes cuando se encuentran en una curva bien específica como rectas o circunferencias.

Procesamiento global vía técnicas de teoría de grafos: Lo que se busca es un enfoque global de detección de bordes y enlaces basado en la representación de segmentos de aristas en la forma de un grafo. Después se busca el grafo con camino de menor costo. Esta técnica es muy útil en imágenes que presentan ruido. Una de las desventajas es la dificultad y el mayor tiempo de procesamiento que ocupa en comparación con otros métodos.

Una técnica de segmentación que se lleva a cabo por medio de umbrales, basado en la distribución de las propiedades de los píxeles, como sus niveles de gris o de color, es la técnica de umbrales.

Técnica de Umbrales: Supongamos que se tiene una imagen $f(x, y)$ compuesta de objetos claros sobre fondo oscuro de tal forma que tienen niveles de gris agrupados en dos modos predominantes. Una manera de separar los objetos del fondo es seleccionar un umbral T que separe estos modos. Entonces, si un punto (x, y) cumple con la condición

$$f(x, y) > T,$$

es un punto del objeto, en caso contrario es un punto del fondo. Cuando T depende sólo del nivel de gris $f(x, y)$ se denomina umbral global. Si T depende tanto del nivel de gris $f(x, y)$ como de cualquier propiedad local de (x, y) se tiene un umbral local. Además, si T depende de las coordenadas (x, y) , el umbral se denomina dinámico o adaptativo.

Para conseguir regiones homogéneas y conexas en la segmentación de imágenes, se debe considerar la información proporcionada por los píxeles del entorno, ya que los de un mismo entorno suelen tener propiedades estadísticas similares y pertenecer a una misma región. Una segmentación basada en regiones consiste en particionar a la imagen digital en n subregiones tales que deben cumplir las siguientes condiciones ([3]):

1. $\bigcup_{i=1}^n Q_i = Q$.
2. Q_i es una región conexa, $i = 1, \dots, n$.
3. $Q_i \cap Q_j = \emptyset$ para todo i y j , $i \neq j$.

4. $P(Q_i) = Verdadero$ para $i = 1, \dots, n$, donde P es una propiedad de los puntos del conjunto Q_i .
5. $P(Q_i \cup Q_j) = Falso$ para $i \neq j$.

La condición 1 indica que la imagen se divide en n regiones de tal forma que cada píxel debe pertenecer a una región. La condición 2 se refiere a que la región Q_i es m -conexa, para $i = 1, 2, \dots, n$. La condición 3 indica que las regiones deben ser disjuntas, es decir, no hay un elemento en Q_i que también esté en Q_j , si $i \neq j$. La condición 4 determina que todos los píxeles que conforman una región satisfacen una misma propiedad, por ejemplo, comparten la misma intensidad de tono gris. Por último, la condición 5 indica que dos regiones m -adyacentes Q_i y Q_j , con $i \neq j$, son diferentes porque su unión no satisface la propiedad P ([3]).

Las técnicas que se basan en encontrar regiones son las de *crecimiento por regiones* o *división y fusión de regiones*.

Crecimiento por regiones: es un procedimiento que consiste en ir formando grupos o regiones de píxeles por incorporación sucesiva de píxeles a los grupos, basado en un criterio predefinido. El procedimiento básico es empezar con un conjunto de píxeles, llamados *puntos semilla*, de tal forma que a partir de cada semilla se hacen crecer regiones añadiendo nuevos píxeles a dicho conjunto, con la característica de que los píxeles tienen propiedades similares a los puntos semilla. La segmentación dará como resultado tantas regiones como semillas haya. Las dos dificultades que presenta esta técnica son la selección de las semillas y la elección de las propiedades adecuadas que permitan ir añadiendo píxeles durante el proceso de crecimiento.

División y fusión por crecimiento: Un método alternativo al crecimiento por regiones es, subdividir la imagen en un conjunto arbitrario de n regiones disjuntas y posteriormente dividir o fusionar estas regiones con el propósito de que se cumplan las condiciones anteriores 1-5. Los tres pasos a seguir de este método son ([12]):

1. Dividir una región Q_i , para $i = 1, \dots, n$, en sus cuatro cuadrantes disjuntos, tal que $P(Q_i) = Falso$.
2. Fusionar dos regiones adyacentes Q_i y Q_j si $P(Q_i \cup Q_j) = Verdadero$, para $i \neq j$ y $1 \leq i, j \leq n$.
3. Detener el algoritmo cuando no se puedan realizar más divisiones y fusiones.

Una técnica que hace uso de muchos de los conceptos de los métodos anteriores (detección de discontinuidades, umbrales y procesamiento de regiones) es la de *watersheds morfológicas*, que a veces produce buenos resultados en la segmentación. Este método visualiza la imagen en tres dimensiones, dos coordenadas espaciales y su nivel de gris. La imagen en niveles de gris puede tener una representación topográfica de un terreno con tres tipos de puntos ([12]):

1. Puntos mínimos de una región.

2. Puntos ladera, en los cuales si cae agua se dirige hacia un único punto mínimo.
3. Puntos cresta, si cae agua en ellos es probable que llegue a varios mínimos.

Estos tres tipos de puntos forman líneas cresta o líneas watersheds. El objetivo de este método es encontrar las líneas waterseheds.

Por último tenemos a la técnica basada en movimiento, herramienta potente para la segmentación de objetos animados sobre fondos estáticos. En aplicaciones de robótica y de navegación autónoma. El movimiento se plantea a partir de un desplazamiento relativo entre el sistema de detección y la escena que se está viendo ([12]).

Capítulo 3

Conexidad difusa y puntos semilla

Como se explicó en el capítulo anterior, los métodos de segmentación por crecimiento de regiones se basan en que el usuario selecciona algunas partes de los objetos que se distinguen a simple vista en la imagen, y se desea determinar qué puntos pertenecen a cada objeto. En la presente tesis se hace algo similar, se seleccionan algunos píxeles, que se denominan *puntos semilla* y como resultado se obtienen subconjuntos difusos correspondientes a cada objeto. Además, los elementos de cada región deben tener propiedades en común que los hagan distinguirse de los elementos de otras regiones. El grado de conexión entre los elementos de la imagen es una de estas propiedades. El grado de conexión debe ser máximo entre elementos de la misma región y menor entre elementos de distintas regiones. El algoritmo de segmentación propuesto en esta tesis se basa en el Teorema 4, cuyo objetivo es asignar a la imagen una única M -segmentación a partir de afinidades definidas mediante puntos semilla.

Este capítulo se divide en tres secciones. En la primera sección se definen los conceptos de K -conexidad y de afinidad, los cuales ayudarán a encontrar el grado de conexión entre dos píxeles. Estos conceptos también permitirán introducir el concepto de M -segmentación en la segunda sección. A fin de determinar una única M -segmentación, en la tercera sección se define un grafo M -difuso.

3.1. K -conexidad y afinidad difusa

En esta sección se proporciona el concepto de afinidad difusa que nos permite medir la relación entre dos píxeles, pero para esto primero se necesita conocer el concepto de adyacencia difusa.

Definición 21. Una relación difusa ω en \mathbb{Z}^2 que es reflexiva y simétrica se dice que es una adyacencia difusa.

Ejemplo 15. Un ejemplo de una adyacencia difusa se puede definir de la siguiente forma:

Sea $c, d \in \mathbb{Z}^2$ y w la relación difusa dada por:

$$\omega(c, d) = \begin{cases} \frac{1}{1 + \sqrt{\sum_{i=1}^2 (c_i - d_i)^2}}, & \sum_{i=1}^2 |c_i - d_i| \leq 1; \\ 0, & \text{en otro caso.} \end{cases} \quad (3.1)$$

Esta relación difusa es reflexiva ya que:

$$\omega(c, c) = \frac{1}{1 + \sqrt{\sum_{i=1}^2 (c_i - c_i)^2}} = 1.$$

También es simétrica:

$$\omega(c, d) = \frac{1}{1 + \sqrt{\sum_{i=1}^2 (c_i - d_i)^2}} = \frac{1}{1 + \sqrt{\sum_{i=1}^2 (d_i - c_i)^2}} = \omega(d, c).$$

A la pareja (\mathbb{Z}^2, ω) se le denomina espacio digital difuso, si ω es una adyacencia difusa.

Para analizar la teoría que se utiliza en esta tesis debemos representar de modo matemático a una imagen, para esto se proporcionan los conceptos de marco y escenario difuso.

Definición 22. Dados $b, d \in \mathbb{Z}_+^2$, tales que $b = (b_1, b_2)$, $d = (d_1, d_2)$ y $b_j < d_j$, para $j = 1, 2$, el conjunto $V(b, d) = \{c \in \mathbb{Z}_+^2 \mid b_j \leq c_j \leq d_j, \text{ para } j = 1, 2\}$ se le denomina marco con extremos b y d .

En la Figura 3.1 se puede observar el marco $V(b, d)$, donde $b = (1, 1)$ y $d = (3, 3)$. En este caso $V(b, d) = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}$.

Definición 23. Dado un marco $V(b, d)$, un escenario difuso sobre V es la pareja $\zeta = (V, f)$, donde f es un subconjunto difuso de V .

El escenario ζ se dice que es no vacío si existe $c \in V$ tal que $f(c) \neq 0$. De la definición anterior, si el rango de f es $\{0, 1\}$ entonces ζ es un escenario binario. Dicho escenario determina el conjunto donde se definen los píxeles de una imagen digital en escala de grises o binaria, respectivamente (ver Ejemplo 16).

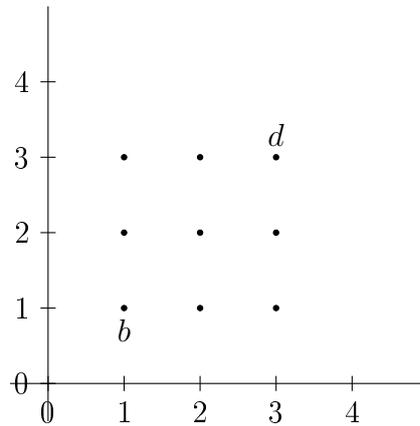


Figura 3.1: Gráfica que representa el marco $V(b, d)$, donde $b = (1, 1)$ y $d = (3, 3)$.

Ejemplo 16. Considere la imagen de la letra “H” de la Figura 3.2(a) y su representación en el plano \mathbb{Z}^2 en la Figura 3.2(b). El marco $V(b, d)$ tiene como extremos a los píxeles $b = (1, 1)$ y $d = (4, 5)$. Se puede definir a g como:

$$g : V \rightarrow D,$$

donde D es el intervalo de las intensidades de los píxeles, es decir, $D = [0, 255]$. Sin embargo, g definida de esta forma no es un subconjunto difuso de V , pero se puede normalizar el intervalo D para obtener el intervalo $L = [0, 1]$, de tal forma que para todo $x \in D$ y $y \in L$:

$$y = \frac{x}{255}. \quad (3.2)$$

Utilizando la ecuación (3.2) se puede construir un subconjunto difuso de V mediante la función f dada por (3.3).

$$f : V \rightarrow L, \quad (3.3)$$

donde $f(x) = \frac{g(x)}{255}$.

Definición 24. Sea $\zeta = (V, f)$ un escenario difuso sobre (\mathbb{Z}^2, ω) . Una afinidad difusa ψ sobre el escenario ζ , es cualquier relación difusa en V que sea reflexiva y simétrica.

Ejemplo 17. Sea ζ el escenario difuso representado en la Figura 3.3. La tabla correspondiente indica los tonos de gris que han sido normalizados utilizando la ecuación (3.2).

Un ejemplo de una afinidad difusa ψ en ζ está dado por:

$$\psi(c, d) = \frac{\omega(c, d)}{1 + 2 | f(c) - f(d) |}, \quad (3.4)$$

donde ω se define por (3.1). Esta afinidad es reflexiva y simétrica ya que $\omega(c, d)$ es reflexiva y simétrica. Haciendo los cálculos correspondientes se obtiene la afinidad difusa

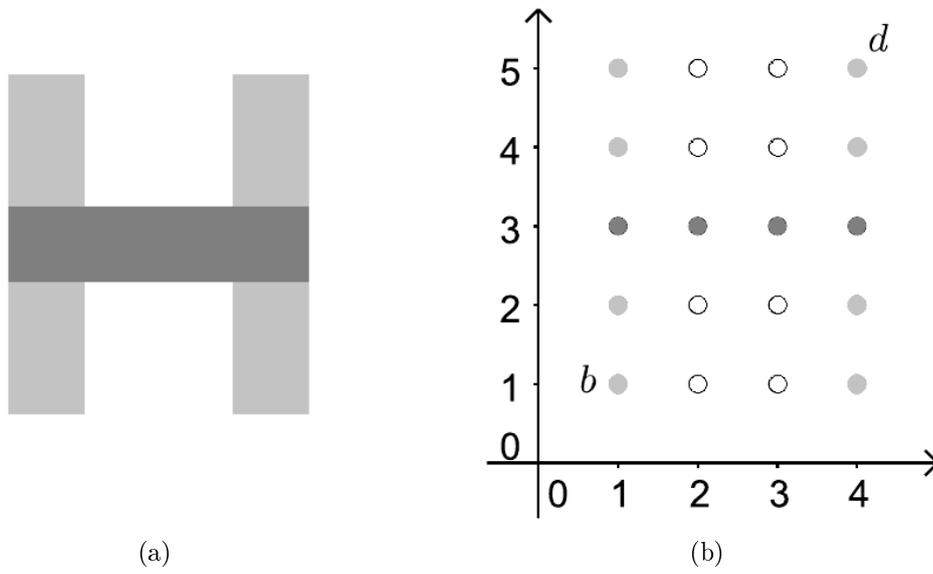


Figura 3.2: Representación de la imagen a) en el plano \mathbb{Z}^2 .

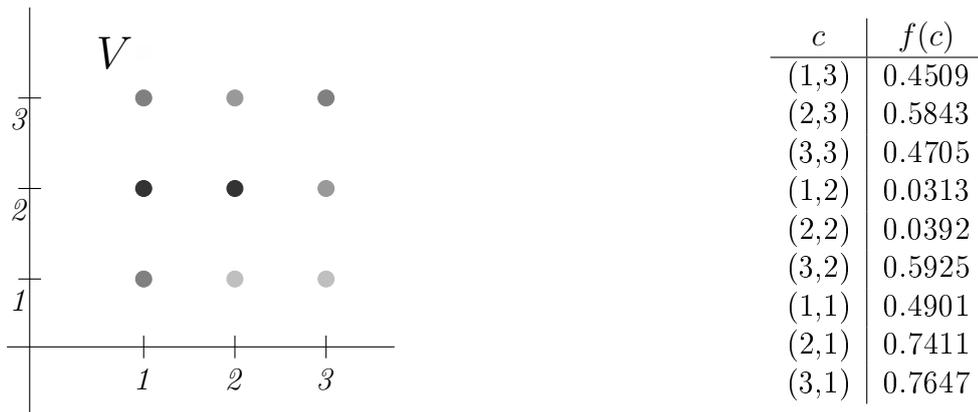


Figura 3.3: Representación gráfica de un escenario difuso sobre V con su respectivo subconjunto difuso f .

que hay entre dos píxeles, como se muestra en la Tabla 3.1.

De la Tabla 3.1 se concluye:

1. La tabla es simétrica debido a que una afinidad difusa es simétrica.
2. Entre más cerca estén dos puntos y sus intensidades sean similares, la afinidad entre ellos es mayor, como se puede ver con los píxeles (2,2) y (1,2) cuya afinidad es 0.4922.
3. La afinidad entre un punto y el mismo es 1, lo cual se observa en la diagonal.
4. Entre más alejados estén los puntos su afinidad es más cercana a cero. Por ejemplo los píxeles (3,1) y (1,3) tienen tonos de gris similares, sin embargo están muy

	(1,3)	(2,3)	(3,3)	(1,2)	(2,2)	(3,2)	(1,1)	(2,1)	(3,1)
(1,3)	1	0.3946	0	0.2718	0	0	0	0	0
(2,3)	0.3946	1	0.4072	0	0.2392	0	0	0	0
(3,3)	0	0.4072	1	0	0	0.4019	0	0	0
(1,2)	0.2718	0	0	1	0.4922	0	0.2607	0	0
(2,2)	0	0.2392	0	0.4922	1	0.2373	0	0.2080	0
(3,2)	0	0	0.4019	0	0.2373	1	0	0	0.3719
(1,1)	0	0	0	0.2607	0	0	1	0.3328	0
(2,1)	0	0	0	0	0.2080	0	0.3328	1	0.4774
(3,1)	0	0	0	0	0	0.3719	0	0.4774	1

Tabla 3.1: Valores de la afinidad difusa definida por (3.4) para cada par de píxeles.

alejados, $\omega(c, d) = 0$, y por lo tanto su afinidad es cero.

- Entre menos similitud haya en sus intensidades, menor es su afinidad. Como ejemplo, los puntos (2,1) y (2,2) son puntos cercanos pero sus tonos de gris no son similares, por lo que $\psi(c, d)$ es pequeño.

La forma de construir una afinidad difusa depende de la aplicación.

Además de definir la afinidad difusa entre dos píxeles, también se pueden construir cadenas de un píxel a otro y definir la fuerza de esta cadena.

Definición 25. Una cadena no vacía entre dos píxeles, $c, d \in V$, es una sucesión ordenada de píxeles $p = \langle c^1, \dots, c^m \rangle$ en V , donde $c^1 = c$ y $c^m = d$. Esta cadena se denota por $p \langle c^1, \dots, c^m \rangle$ y cuando no sea relevante conocer los puntos, se escribirá simplemente p_{cd} .

Una cadena vacía en V es una sucesión que no tiene elementos y se denota por $\langle \rangle$.

Ejemplo 18. La Figura 3.4 representa una cadena del píxel c al píxel d marcado con líneas. Se puede observar que la cadena pasa dos veces por el mismo píxel e , lo cual muestra que los elementos consecutivos de la cadena no necesariamente son distintos y no es la única cadena que se puede formar de c a d .

Definición 26. Sea $\zeta = (V, f)$ un escenario difuso y ψ una afinidad difusa sobre ζ . Dada una cadena $p = \langle c^1, \dots, c^m \rangle$ en V , su fuerza está dada por:

$$\mathcal{N}(p) = \min [\psi(c^1, c^2), \psi(c^2, c^3) \dots, \psi(c^{m-1}, c^m)].$$

Al valor de $\mathcal{N}(p)$ también se le conoce como la fuerza de p . Esta definición nos dice que la fuerza de una cadena es el valor mínimo de las afinidades entre cada par consecutivo de píxeles en la cadena. Esto se puede comparar con una cadena de acero: la fuerza de dicha cadena es la del eslabón más débil.

Definición 27. Dado un escenario difuso $\zeta = (V, f)$, la K -conexidad difusa en ζ ,

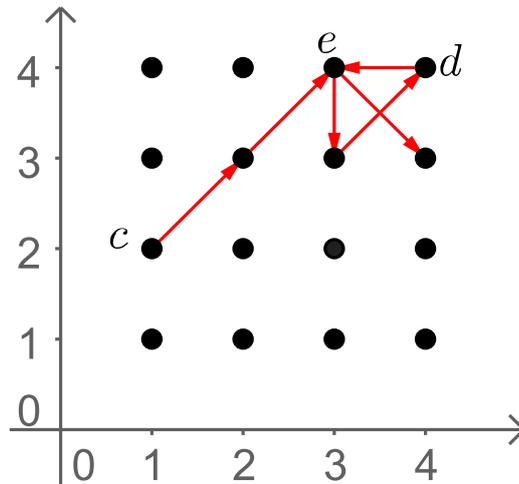


Figura 3.4: Ejemplo de una cadena que comienza en el píxel c y termina con el píxel d .

denotada por K , es la *relación difusa* en V tal que, para todo $c, d \in V$

$$K(c, d) = \max_{p \in P_{cd}} |\mathcal{N}(p)|. \tag{3.5}$$

donde P_{cd} es el conjunto de todas las cadenas de c a d .

A (3.5) también se le conoce como *grado de conexión* entre c y d . Entonces, para conocer el grado de conexión entre c y d se debe calcular la fuerza de todas las cadenas que se pueden formar de c a d , y de ellas elegir la máxima fuerza.

Ejemplo 19. Considere una imagen que solo consta de 4 píxeles (ver Figura 3.5), con sus respectivos tonos de gris. En este ejemplo se desea encontrar el grado de conexión entre los píxeles $c = (1, 1)$ y $d = (2, 2)$.

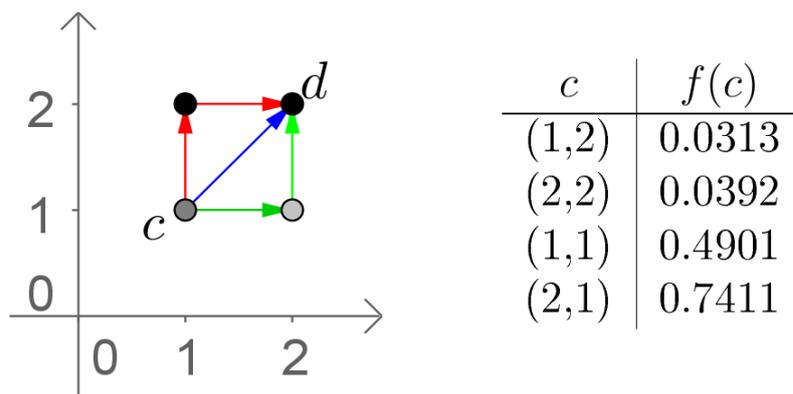


Figura 3.5: Gráfica que muestra algunas cadenas que se pueden formar desde el píxel c al d .

Para calcular el grado de conexión primero se debe calcular la fuerza de cada cadena de c a d . En la Tabla 3.2 se muestra la afinidad entre cada par de puntos para calcular la fuerza de las cadenas. Para esto se usó la afinidad difusa descrita en (3.4).

	(1,1)	(2,1)	(1,2)	(2,2)
(1,1)	1	0.3328	0.2607	0
(2,1)	0.3328	1	0	0.2080
(1,2)	0.2607	0	1	0.4922
(2,2)	0	0.2080	0.4922	1

Tabla 3.2: Afinidad difusa (3.4) que hay entre cada par de puntos de la Figura 3.5.

Una vez que se obtienen las afinidades entre los pares de puntos, se encuentra la fuerza de cada cadena. La Tabla 3.3 contiene la siguiente información:

1. En la primera columna se describen las cadenas que van de c a d .
2. Los valores de la segunda columna representan la fuerza de cada cadena.
3. El valor de la tercera columna es el grado de conexión entre c y d , es decir, el máximo valor de la segunda columna.

cadena	$\mathcal{N}(p)$	$\psi(c, d)$
$\langle(1, 1), (2, 2)\rangle$	0	
$\langle(1, 1), (1, 2), (2, 2)\rangle$	0.2607	
$\langle(1, 1), (2, 1), (2, 2)\rangle$	0.2080	0.2607
$\langle(1, 1), (1, 2), (2, 1), (2, 2)\rangle$	0	
$\langle(1, 1), (2, 1), (1, 2), (2, 2)\rangle$	0	

Tabla 3.3: Tabla que muestra la fuerza de las cadenas de la Figura 3.5.

Cabe mencionar que para calcular la K -conexidad teniendo sólo cuatro píxeles en el espacio, se encontraron 5 cadenas de c a d , pero si se agregan más píxeles, el número de cadenas aumenta considerablemente y el cálculo del grado de conexión entre dos píxeles se vuelve muy costoso. Se debe tener cuidado, ya que las cadenas pueden pasar varias veces por píxeles que ya están en la cadena, por lo que estas cadenas son descartadas.

3.2. M -segmentación y M -semisegmentación

El algoritmo de segmentación que se presenta en esta tesis requiere que el usuario indique el número de objetos M en los que se va a segmentar la imagen y cuales son algunos puntos iniciales de cada uno de los objetos de la misma.

Definición 28. Sea $\zeta = (V, f)$ un escenario difuso y $M \in \mathbb{Z}_+$. Para cada m , $1 \leq m \leq M$, sea $S_m \subseteq V$ con $S_i \cap S_j = \emptyset$ si $i \neq j$. A los elementos de S_m se les llama punto semilla del objeto m .

Los píxeles $v \in S_m$ sirven como puntos de comienzo del crecimiento por regiones para el objeto m . Estos puntos son “sembrados” por el usuario, es decir, el usuario decide que puntos serán semillas.

Definición 29. Sea $\zeta = (V, f)$ un escenario difuso y $\{S_m\}_{m=1}^M$ una colección de puntos semilla. Una segmentación difusa de ζ con puntos semillas $\{S_m\}$ es una colección $\{V_m\}_{m=1}^M$ de subconjuntos difusos de V tales que: $V_m(v) = 1$, si $v \in S_m$.

Antes de proponer la M -segmentación que se quiere construir en esta tesis primero se proporciona el concepto de una segmentación difusa.

Definición 30. Sea M un entero positivo, $\zeta = (V, f)$ un escenario difuso y $\{V_m\}_{m=1}^M$ una segmentación difusa. Una función $\sigma : V \rightarrow \mathbb{R}^{M+1}$ es una M -semisegmentación de ζ con puntos semilla $\{S_m\}_{m=1}^M$ si:

1. Para todo $v \in V$, $\sigma(v) = (\sigma_0^v, \sigma_1^v, \dots, \sigma_M^v)$, con $\sigma_i^v = V_i(v)$, $i = 1, \dots, M$.
2. Para algún m , $1 \leq m \leq M$, $\sigma_0^v = \sigma_m^v$.
3. Para cualquier k , $1 \leq k \leq M$, es $\sigma_k^v = 0$ o $\sigma_k^v = \sigma_0^v$.

A σ se le conoce como una M -segmentación si para cada v , σ_0^v es positivo.

El objetivo de la M -segmentación es asociar un conjunto difuso a cada objeto de la imagen. De la definición anterior, la imagen a segmentar se le denota por ζ , a cada píxel $v \in V$ le corresponde un vector $\sigma(v)$. Al ejecutar el algoritmo de segmentación que se presenta en esta tesis, el usuario debe seleccionar un conjunto de puntos semilla S_m por cada objeto de la M -segmentación, el total de estos conjuntos es representado por M , con $1 \leq m \leq M$. El total de conjuntos de puntos semilla, M , depende de las regiones en las que se desea segmentar la imagen. El algoritmo proporciona como salida los M conjuntos difusos $\{V_m\}_{m=1}^M$ en que se segmenta a la imagen. El valor de σ_0^v indica el grado de pertenencia del píxel v al conjunto difuso V_m , donde el subíndice m es del elemento $\sigma_m^v \neq 0$, para $v \in V$ y $1 \leq m \leq M$. Por ejemplo, si el usuario seleccionó 5 conjuntos de puntos semilla en la imagen, entonces $M = 5$ y la imagen se segmenta en 5 conjuntos difusos. Por lo que, el tamaño del vector asociado de la M -segmentación a cada píxel, $\sigma(v)$, es de 6. Además, si suponemos que para un $c \in V$, $\sigma(c) = (0.83, 0, 0, 0, 0.83, 0)$ entonces el píxel c pertenece al conjunto difuso V_4 con un grado de pertenencia de 0.83, ya que $\sigma_0^c = 0.83$ y $\sigma_4^c \neq 0$.

La segmentación debe construirse de tal manera que tome en cuenta el número de objetos especificados, el tipo de l -adyacencia considerado, para $l = 4$ u 8 , la intensidad de gris y la afinidad definida en ella. En el Ejemplo 20 se ilustra una M -segmentación suponiendo que se conocen los valores de la función de afinidad para cada píxel.

Ejemplo 20. Considere el conjunto V formado por 4 subconjuntos: T , L , R y B (ver Figura 3.6). El conjunto T consta de los tres puntos verticales de la parte superior de la gráfica. Los 5 puntos horizontales que están a la izquierda forman al conjunto L ; R consta de los 3 puntos horizontales que están a la derecha y B de los tres puntos verticales de la parte inferior. En este ejemplo se desea encontrar una M -segmentación para el escenario $\zeta = (V, f)$ donde el marco está dado por $V(b, d)$ con $b = (1, 1)$, $d = (9, 9)$ y f está dada por (3.6).

$$f(v) = \begin{cases} 1, & \text{si } v \in L \cup R \cup T \cup B; \\ 0, & \text{en otro caso.} \end{cases} \quad (3.6)$$

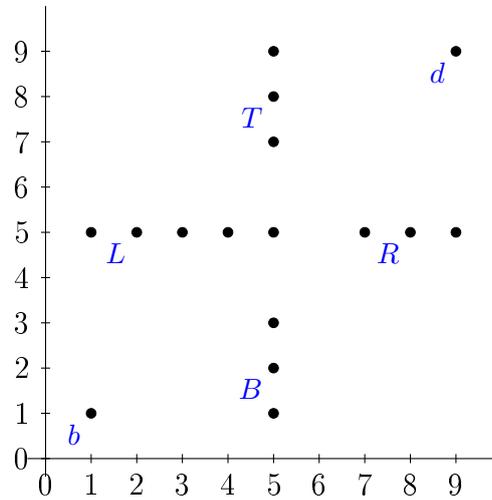


Figura 3.6: Conjunto de puntos V para construir una M -segmentación.

Supongamos que los dos conjuntos de puntos semilla que selecciona el usuario son $S_1 = T$ y $S_2 = L$, entonces $M = 2$. Ya que se tienen S_1 y S_2 , una M -segmentación de ζ consta de dos conjuntos difusos V_1 y V_2 . Dado que σ_0^v representa el grado de pertenencia de v a V_m , para $1 \leq m \leq M$, entonces para $v \in T$, $\sigma_0^v = 1 = \sigma_1^v$ mientras que para $v \in L$ se cumple que $\sigma_0^v = 1 = \sigma_2^v$. Luego,

para todo $v \in T$, $\sigma(v) = (1, 1, 0)$ y para todo $v \in L$, $\sigma(v) = (1, 0, 1)$.

Supongamos que para $v \in B$, $V_1(v) = 0.5$ y $V_2(v) = 0.4$. Para construir σ^v se puede escoger, por ejemplo, el máximo valor de $V_m(v)$, para $1 \leq m \leq M$, entonces

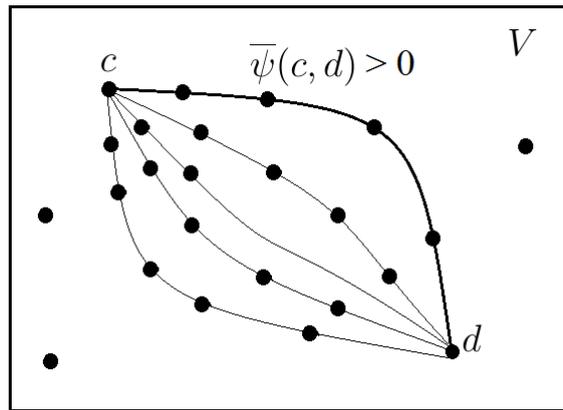
para $v \in B$, $\sigma(v) = (0.5, 0.5, 0)$.

Por otro lado, supongamos que para $v \in R$, $V_1(v) = 0.3$ y $V_2(v) = 0.7$, utilizando el razonamiento anterior, el vector $\sigma(v)$ se puede definir como:

para $v \in R$, $\sigma(v) = (0.7, 0, 0.7)$.

Con esto queda definida una M -segmentación a partir de los puntos semilla dados.

Observemos que en el ejemplo anterior no se utilizó el valor de las intensidades de los píxeles, solo se consideró su posición en el escenario. Los valores del vector σ solo son para poder ejemplificar la forma de definir una M -segmentación. Para un caso real, se encuentra con la dificultad de construir una M -segmentación que se relacione de forma apropiada con los tonos de gris y con la posición de los píxeles. Hay que tener en cuenta que la M -segmentación permitirá construir los M conjuntos difusos que se desean y cada valor representará el grado de pertenencia de cada píxel al correspondiente objeto.

Figura 3.7: Representación de un grafo M -difuso.

3.3. Grafo M -difuso

Una vez que se tiene la definición de una M -segmentación, se necesita una estructura matemática que permita representar a la imagen tomando en cuenta la posición de los píxeles y las intensidades de los tonos de gris, así como el número de objetos en los que se va segmentar la imagen.

Definición 31. Sea $\zeta = (V, f)$ un escenario difuso y ψ una afinidad del conjunto V . Se dice que V es ψ -conexo, si para cualquier par de píxeles c y d en V , existe una cadena $\langle c_1, c_2, \dots, c_k \rangle$ en V , con fuerza positiva, donde $c_1 = c$ y $c_k = d$.

Es claro que el conjunto V definido en el Ejemplo 19 es ψ -conexo, ya que existen cadenas con fuerza positiva entre dos píxeles cualesquiera de V (ver Tabla 3.3).

Para construir una M -segmentación de una imagen, a cada conjunto de puntos semilla se le asociará una afinidad difusa, que permita determinar el grado de pertenencia de cada píxel al objeto correspondiente.

Definición 32. Un grafo M -difuso es una pareja (V, Ψ) , donde $V \neq \emptyset$ finito y $\Psi = (\psi_1, \dots, \psi_M)$, con ψ_m una afinidad en V , $1 \leq m \leq M$, tal que V es $\bar{\psi}$ -conexo, donde

$$\bar{\psi}(c, d) = \min_{1 \leq m \leq M} \psi_m(c, d). \quad (3.7)$$

Esto es, un grafo M -difuso es una pareja que consta de un conjunto V y M afinidades tal que para cualquier par de píxeles c y d exista una cadena p_{cd} con fuerza positiva con respecto a $\bar{\psi}$. Con esta definición se asegura que todas las cadenas que se forman en V tienen fuerza positiva (ver Figura 3.7).

Ejemplo 21. Para dar un ejemplo de un grafo M -difuso, considere las siguientes tres afinidades difusas, donde $\| \cdot \|$ es la norma euclidiana.

$$\psi(c, d) = \begin{cases} 0, & \text{si } c = d; \\ \frac{1}{\|c - d\|}, & \text{en otro caso.} \end{cases} \quad (3.8)$$

$$\tilde{\psi}(c, d) = \begin{cases} \psi(c, d), & \text{si } \|c - d\| \leq 3; \\ 0, & \text{en otro caso.} \end{cases} \quad (3.9)$$

$$\widehat{\psi}(c, d) = \begin{cases} \frac{1}{3}, & \text{si } \|c - d\| \leq 4; \\ 0, & \text{en otro caso.} \end{cases} \quad (3.10)$$

Las funciones dadas en (3.8), (3.9) y (3.10) son afinidades difusas porque son relaciones difusas cuyo dominio es $V \times V$ y su rango es el intervalo $[0, 1]$. También son relaciones simétricas porque $\| \cdot \|$ es una operación simétrica, es decir:

$$\|a - b\| = \|b - a\|.$$

Sea el conjunto V que consta de 5 subconjuntos T , L , B , R y O como se muestra en la Figura 3.8. Se puede probar que la pareja $(V, (\psi_1, \psi_2))$, donde $\psi_1 = \psi$ y $\psi_2 = \widehat{\psi}$, es un grafo 2-difuso. Para esto, se debe probar que para cualquier par de puntos existe una cadena con fuerza positiva. En efecto, considere los puntos $c = (5, 9)$, $d = (1, 5)$, $o = (9, 1)$, $u = (5, 1)$, y $c' = (5, 5)$. La distancia máxima es de c a o o de d a o , pero existe al menos una cadena dada por $p_{c,o} = \langle c, c', u, o \rangle$ o $p_{d,o} = \langle d, c', u, o \rangle$ tal que la fuerza de dichas cadenas está dada por:

$$\mathcal{N}(p_{c,o}) = \frac{1}{4} \text{ y } \mathcal{N}(p_{d,o}) = \frac{1}{4}, \text{ con } \bar{\psi} \text{ definido en (3.7).}$$

Por lo que $(V, (\psi_1, \psi_2))$ es un grafo 2-difuso. Por otro lado, si $\psi_1 = \tilde{\psi}$ o $\psi_2 = \tilde{\psi}$ entonces $(V, (\psi_1, \psi_2))$ no es un grafo 2-difuso, debido a que la distancia entre el punto más cercano a o es mayor a 3, es decir, $\|u - o\| > 3$ y por tanto la fuerza de las cadenas que tengan como elemento al punto o es 0.

Como se mencionó anteriormente, M indica el número de conjuntos semillas que el usuario selecciona, y ψ_i se construye a partir de los puntos semilla S_i tomando en cuenta sus intensidades de tonos de gris, para $i = 1, 2, \dots, M$.

La siguiente definición nos permite relacionar una cadena con el concepto de una M -segmentación. Esta definición es la que falta para poder enunciar el teorema que nos permite construir la M -segmentación presentada en esta tesis.

Definición 33. Sea M entero positivo, $\zeta = (V, f)$ un escenario difuso, σ una M -segmentación de V y $1 \leq m \leq M$. La cadena $\langle c^{(0)}, c^{(1)}, \dots, c^{(K)} \rangle$ se dice que es una σm -cadena si $\sigma_m^{c^{(k)}} > 0$, para todo $0 \leq k \leq K$.

Como se mencionó anteriormente, el valor de $\sigma_m^{c^{(k)}}$ indica el grado de pertenencia de $c^{(k)}$ al objeto V_m , para $1 \leq m \leq M$ y todo $1 \leq k \leq K$. Se pretende que el grado de pertenencia de cada elemento $c^{(k)}$ sea positivo, con respecto al conjunto difuso V_m . En particular, es importante considerar las cadenas que empiecen en algún punto semilla como se puede observar en la Figura 3.9.

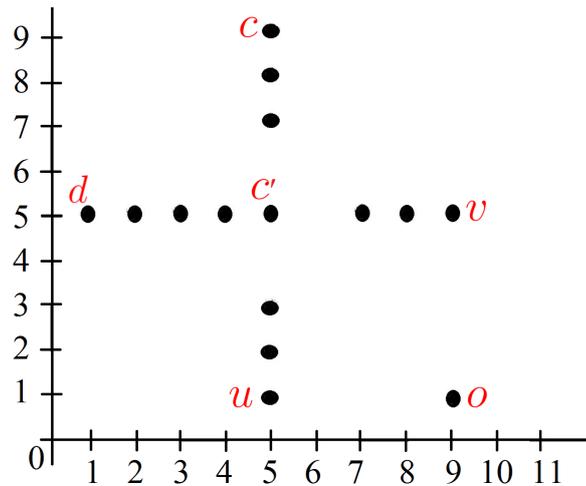
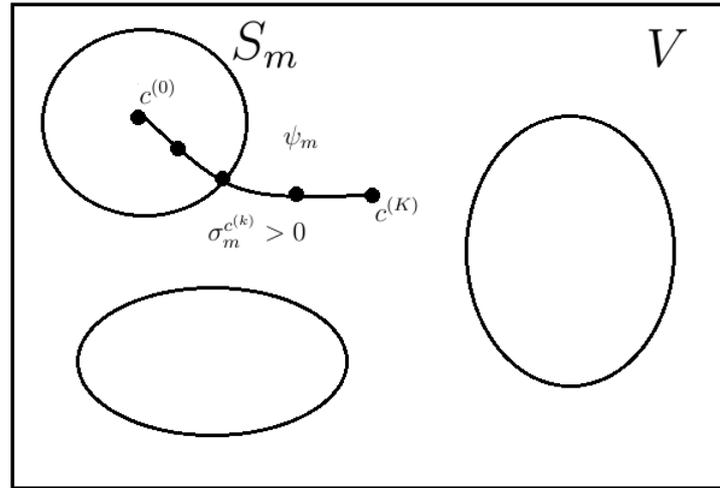
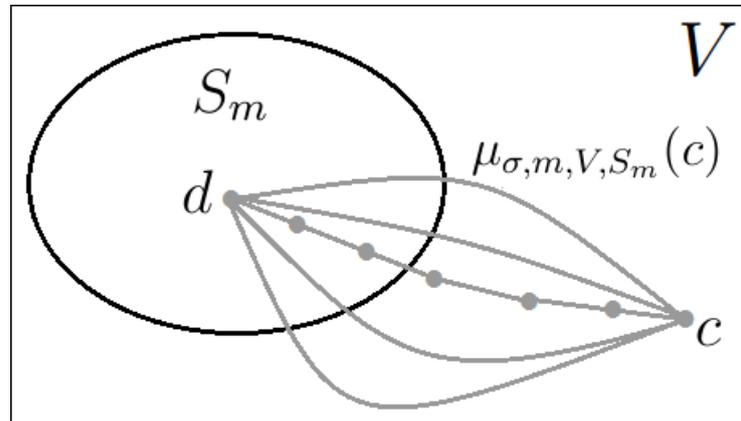


Figura 3.8: El conjunto V consta de 5 subconjuntos: T, L, B, R, O . Los tres puntos de la parte superior representan al conjunto T . El conjunto L consta de los 5 puntos horizontales que están a la izquierda en la gráfica. Los 3 puntos horizontales en la derecha representan el conjunto R . B consta de los tres puntos verticales que están debajo, y el punto que se encuentra aislado forma el subconjunto O .

Para $S_m \subset V$ y $c \in V$, $\mu_{\sigma, m, V, S_m}(c)$ denota la *máxima* ψ_m -fuerza de todas las σm -cadenas en V , de un punto $d \in S_m$ a c . Dicho valor es cero si no existe tal cadena. Esto es, suponga que la M -segmentación asocia en M conjuntos difusos a la imagen. Considere un conjunto de puntos semilla S_m , para $1 \leq m \leq M$, y a los píxeles $d \in S_m$ y $c \in V$ como se observa en la Figura 3.10. Para calcular el valor de $\mu_{\sigma, m, V, S_m}(c)$, se forman todas las σm -cadenas de d a c , es decir, todas las cadenas p_{cd} tal que el grado de pertenencia del píxel c al conjunto difuso V_m sea positivo ($\sigma_m^c > 0$). Se mide la fuerza de todas las σm -cadenas de d a c , y se escoge la máxima fuerza de estas, por cada conjunto difuso V_m que existe. En una imagen que tiene miles de píxeles esto resulta una tarea difícil porque se puede formar una cantidad considerablemente grande de σm -cadenas para un conjunto difuso V_m , $1 \leq m \leq M$.

En la Figura 3.11 se ilustra la importancia de usar $\mu_{\sigma, m, V, S_m}(c)$, ya que si solo se calcula $\psi(c, v)$ este valor puede ser muy pequeño, dado que el tono de gris de c y v no son el mismo. Lo que conviene es buscar un punto d que no muestre mucha diferencia entre el tono de gris de c y v . Esto se consigue calculando la máxima fuerza de las cadenas p_{cv} , es decir, $\mu_{\sigma, m, V, S_m}(c)$.

Ejemplo 22. Considere el conjunto V que se define en la Figura 3.8 y las afinidades $\Psi = (\hat{\psi}, \psi)$, definidas en (3.10) y (3.8), respectivamente. Suponga una M -segmentación σ para V como (3.11), construida a partir de los puntos semilla: $S_1 = T$ y $S_2 = L$, entonces $M = 2$, $\psi_1 = \hat{\psi}$ y $\psi_2 = \psi$ son las afinidades de S_1 y S_2 , respectivamente. Se define

Figura 3.9: Representación de una σm -cadena.Figura 3.10: Cada línea gris representa una σm -cadena que va de un píxel en $d \in S_m$ al píxel $c \in V$, para $1 \leq m \leq M$.

$$\sigma^c = \begin{cases} (1, 1, 0), & \text{si } c \in S_1; \\ (1, 0, 1), & \text{si } c \in S_2; \\ (1/\sqrt{8}, 1/\sqrt{8}, 0), & \text{si } c \in R; \\ (1/\sqrt{8}, 1/\sqrt{8}, 1/\sqrt{8}), & \text{si } c \in B; \\ (1/4, 1/4, 1/4), & \text{si } c \in O. \end{cases} \quad (3.11)$$

Como $\sigma_1^c = 0$ para $c \in S_2$ entonces no se puede formar una $\sigma 1$ -cadena de $c \in S_2$ a $o \in O$. Además, no se puede formar una $\sigma 2$ -cadena de $c \in S_1$ a $o \in O$, porque $\sigma_2^c = 0$ para $c \in S_1$. Así, la máxima ψ_m -fuerza, para $1 \leq m \leq M$, de estas cadenas, está dada por:

$$\mu_{\sigma, 1, V, S_1}(o) = 0 = \mu_{\sigma, 2, V, S_2}(o).$$

Sin embargo, para $c \in S_1$ y $o \in O$, existe una $\sigma 1$ -cadena de c a o y en particular está contenida en el subconjunto $S_1 \cup B \cup O$, ya que $\sigma_1^c > 0$. También, existe una $\sigma 1$ -cadena

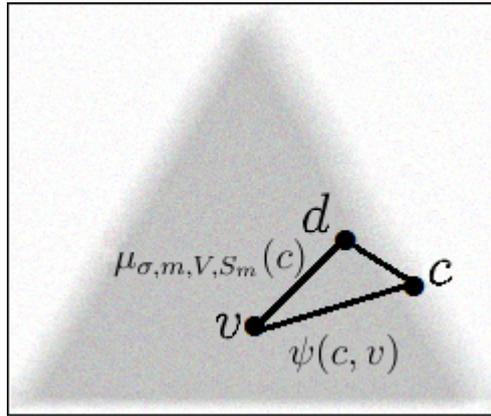


Figura 3.11: Representación de $\mu_{\sigma, m, V, S_m}(c)$ en una imagen.

de c a o pero contenida en $S_1 \cup R \cup O$. Haciendo los cálculos correspondientes con ψ_1 se tiene,

$$\mu_{\sigma, 1, S_1 \cup B \cup O, S_1}(o) = \mu_{\sigma, 1, S_1 \cup R \cup O, S_1}(o) = \frac{1}{3}.$$

Por otro lado, se puede formar una σ 2-cadena de $c \in S_2$ a o contenida en $S_2 \cup B \cup O$ cuya máxima ψ_2 -fuerza está dada por:

$$\mu_{\sigma, 2, S_2 \cup B \cup O, S_2}(o) = \frac{1}{4}.$$

No se puede formar una σ 2-cadena que esté contenida en particular en $S_2 \cup R \cup O$ porque $\sigma_2^c = 0$ para $c \in R$, pero se puede formar si la σ 2-cadena está contenida en $S_2 \cup O$, entonces

$$\mu_{\sigma, 2, S_2 \cup O, S_2}(o) = \frac{1}{\sqrt{32}}.$$

Se puede observar en la Figura 3.8 que en el subconjunto $S_1 \cup B \cup O$, el punto más cercano a o es el punto $u \in B$ y la distancia entre ellos es $\frac{1}{4}$. Por otro lado, para el caso en que se tiene el subconjunto $S_2 \cup O$, el punto más cercano a o es el punto c' con una distancia igual a $\frac{1}{\sqrt{32}}$.

El siguiente teorema indica la unicidad de una M -segmentación especial, construida a partir de las afinidades de los objetos seleccionados por el usuario.

Teorema 4. Si (V, ψ) es un grafo M -difuso y $S_m \subset V$, para toda $1 \leq m \leq M$, es tal que al menos uno de estos subconjuntos es no vacío, entonces existe a lo más una M -segmentación σ de V con la siguiente propiedad:

Para cada $c \in V$, si para $1 \leq n \leq M$

$$s_n^c = \begin{cases} 1, & \text{si } c \in S_n; \\ \max_{d \in V} (\min(\mu_{\sigma, n, V, S_n}(d), \psi_n(d, c))), & \text{en otro caso;} \end{cases} \quad (3.12)$$

entonces para $1 \leq m \leq M$

$$\sigma_m^c = \begin{cases} s_m^c, & \text{si } s_m^c \geq s_n^c \text{ para } 1 \leq n \leq M; \\ 0, & \text{en otro caso.} \end{cases} \quad (3.13)$$

Demostración. Por demostrar que la M -segmentación σ es única. Suponga que existen dos M -semisegmentaciones de V , σ y τ tal que $\sigma \neq \tau$. Elija $c \in V$ que cumpla

$$\sigma^c \neq \tau^c \quad (3.14)$$

y para todo $d \in V$,

$$\max(\sigma_0^d, \tau_0^d) > \max(\sigma_0^c, \tau_0^c), \quad (3.15)$$

$$\sigma^d = \tau^d. \quad (3.16)$$

Sin pérdida de generalidad suponga que:

$$\sigma_0^c \geq \tau_0^c, \quad (3.17)$$

entonces $\sigma_m^c > \tau_m^c$ para algún $m \in \{1, \dots, M\}$, por la ecuación (3.13) se cumple que $\sigma_m^c = s_m^c$ y $c \notin S_m$. Luego, por el Teorema 4 existe una σm -cadena $\langle d^0, \dots, d^L \rangle \in V$ con ψ_m -fuerza no menor que σ_m^c tal que

$$d^0 \in S_m \text{ y } \psi_m(d^L, c) \geq \sigma_m^c.$$

Por inducción se puede probar que $\langle d^0, \dots, d^L \rangle$ es una τm -cadena para todo l , $0 \leq l \leq L$, es decir, se debe probar que $\tau_m^{d^l} > 0$. Dado que $d^0 \in S_m$ y por (3.12) entonces se cumple la condición. Supongamos que es válido para $l-1$, es decir, $\langle d^0, \dots, d^{l-1} \rangle$ es una τm -cadena en V , para $1 \leq l \leq L$. Dado que $\langle d^0, \dots, d^{l-1} \rangle$ es una σm -cadena con ψ_m -fuerza de al menos σ_m^c entonces

$$\mu_{\tau, m, V, S_m}(d^{l-1}) \geq \sigma_m^c. \quad (3.18)$$

Además, por (3.12) se cumple que

$$\psi(d^{l-1}, d^l) \geq \sigma_m^c. \quad (3.19)$$

Luego, por las ecuaciones (3.18) y (3.19) se tiene que

$$t_m^{d^l} \geq \sigma_m^c, \quad (3.20)$$

donde t está definido para τ de forma similar a s para σ . Se quiere probar que $\tau_m^{d^l} \neq 0$, para esto supongamos que $\tau_m^{d^l} = 0$, entonces existe un $n \in \{1, \dots, M\}$ tal que

$$t_n^{d^l} > t_m^{d^l}.$$

Pero,

$$t_n^{d^l} = \tau_n^{d^l} = \tau_0^{d^l} \leq \max(\sigma_0^{d^l}, \tau_0^{d^l}). \quad (3.21)$$

Por otro lado,

$$t_m^{d^l} \geq \sigma_m^c = \sigma_0^c = \max(\sigma_0^c, \tau_0^c). \quad (3.22)$$

Por (3.21) y (3.22) se tiene que

$$\max(\sigma_0^c, \tau_0^c) \leq \max(\sigma_0^{d^l}, \tau_0^{d^l}). \quad (3.23)$$

Por la forma en que se eligió c en (3.16) y (3.23) se deduce que $\tau^{d^l} = \sigma^{d^l}$, lo cual no puede ser ya que $\langle d^0, \dots, d^L \rangle$ es una σm -cadena, es decir, $\sigma_m^{d^l} \neq 0$. Con esta parte se concluye que $\tau_m^{d^l} \neq 0$, es decir, la cadena también es una τm -cadena.

Ahora, reuniendo todas las desigualdades obtenidas hasta el momento:

$$\tau_0^c \geq t_m^c \quad (\text{por (3.12)}) \quad (3.24)$$

$$\geq \sigma_m^c \quad (\text{por (3.20)}) \quad (3.25)$$

$$= \sigma_0^c \quad (\text{por ser } \sigma \text{ una } M\text{-semisegmentación}) \quad (3.26)$$

$$\geq \tau_0^c \quad (\text{por (3.17)}). \quad (3.27)$$

Por lo tanto

$$\sigma_m^c = t_m^c = \tau_m^c;$$

lo que contradice la suposición de que $\sigma_m^c > \tau_m^c$. Así $\sigma = \tau$. Con esto se prueba la unicidad con respecto a M -semisegmentación.

Resta probar que cualquier M -semisegmentación definida por (3.13) es una M -segmentación. Sea $c \in V$; por (3.13) se deduce que

$$\sigma_0^c = \max_{1 \leq m \leq M} s_m^c.$$

Sea $\langle c^0, \dots, c^K \rangle$ con $(\min_{1 \leq m \leq M} \psi_m)$ -fuerza positiva, donde $c^0 \in S_m$. Por inducción se prueba que dicha cadena es una σm -cadena. Para $k = 0$, es claro que $\sigma_0^{c^k} > 0$. Suponga que para $k - 1$ se cumple la condición, para $1 \leq k \leq K$. Eligiendo algún $m \in \{1, \dots, M\}$ tal que

$$\sigma_0^{c^{k-1}} = \sigma_m^{c^{k-1}} = s_m^{c^{k-1}},$$

entonces, existe una σm -cadena de ψ_m -fuerza positiva de un elemento en S_m a c^{k-1} , y dado que $\psi_m(c^{k-1}, c^k) > 0$, se concluye que

$$\sigma_0^{c^k} \geq s_m^{c^k} > 0.$$

Así, $\langle c^0, \dots, c^K \rangle$ es una σm -cadena y por lo tanto σ es una M -segmentación. \square

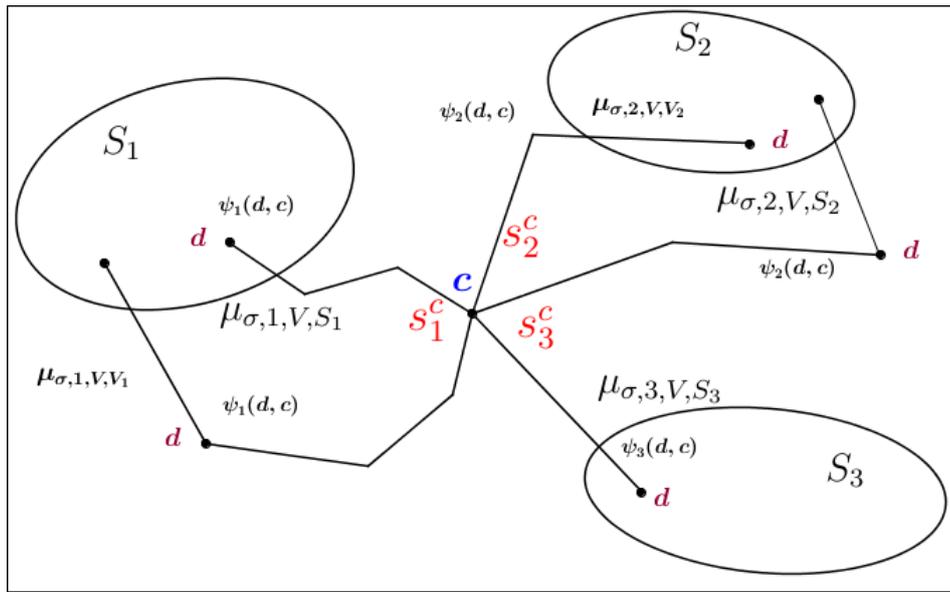


Figura 3.12: Representación gráfica del Teorema 4. Cada conexión a d representa una σm -cadena que comienza en un elemento de S_n , para $1 \leq n \leq M$. La conexión de d a c representa a $\psi_n(c, d)$.

La idea para construir la M -segmentación a partir de las afinidades es la siguiente: si el píxel c es un punto semilla, es decir, $c \in S_m$, entonces es claro que pertenece totalmente al conjunto difuso V_m , para $1 \leq m \leq M$. Si el píxel c no es un punto semilla, entonces se construyen todos los caminos posibles de c a $u \in S_m$, para $1 \leq m \leq M$, y se escoge el mejor (ver la ecuación 3.12). Cada uno de los caminos se divide en dos subcaminos, la σm -cadena que empieza en algún elemento en $u \in S_m$ a $d \in V$, y el camino que se forma de c a d (ver la Figura 3.12). La fuerza del camino está dado por el valor mínimo entre la fuerza mínima de la σm -cadena y la afinidad entre c y d . El mejor camino es el que tenga la máxima fuerza, con lo cual se obtiene el valor denotado por s_m^c . El valor de s_m^c corresponde a cada conjunto semilla S_m . Para saber a qué objeto corresponde el píxel c , se escoge el valor más grande de s_m^c , para $1 \leq m \leq M$.

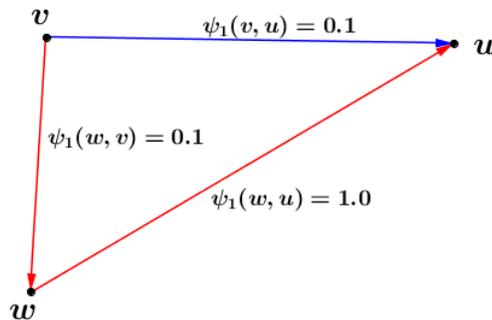


Figura 3.13: Conjunto $V = \{u, v, w\}$, donde $V_1 = \{v\}$ y ψ_1 está definida por la función dada en (3.28)

La forma en que se relacionan la M -segmentación y las afinidades permite garantizar

la unicidad. Se puede pensar que es innecesario utilizar μ_{σ,n,V,S_n} , y que resulta más sencillo remplazarlo por σ_n^d . Sin embargo, este cambio no asegura la unicidad de la segmentación. Como ejemplo considere el conjunto V que consta de 3 puntos, u, v, w , donde $S_1 = \{v\}$ es el único objeto seleccionado (ver la Figura 3.13). Considere el grafo 1-difuso $(V, \{\psi_1\})$ donde

$$\psi_1(c, d) = \begin{cases} 0.1, & \text{si } v \in \{c, d\}; \\ 1.0, & \text{si } v \notin \{c, d\}. \end{cases} \quad (3.28)$$

Con la siguiente 1-segmentación (a modo de ejemplo),

$$\sigma^v = (1, 1) \quad \text{y} \quad \sigma^u = \sigma^w = (r, r).$$

Sólo hay un único valor de r que hace que la 1-segmentación satisfaga las propiedades del Teorema 4, $r=0.1$. Para σ^u se puede formar una σ 1-cadena de v a w , y otra de v a v , con

$$\mu_{\sigma,1,V,S_1}(w) = 0.1 \quad \text{y} \quad \mu_{\sigma,1,V,S_1}(v) = 1.0,$$

como $v \notin \{w, u\}$ entonces $\psi_1(w, u) = 1.0$, entonces

$$\begin{aligned} s_1^u &= \max_{d \in V} (\min(\mu_{\sigma,n,V,S_n}(d), \psi_n(d, c))) \\ &= 0.1 \end{aligned}$$

Se hacen operaciones similares para σ^w , sin embargo, cambiar la condición μ_{σ,n,V,S_n} por σ_n^d , para valores de r entre 0.1 y 1.0 cumplen con las condiciones del teorema, por lo que la segmentación no es única. Por ejemplo, si $r=0.5$:

$$\begin{aligned} s_1^u &= \max_{d \in V} (\min(\sigma_1^d, \psi_n(d, c))) \\ &= 0.5. \end{aligned}$$

El propósito de este capítulo fue proporcionar las propiedades de una posible segmentación de una imagen en tonos de gris utilizando conjuntos difusos. Para esto se necesitan definir una afinidad difusa correspondiente a cada uno de los conjuntos en los que se desea segmentar esta imagen. El Teorema 4 garantiza la unicidad de una M -segmentación que se relacionan apropiadamente con las afinidades difusas dadas. En el siguiente capítulo se describe un algoritmo que permite construir dicha M -segmentación.

Capítulo 4

Algoritmo de segmentación con puntos semilla

Antes de la invención de las computadoras veloces las imágenes se segmentaban manualmente, como consecuencia la calidad de la segmentación dependía de la experiencia de las personas y no siempre lograban ponerse de acuerdo en cual era una segmentación apropiada. Con el advenimiento de mejores computadoras empezaron a crearse algoritmos de segmentación que sólo podían probarse en computadoras ubicadas en centros de investigación especializados en el tema. La principal dificultad de los algoritmos de segmentación es que en la imagen no hay cambios continuos de colores que permitan distinguir entre una región y otra. En muchos casos el usuario puede indicar el número de objetos que hay en una imagen, pero no siempre es claro cuáles son todos los píxeles que corresponden a cada objeto. Algunos algoritmos de segmentación dividen la imagen en demasiados objetos, lo que reduce su utilidad. Por lo tanto, resulta importante contar con algoritmos de segmentación que aprovechen la experiencia del usuario para dar una mejor segmentación.

En este capítulo se describen brevemente los algoritmos voraces y el algoritmo de Dijkstra que es la base para el algoritmo de segmentación utilizado en este trabajo y por último se muestran los resultados obtenidos con este algoritmo.

4.1. Algoritmos voraces

La forma de segmentar una imagen siguiendo el Teorema 4 del capítulo 3 tiene cierta dificultad como se pudo mostrar, ya que para encontrar una relación entre un píxel c y los puntos semilla S_m , $1 \leq m \leq M$, se forman todos los caminos existentes de c que pasan por algún punto $d \in V$ hasta llegar a un elemento de S_m . Este procedimiento se tiene que hacer para cada objeto m de los M objetos seleccionados por el usuario, y el número de caminos aumenta considerablemente conforme crece el número de píxeles en la imagen. Realizar un algoritmo con “fuerza bruta” causaría un gran consumo de tiempo y de memoria, por lo que en este trabajo se optó por utilizar el algoritmo de

Dijkstra que es un algoritmo de tipo *voraz* ([9]).

Un algoritmo voraz se puede utilizar para resolver problemas de optimización. La estrategia de un algoritmo voraz es obtener una solución óptima a un problema haciendo una secuencia de elecciones. Para cada decisión en el algoritmo se elige la opción que mejor le parezca en el momento. Esta estrategia heurística no siempre produce una solución óptima, pero a veces lo hace. Para resolver un problema con un algoritmo voraz se deben seguir los siguientes pasos ([9]):

1. Determinar la subestructura óptima del problema.
2. Desarrollar una solución recursiva.
3. Verificar que en cualquier etapa de la recursión, una de las opciones óptimas es la elección voraz.
4. Demostrar que todos menos uno de los subproblemas inducidos por haber tomado la decisión voraz es vacío.
5. Desarrollar un algoritmo recursivo que implemente la estrategia voraz.
6. Convertir el algoritmo recursivo a iterativo.

En la práctica, estos pasos son muy útiles para plantear el problema con la técnica voraz, pero ¿cómo saber cuando utilizar este tipo de algoritmos? De hecho no hay una regla general que nos permita saber esto, pero si un problema cumple con *la propiedad de elección voraz y subestructura óptima* entonces la técnica voraz puede funcionar.

La *propiedad de elección voraz* consiste en hallar una solución óptima a partir de opciones óptimas locales, es decir, cuando se considera cualquier elección se toma la mejor que se tenga para el problema actual sin considerar los resultados de los subproblemas. Con un algoritmo voraz las decisiones se toman con la información que se tenga en el momento sin depender de las decisiones futuras o de las soluciones de los subproblemas, sin embargo la decisión que se toma determina los subproblemas que se deben resolver. Un problema exhibe *subestructura óptima* si una solución óptima al problema contiene soluciones óptimas de los subproblemas. Se tiene la ventaja de asumir que se llega a un subproblema por haber tomado la decisión voraz en el problema original ([9]).

Existen dos tipos de enfoques en las técnicas:

- Bottom-up: Primero se encuentran soluciones óptimas a subproblemas, y una vez resuelto esto se encuentra una solución óptima al problema.
- Top-down: El problema se divide en subproblemas y estos se resuelven recordando las soluciones en caso de que sean necesarias.

Usualmente la técnica voraz se resuelve mediante un enfoque top-down haciendo una elección voraz tras otra, reduciendo en cada caso determinado el problema a uno más pequeño. La propiedad de elección voraz a menudo ayuda a ganar algo de eficiencia en la toma de decisión en un subproblema. Frecuentemente en el caso del procesamiento de entrada es bueno usar una estructura apropiada, como por ejemplo las colas de

prioridad, que pueden hacer más rápido una elección voraz y así obtener un algoritmo eficiente.

Regresando al Teorema 4, se ocupa un algoritmo voraz porque para encontrar la segmentación se deben formar caminos desde un elemento de algún objeto S_m al punto del que se quiere encontrar su vector de segmentación, $c \in V$. Más aún, estos caminos se dividen en dos partes, la primera parte la conforma una σm -cadena que empieza en un elemento de S_m a un punto $d \in V$, y la segunda parte consta del camino de d a c . La fuerza μ_{σ,n,V,S_n} de la σm -cadena ya es una solución óptima de ese subcamino dado que es la cadena con fuerza máxima, por lo que la solución del problema exhibe subestructura óptima (observar Figura 3.12).

4.2. Colas de prioridad

Cuando las personas van a la sala de urgencias en un hospital se atienden dependiendo de la gravedad de su enfermedad, o para la planificación de tareas primero se hacen las de mayor prioridad. Para poder representar este tipo de casos se hace uso de una estructura llamada cola de prioridad. Una cola de prioridad, H , es una estructura de datos en la que sus elementos tienen asociado un valor numérico, conocido como prioridad o clave. Los elementos de la cola se ordenan mediante dicha clave, pero si varios elementos tienen la misma prioridad se atenderán de modo convencional según la posición que ocupen. Existen dos tipos de colas de prioridad ([9]):

Colas de mínima prioridad: los elementos se extraen de menor a mayor prioridad.

Colas de máxima prioridad: los elementos se extraen de mayor a menor prioridad.

Existen 4 operaciones para una cola de prioridad, en este caso se describen las operaciones para una cola de máxima prioridad ya que es la que se utilizará en el algoritmo de segmentación, pero se pueden hacer cambios sencillos para obtener las operaciones de una cola de mínima prioridad.

1. INSERT(H, x): Inserta el elemento x al conjunto H y se denota por $H \leftarrow H \cup \{x\}$. El tiempo de ejecución es de $O(\log n)$.
2. MAXIMUM(H): Regresa el elemento de H con mayor prioridad y su tiempo de ejecución es de $\Theta(1)$ ¹.
3. EXTRACT-MAX(H): Regresa el valor del elemento en H con mayor prioridad y elimina dicho elemento de la cola, con tiempo de ejecución $O(\log n)$.
4. INCREASE-KEY(H, x, k): Si la clave k es mayor que la clave actual del elemento x , entonces se actualiza la clave al valor de k . Se ejecuta en tiempo $O(\log n)$.

La Figura 4.1 y 4.2 corresponde a la representación gráfica de una cola de máxima prioridad y la prioridad de los elementos se representa por un número del conjunto

¹ $\Theta(1)$ indica que el tiempo de ejecución de un algoritmo es constante para mayor información se puede consultar ([7])

$\{1, 2, 3, 4\}$, donde 4 representa la máxima prioridad. En la Figura 4.1 se aplica la operación $\text{INSERT}(H, x)$ a la cola de prioridad de la izquierda para agregar el “Elemento 5” con prioridad 3 y se obtiene como resultado la cola de prioridad de la derecha. El elemento con mayor prioridad es el que está más abajo y el de menor prioridad se encuentra en el tope. Como se puede observar, los elementos están ordenados respecto a su prioridad.

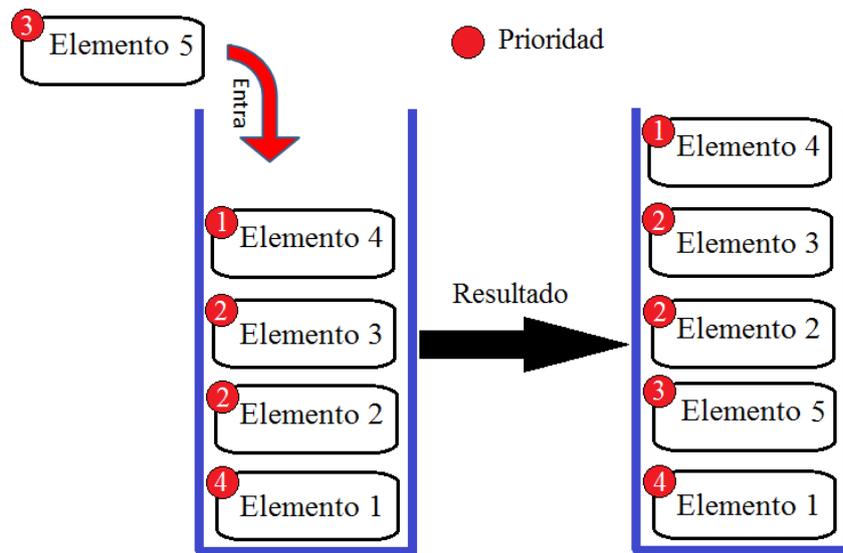


Figura 4.1: Representación gráfica de una cola de máxima prioridad a la que se le aplica la operación $\text{INSERT}(H, x)$.

Para extraer un elemento de la cola de prioridad se aplica la operación $\text{EXTRACT-MAX}(H)$ como se puede observar en la Figura 4.2. El elemento que se extrae es el que tiene la máxima prioridad, en este caso es el “Elemento 1” y como resultado se obtiene la cola de prioridad de la derecha.

4.3. Algoritmo de Dijkstra

Sea $G = (V, E)$ un grafo dirigido, donde V es un conjunto de vértices y E un conjunto de aristas con pesos no negativos. La función de pesos del grafo G está dado por $w : E \rightarrow \mathbb{R}$, es decir, $w(e), e \in E$ indica el peso de la arista e . El algoritmo de Dijkstra resuelve el problema del camino de costo mínimo simple de un vértice de partida s a los restantes vértices del grafo G ([9]).

Los costos de las aristas del grafo se almacenan en una matriz W y cada elemento se define de la siguiente forma:

$$w_{ij} = \begin{cases} 0, & \text{si } i = j; \\ \text{el costo de la arista } (i, j), & \text{si } i \neq j \text{ e } (i, j) \in E; \\ \infty, & \text{si } i \neq j \text{ e } (i, j) \notin E. \end{cases}$$

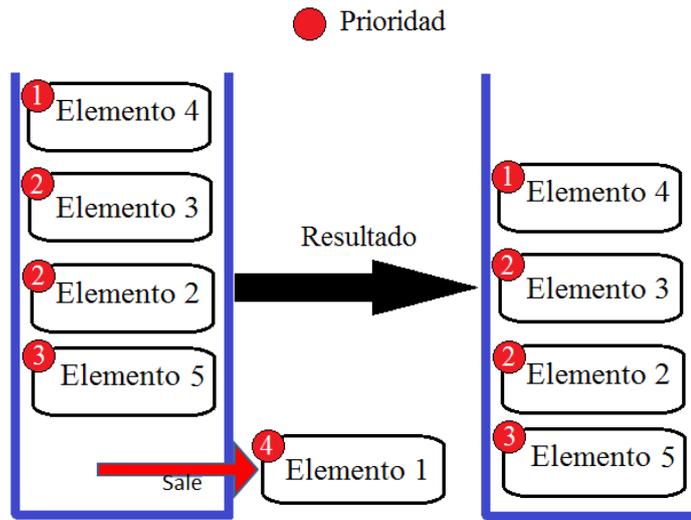


Figura 4.2: Representación gráfica de una cola de máxima prioridad a la que se le aplica la operación EXTRACT-MAX(H).

Para dos vértices $u, v \in V$ el costo del camino $p = (v_0, v_1, v_2, \dots, v_n)$ es la suma del costo de todas las aristas que lo constituyen, donde $v_0 = u$ y $v_n = v$,

$$w(p) = \sum_{i=1}^n w(v_{i-1}, v_i).$$

El camino p que va de u a v se denota como $u \rightsquigarrow v$. El costo de un camino corto de un vértice u a v se define como,

$$\delta(u, v) = \begin{cases} \text{mín} \{w(p) : p = u \rightsquigarrow v\}, & \text{si existe un camino } p \text{ de } u \text{ a } v ; \\ \infty, & \text{en otro caso .} \end{cases}$$

Se define un camino corto p de u a v como algún camino que cumple con $w(p) = \delta(u, v)$. Para estructurar una solución óptima del problema se requiere el siguiente lema ([9]).

Lema 1. *Sea G un grafo dirigido con función de costos $w : E \rightarrow \mathbb{R}$, sea $p = (v_1, v_2, \dots, v_n)$ un camino corto del vértice v_1 al vértice v_n y sea $p_{ij} = (v_i, v_{i+1}, \dots, v_j)$ para i y j tales que $1 \leq i \leq j \leq n$, un subcamino de p que va del vértice v_i a v_j . Entonces p_{ij} es un camino corto del vértice v_i al v_j .*

Demostración: Si descomponemos el camino p en $p_{1i} : v_1 \rightsquigarrow v_i, p_{ij} : v_i \rightsquigarrow v_j$ y $p_{jn} : v_j \rightsquigarrow v_n$

$$v_1 \xrightarrow{p_{1i}} v_i \xrightarrow{p_{ij}} v_j \xrightarrow{p_{jn}} v_n,$$

entonces

$$w(p) = w(p_{1i}) + w(p_{ij}) + w(p_{jn}).$$

Supongamos que existe un camino p'_{ij} que va del vértice v_i al vértice v_j con costo $w(p'_{ij}) < w(p_{ij})$, entonces el camino $v_1 \rightsquigarrow v_n$ que pasa por p'_{ij} con peso igual a $w(p_{1i}) +$

$w(p'_{ij}) + w(p_{jn})$, es menor que $w(p)$:

$$v_1 \xrightarrow{p_{1i}} v_i \xrightarrow{p'_{ij}} v_j \xrightarrow{p_{jn}} v_n.$$

Esto contradice el supuesto de que p es un camino corto de v_1 a v_n . \square

Con esto, el algoritmo de Dijkstra cumple con subestructura óptima, ya que cualquier subcamino del camino óptimo del vértice v_1 a v_n , también es óptimo. Para encontrar una solución recursiva a este problema denotemos d_{ij}^m como el peso mínimo de la ruta del vértice v_i al vértice v_j que tiene al menos m aristas. Si $m = 0$ no hay una arista que conecte directamente al vértice v_i con v_j y se pueden tener dos casos: que $v_i = v_j$ por lo que el peso es 0, y que $v_i \neq v_j$ entonces el peso vale ∞ , es decir:

$$d_{ij}^0 = \begin{cases} 0, & \text{si } i = j; \\ \infty, & \text{si } i \neq j. \end{cases}$$

Si $m \geq 1$, para encontrar el peso del vértice v_i a v_j se utiliza la técnica conocida como relajación. Esta técnica consiste en probar si se puede o no mejorar el camino corto a un vértice v_j que se ha encontrado hasta el momento pero pasando por un vértice v_k (ver Figura 4.3). En caso de que se logre mejorar, entonces se actualiza el camino. El resultado de una etapa de relajación está dado por la ecuación (4.1) y puede ser visto como una relajación de la restricción $d[v] \leq d[u] + w(u, v)$, la cual se le conoce como desigualdad del triángulo.

$$d_{ij}^m = \min \left\{ d_{ij}^{m-1}, \min_{1 \leq k \leq n} \{ d_{ik}^{m-1} + w(k, j) \} \right\}. \quad (4.1)$$

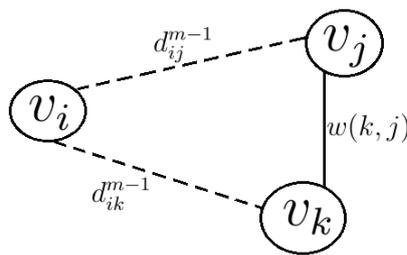


Figura 4.3: Representación de la técnica de relajación.

La ecuación (4.1) describe la solución recursiva al problema de Dijkstra. Consideremos el camino $v_i \rightsquigarrow v_k$ y que pasa por el vértice v_j .

$$v_i \xrightarrow{p_{ij}} v_j \xrightarrow{p_{jk}} v_k$$

Para encontrar el camino de costo mínimo del vértice v_i al vértice v_k primero se busca el camino de costo mínimo del vértice v_i a v_j y luego el de v_j a v_k . Una técnica es ir probando todos los caminos que puedan existir, entonces v_j puede ser cualquier vértice

y se tendrían dos problemas con muchas opciones. Con ayuda de la técnica voraz, el algoritmo de Dijkstra encuentra al vértice v_j tal que cumpla con d_{ij}^{m-1} , por lo que ya no se preocupa por el resto de los caminos.

El Algoritmo 1 es el de Dijkstra, que obtiene como salida un conjunto S de vértices, que representa el camino más corto del vértice inicial s . Este algoritmo hace uso de una cola de mínima prioridad Q por lo que su tiempo de ejecución es de $O(E \log V)$.

Algoritmo 1 Dijkstra (G, w, s)

Entrada: Matriz de adyacencia W

Salida: Conjunto S de vértices

```

1: INITIALIZE-SINGLE-SOURCE( $G, S$ )
2:  $S \leftarrow \emptyset$ 
3:  $Q \leftarrow V[G]$ 
4: mientras  $Q \neq \emptyset$  hacer
5:    $u \leftarrow \text{EXTRACT-MIN}(Q)$ 
6:    $S \leftarrow S \cup \{u\}$ 
7:   para cada vértice  $v \in \text{Adj}[u]$  hacer
8:     RELAX( $u, v, w$ )
9:   fin para
10: fin mientras

```

El Algoritmo 2 describe la función que se utiliza en el paso 1 del Algoritmo 1. A cada vértice v le corresponde el valor $d[v]$ que es un límite superior del peso de la ruta más corta desde el vértice inicial s a v . El vector π guarda el antecesor del vértice actual y, al final, con él se puede encontrar el camino de costo mínimo.

Algoritmo 2 INITIALIZE-SINGLE-SOURCE(G, s)

```

1: para para cada vértice  $v \in V[G]$  hacer
2:    $d[v] \leftarrow \infty$ 
3:    $\pi[v] \leftarrow NIL$ 
4: fin para
5:  $d[s] \leftarrow 0$ 

```

En los pasos 4-8 del Algoritmo 1 selecciona un vértice $u \in V - S$ con el camino más corto, agrega u a S y hace la relajación de todos los vértices que salen de u . En el Algoritmo 3 se describe la técnica de relajación y en el Ejemplo 23 se puede ver una implementación del algoritmo de Dijkstra.

Algoritmo 3 Relax(u, v, w)

```

1: si  $d[v] > d[u] + w(u, v)$  entonces
2:    $d[v] \leftarrow d[u] + w(u, v)$ 
3:    $\pi[v] \leftarrow u$ 
4: fin si

```

Ejemplo 23. En el grafo que se observa en la Figura 4.4 a) se desea encontrar el camino con costo mínimo del vértice A a cualquier otro vértice. Para cada elemento, la variable $d[]$ se inicializa con un valor suficientemente grande que se denota por ∞ y el valor de π se deja vacío porque aún no se encuentra el camino correspondiente (paso 1, Algoritmo 1).

En b) se agregan todos los elementos a la cola de mínima prioridad Q para buscar el camino con costo mínimo y se parte del nodo A que es el elemento en la primera posición de la cola (paso 3, Algoritmo 1). Se inicializa $d[A] = 0$ porque aún no se forma ningún camino hacia él, y $\pi[A]$ no contiene valor. Luego, se aplica el método de relajación de A a todos sus nodos adyacentes y la distancia correspondiente se actualiza al igual que el valor de π (paso 8, Algoritmo 1). Por ejemplo en c), $d[B] = 2$ y $\pi[B] = A$ porque proviene de A , después B es eliminado de Q . Se agrega el nodo visitado A al conjunto S que almacena el camino de costo mínimo. Este paso se repite para cada nodo adyacente.

De d) a g) se hace el mismo procedimiento que en c) por ejemplo, para el nodo E , $d[E] = 6$ que es la suma del camino de A a B y de B a E . También, $\pi[E] = B$ porque el camino más corto hacia E proviene de B . En h) se tienen tres puntos a destacar: la cola de prioridad está vacía porque se han visitado todos los nodos, d contiene el valor de las distancias mínimas para su respectivo nodo y π contiene el camino mínimo del nodo final al nodo de inicio. Por ejemplo, para el nodo I , el camino mínimo proviene de G , y el nodo anterior a G es D y luego A , es decir:

$$p_{AD} = (A, D, G, I).$$

En la siguiente sección se describe la relación entre el algoritmo de Dijkstra y el de segmentación que se desarrolla en la presente tesis.

4.4. Algoritmo de segmentación

El objetivo del algoritmo de Dijkstra es encontrar el costo mínimo que hay entre un vértice de inicio hacia el resto de los vértices, por lo que utiliza una cola de mínima prioridad, por el contrario, el algoritmo de segmentación utiliza una cola de máxima prioridad ya que se desea encontrar el objeto al que pertenece un píxel con el máximo grado de pertenencia. La matriz de adyacencia del algoritmo de Dijkstra, visto desde la segmentación, corresponden a las afinidades entre los píxeles, pero estas se calculan en el proceso del algoritmo.

El Algoritmo 4 encuentra la segmentación de una imagen utilizando la teoría desarrollada en el Capítulo 3. El algoritmo recibe como entrada los puntos semilla seleccionados por el usuario. Dependiendo del número de conjuntos de puntos semilla, M , seleccionados es el número de conjuntos difusos en los que se segmenta la imagen. Los

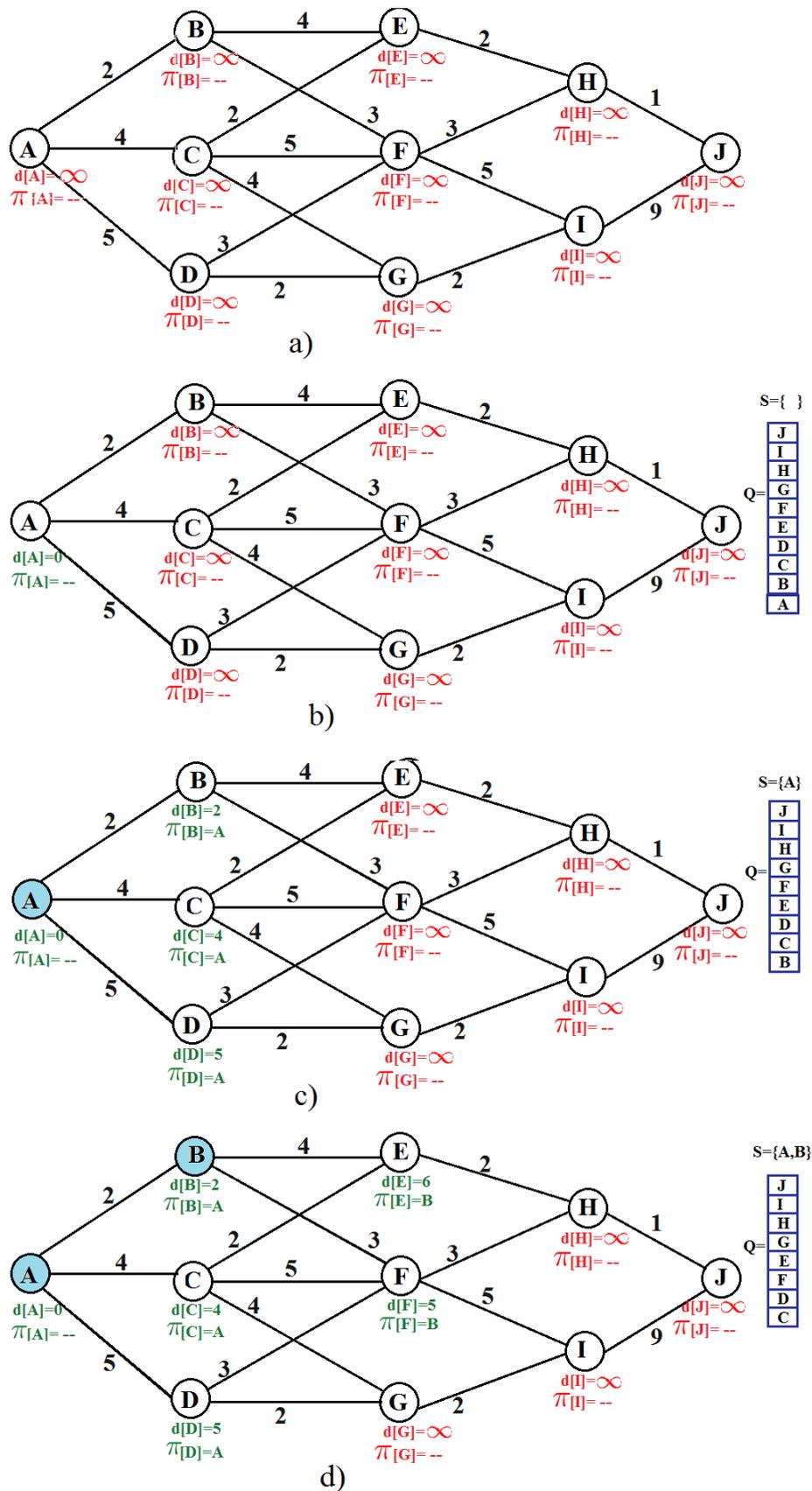


Figura 4.4: Aplicación del método de Dijkstra al grafo a) e inicialización de los vectores d y π . En b) el algoritmo se aplica al primer nodo, A. De c) a g) se muestran algunas ejecuciones del algoritmo (Ver la continuación en la Figura 4.5).

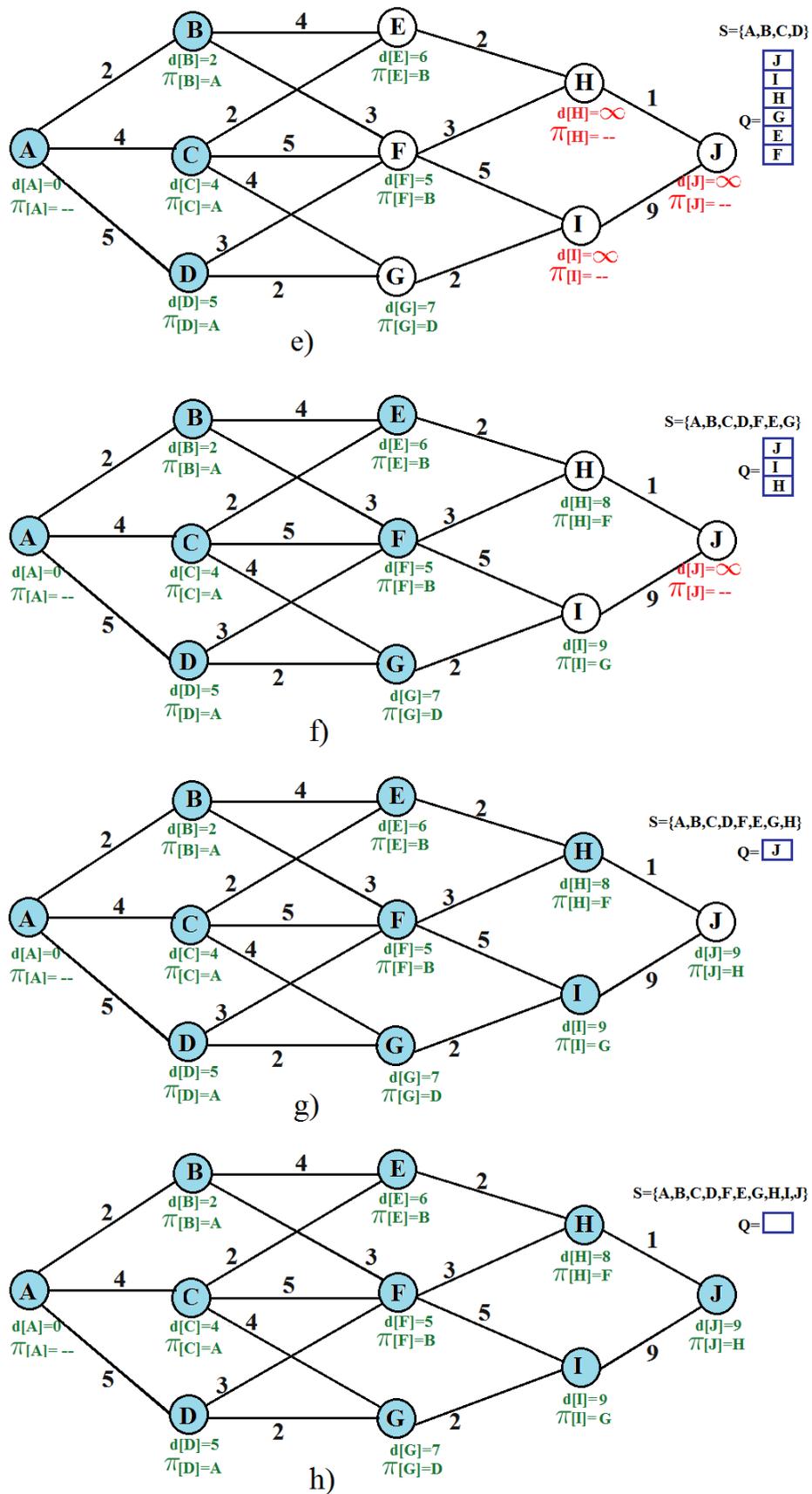


Figura 4.5: Aplicación del método de Dijkstra. En h) se puede observar el camino con costo mínimo de A a cualquier nodo (Ver la primera parte en la Figura 4.4).

M conjuntos de puntos semilla se denotan por S_m y los conjuntos difusos obtenidos de la segmentación por V_m , para $1 \leq m \leq M$.

El algoritmo utiliza una cola de prioridad para almacenar los píxeles de la imagen denotada por H y sus elementos, $c \in H$, son los píxeles cuya clave asociada es σ_0^c . Cada píxel se almacena una vez en H , lo que se denota por $H \leftarrow H \cup \{c\}$, y se extrae cuando el vector σ^c obtiene su valor final, usando la operación EXTRACT-MAX(H). La variable l representa el valor de la máxima clave que se obtiene por MAXIMUM(H).

En los pasos del 1 al 5, se inicializa el valor de σ_m^c en 0, para cada píxel $c \in V$ y $0 \leq m \leq M$. Luego, para todos los píxeles semilla, $c \in S_m$, los valores de σ_0^c y σ_m^c toman el valor de 1, y se almacenan en la cola de prioridad (pasos 6-14), porque estos valores pertenecen totalmente a los objetos y su grado de pertenencia es igual a 1. Por lo tanto, l inicializa con el valor 1. Después de este proceso se cumplen las siguientes condiciones:

1. σ es una M -segmentación de V .
2. $c \in H$ si y sólo si $0 < \sigma_0^c \leq l$.
3. l es la máxima clave de H .
4. Para $1 \leq m \leq M$, $U_m = \{c \in H \mid \sigma_m^c = l\}$.

La condición 1 se refiere a que por el momento se tiene una posible M -segmentación, donde cada objeto consta de los puntos semillas que fueron seleccionados por el usuario. Como hasta el momento se agregaron a la cola todos los puntos semilla, entonces $\sigma_0^c = 1$ (condición 2), y $l = 1$ es la máxima clave de H (condición 3), por lo tanto se tiene la condición 4.

Después de la inicialización sigue el ciclo principal, el cual se repite mientras el valor de l sea positivo, y en cada iteración decrece su valor. La sección esencial del algoritmo es la primera parte del ciclo principal (pasos 18- 41). En esta parte, el algoritmo tiene como objetivo actualizar la mejor estimación hasta el momento del valor final de σ_m^c . Dicho valor se actualiza si se encuentra una σm -cadena de algún píxel en U_m a c con ψ_m -fuerza mayor a su valor anterior, en otro caso, si se encuentra una σn -cadena, $n \neq m$, cuya ψ_n -fuerza es mayor al valor anterior de σ_m^c , entonces este último se reemplaza por 0. Este procedimiento se realiza mediante la construcción de un conjunto C (paso 21), de tal forma que para cada $c \in C$ sea adyacente a un elemento $d \in U_m$ y que esté lo más cerca posible a c , por tal motivo, se eligen todos aquellos píxeles con σ_m^c menor a l o a $\psi_m(c, d)$. La cardinalidad del conjunto C depende de la adyacencia que se considere, para las pruebas que se realizaron en esta tesis solo puede ser de 4 u 8, ya que se considera una 4 u 8-adyacencia. La parte final del ciclo principal, pasos 42-48, permite escoger a l de tal forma que representa la máxima clave conforme se eliminan los elementos en H . También agrega los elementos que han obtenido su valor final de σ_0^c hasta el momento a sus respectivos objetos, es decir, para $1 \leq m \leq M$, $U_m = \{c \in H \mid \sigma_m^c = l\}$. Una vez terminado este proceso en cada iteración se puede decir que se cumplen las cuatro condiciones anteriores. Se cuenta con una M -segmentación donde los conjuntos U_m contienen más elementos que en la iteración anterior aunque aún no

se tenga la M -segmentación completa (condición 1). Los elementos que están en espera en la cola de prioridad tienen clave igual o menor a l , pero l decrece en cada iteración. Por último, la condición 4 se cumple por el paso 47 del algoritmo.

Recordando el algoritmo de Dijkstra, para calcular el costo mínimo de un nodo de inicio al resto empieza por los nodos más cercanos a él, es decir, con los nodos conectados a él y que el peso de la arista sea mínima. Este proceso se realiza con todos los nodos, y si en el proceso encuentra un camino con menor peso entonces hace la relajación. El algoritmo de segmentación descrito aquí se relaciona con el algoritmo de Dijkstra ya que primero analiza los píxeles semilla y después busca los píxeles adyacentes a él, dependiendo de la función de afinidad. Toma todos los píxeles $c \in V$ que pueden ser muy similares a los puntos semilla, es decir, aquellos que pueden tener un grado de pertenencia igual a 1 pero que no están considerados como puntos semilla, por lo que el valor de l es 1, y trata de relajar el valor que tienen hasta el momento. La relajación se realiza en los pasos 24 y 36. Este procedimiento hace uso de σm -cadenas que van de algún píxel en U_m a c . Una vez que se termina de analizar estos píxeles, se buscan aquellos $c \in V$ que en la cola de prioridad tengan la máxima clave, es decir, aquellos píxeles cuyo grado de adyacencia a los puntos semilla sea mayor en comparación con el resto que esperan en la cola, pero no tanto como los que ya han sido analizados. El algoritmo termina hasta que la cola esté vacía.

En el procedimiento del ciclo principal se encuentran cuatro tipos de píxeles, d :

1. Los que son agregados a H y posteriormente son eliminados de la cola de prioridad.
2. Los que cumplen con $\sigma_0^d = \sigma_m^d = l$, estos píxeles tal vez no han sido almacenados en H .
3. Los que cumplen con $0 < \sigma_0^c < l$, estos píxeles ya tienen un valor para σ_0^c , para un cierto U_m y puede darse el caso de encontrar una σm -cadena para algún $v \in U_n$ tal que σ_0^c sea mayor.
4. Los que cumplen con $\sigma_0^c = 0$, que son píxeles que por el momento no se ha encontrado una σm -cadena a algún punto semilla de S_m .

El tiempo de ejecución para las operaciones de la cola de prioridad ($H \leftarrow H \cup \{c\}$) y eliminar (EXTRACT-MAX(H)) se realizan en $O(\log n)$. En el tiempo de ejecución del Algoritmo 4 influye el tiempo en que se realizan las operaciones de la cola de prioridad y el tiempo para construir el conjunto C (paso 21). Sin embargo, el tiempo para construir C depende de la adyacencia que se utiliza, en el caso de las pruebas o bien es 4 u 8. Esta constante se denotará por L . Este proceso se realiza para los M objetos que selecciona el usuario. Cada proceso se realiza para los N píxeles de la imagen, entonces, el tiempo de ejecución del Algoritmo 4 es de $O(N(\log N + ML))$.

Algoritmo 4 Segmentación

```

1: para  $c \in V$  hacer
2:   para  $m \leftarrow 0$  a  $M$  hacer
3:      $\sigma_m^c \leftarrow 0$ 
4:   fin para
5: fin para
6:  $H \leftarrow \emptyset$ 
7: para  $m \leftarrow 1$  a  $M$  hacer
8:   para  $c \in S_m$  hacer
9:     si  $\sigma_0^c \leftarrow 0$  entonces
10:       $H \leftarrow H \cup \{c\}$ 
11:       $\sigma_0^c \leftarrow \sigma_m^c \leftarrow 1$ 
12:     fin si
13:   fin para
14: fin para
15:  $l \leftarrow 1$ 
16:  $U_m \leftarrow S_m$ 
17: mientras  $l > 0$  hacer
18:   para  $m \leftarrow 1$  a  $M$  hacer
19:     mientras  $U_m \neq \emptyset$  hacer
20:       eliminar un píxel  $d$  de  $U_m$ 
21:        $C \leftarrow \{c \in V \mid \sigma_m^c < \text{mín}(l, \psi_m(d, c)) \text{ y } \sigma_0^c \leq l\}$ 
22:       mientras  $C \neq \emptyset$  hacer
23:         eliminar un píxel  $c$  de  $C$ 
24:          $t \leftarrow \text{mín}(l, \psi_m(d, c))$ 
25:         si  $l = t$  y  $\sigma_m^c < l$  entonces
26:            $U_m \leftarrow U_m \cup \{c\}$ 
27:         fin si
28:         si  $\sigma_0^c < t$  entonces
29:           si  $\sigma_0^c = 0$  entonces
30:              $H \leftarrow H \cup \{c\}$ 
31:             para  $n \leftarrow 1$  a  $M$  hacer
32:                $\sigma_n^c \leftarrow 0$ 
33:             fin para
34:           fin si
35:           si  $\sigma_0^c \leq t$  entonces
36:              $\sigma_0^c \leftarrow \sigma_m^c \leftarrow t$ 
37:           fin si
38:         fin si
39:       fin mientras
40:     fin mientras
41:   fin para
42:   mientras Maximum-Key ( $H$ ) =  $l$  hacer
43:     Remove-Max ( $H$ )
44:   fin mientras
45:    $l \leftarrow \text{Maximum-Key} (H)$ 
46:   para  $m \leftarrow 1$  a  $M$  hacer
47:      $U_m \leftarrow \{c \in H \mid \sigma_m^c = l\}$ 
48:   fin para
49: fin mientras

```

4.5. Resultados

La función de afinidad que se utilizó en las siguientes aplicaciones y está dada por:

$$\psi_m(c, d) = \begin{cases} 0, & \text{si } c = d; \\ \frac{g_{1m}(f(c) + f(d)) + g_{2m}(|f(c) - f(d)|)}{2}, & \text{en otro caso.} \end{cases} \quad (4.2)$$

donde $1 \leq m \leq M$ y para $i = 1, 2$,

$$g_{im}(x) = \exp\left(-\frac{(x - \mu_{im})^2}{2\sigma_{im}^2}\right). \quad (4.3)$$

Los parámetros μ_{1m} y σ_{1m} de la función 4.3 corresponden a la media y desviación estándar, respectivamente, de $f(c) + f(d)$ siendo c y d elementos de S_m adyacentes. Por otro lado, μ_{2m} y σ_{2m} son la media y desviación estándar de la diferencia $|f(c) - f(d)|$ de los píxeles adyacentes c y d en S_m . La afinidad entre dos píxeles adyacentes c y d crece si $f(c) + f(d)$ y $|f(c) - f(d)|$ son similares en alguna región identificada. En caso contrario, el valor de dicha afinidad es pequeño. Esta función se obtuvo de [16], consideramos esta afinidad difusa porque toma en cuenta la posición espacial de los píxeles, el promedio y la varianza entre las intensidades de los puntos semilla.

Para las pruebas se utilizó una 4 y 8-adyacencia para comparar dichos resultados, pero se puede observar que no hay gran diferencia entre los dos casos.

4.5.1. Características de la máquina y ejecución del algoritmo

La implementación del algoritmo se hizo en *C++* con algunas librerías de *OpenCV* (versión 2.4.9) para seleccionar los objetos en la imagen, y tiene por nombre *MSegmentacion.cpp*. Las características de la máquina donde se ejecutó el programa son: HP-1000-Notebook-PC, memoria 1.8 Gb, procesador Intel celeron CPU B820 1.70GHz. En el algoritmo se debe especificar el nombre de la imagen y puede leer cualquier tipo de imagen: png, jpg, bmp, gif, etc. Al ejecutar el código, el usuario debe ingresar el número de conjuntos de puntos semilla y con clic izquierdo se deben seleccionar los puntos semilla de cada conjunto sin soltar el mouse. Para seleccionar otro conjunto de puntos semilla se debe dar clic derecho y al finalizar la selección se debe dar "enter" para que el programa inicie la segmentación. El programa dará como resultado tantas imágenes como objetos se hayan indicado, esto para poder visualizar el resultado de la M -segmentación. Las M imágenes que se obtienen son en tonos de gris y cada una corresponde a un conjunto difuso obtenido de M -segmentación. El valor de σ_m^c es el grado de pertenencia del píxel c al conjunto difuso V_m para $1 \leq m \leq M$; si el grado de pertenencia de c a V_m es 1 entonces le corresponde el tono blanco en la imagen. Entre más pequeño sea el grado de pertenencia el nivel de gris se va degradando hasta llegar al negro, lo que indica que el píxel pertenece al conjunto difuso V_m con un valor cercano a 0.

Como se puede notar, la intervención del usuario es casi nula, solo debe seleccionar algunos puntos semilla de los objetos y el algoritmo hace el resto del trabajo. La ejecución del algoritmo es rápida, tarda unos segundos para imágenes grandes y la selección de objetos es sencilla.

A continuación se muestran algunas pruebas que se hicieron con el algoritmo de segmentación 4.

4.5.2. Figura geométrica

La Figura 4.6(a) es de 913x794 píxeles, contiene un rombo en tonalidad gris más oscuro que el fondo y en algunas esquinas del rombo hay pequeños círculos de un gris similar al rombo, por lo que es difícil diferenciarlos. La Figura 4.6 es la segmentación resultante obtenida con una 4-adyacencia y la Figura 4.7 corresponde a una 8-adyacencia. Se seleccionan dos conjuntos de puntos semilla en la imagen, S_1 : puntos semilla marcados en amarillo (algunos píxeles en el rombo) y S_2 : marcados en azul (parte del fondo) por lo que se obtiene como resultado dos imágenes en tono de gris. La segmentación divide a la imagen en dos conjuntos difusos V_1 y V_2 . Las imágenes 4.6(b) y 4.7(b) tienen las siguientes características:

- Resaltan en blanco los píxeles que pertenecen totalmente a V_1 .
- Debido a que la intensidad de gris de los círculos es similar a los puntos semilla S_1 en comparación a S_2 , el grado de pertenencia de los píxeles del círculo es mayor a V_1 que a V_2 . Por esto, la intensidad de los píxeles de los pequeños círculos es muy cercano al blanco.
- El grado de pertenencia de los píxeles del fondo a V_1 es igual a 0, por lo que la tonalidad de dichos píxeles es negro.

En las imágenes 4.6(c) y 4.7(c) se representa el grado de pertenencia para el segundo objeto:

- El fondo es blanco ya que los píxeles que conforman al fondo tienen un grado de pertenencia de 1 hacia V_2 .
- Los píxeles que conforman al rombo tienen un grado de pertenencia igual a 0 por lo que su intensidad es negro.

El tiempo de ejecución utilizando una 4-adyacencia es de 7.882 segundos mientras que para una 8-adyacencia es de 11.74 segundos y los resultados obtenidos son muy similares a pesar de que no son los mismos píxeles que el usuario seleccionó en ambos casos.

La elección de los puntos semilla es importante debido a que puede resultar una segmentación inadecuada si no se eligen correctamente, como se puede observar en la Figura 4.8. En dicha figura se seleccionó la parte izquierda del fondo como primer objeto (Figura 4.8(b)) y el rombo como segundo objeto (Figura 4.8(c)). Dado que el rombo divide al fondo, no hay un píxel del lado derecho del fondo que conecte directamente

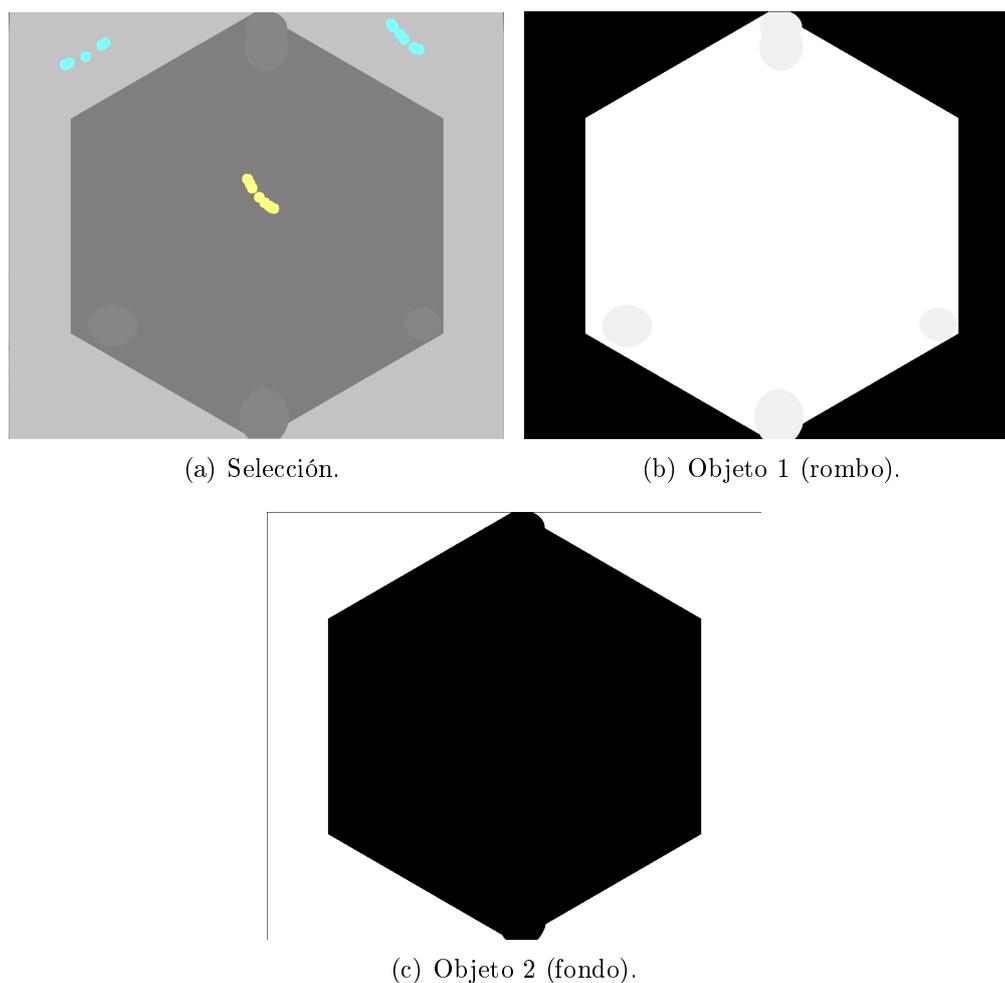


Figura 4.6: Segmentación que resulta de la imagen (a). Las imágenes (b) y (c) son el resultado de la segmentación al utilizar una 4-adyacencia.

con el lado izquierdo, por lo que el algoritmo de segmentación propuesto asigna la parte derecha del fondo al rombo. Debido a este inconveniente, se deben seleccionar los puntos semilla de la mejor manera, esto se deja a la experiencia del usuario. Este caso de prueba se realizó con una 8-adyacencia en 11.025 segundos, pero utilizando una 4-adyacencia se obtienen resultados similares.

4.5.3. Fotógrafo

La Figura 4.9 contiene imágenes de un fotógrafo y es muy utilizada en el procesamiento de imágenes, esta imagen se obtuvo de [3] pero fue modificada a tamaño 700×696 . Se seleccionaron 3 conjuntos de puntos semilla, el primer objeto marcado en amarillo (píxeles en el cielo), el segundo marcado en color verde (píxeles sobre el suelo) y el tercer objeto etiquetado en azul (el fotógrafo). La Figura 4.9 corresponde a la segmentación con una 4-adyacencia en un tiempo de ejecución de 5.032 seg. y la Figura 4.10 es para una 8-adyacencia en un tiempo de 7.292 seg. En ambos casos, en la

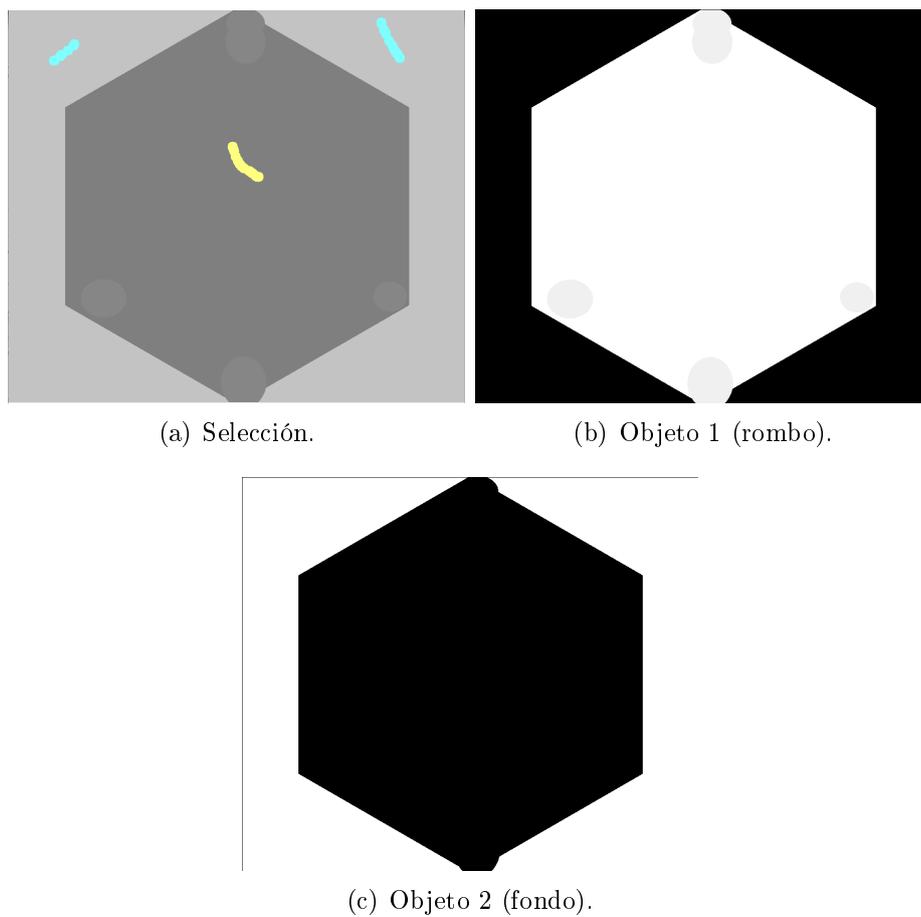


Figura 4.7: Las imágenes (b) y (c) son el resultado de la segmentación aplicado a la imagen (a) utilizando una 8-adyacencia.

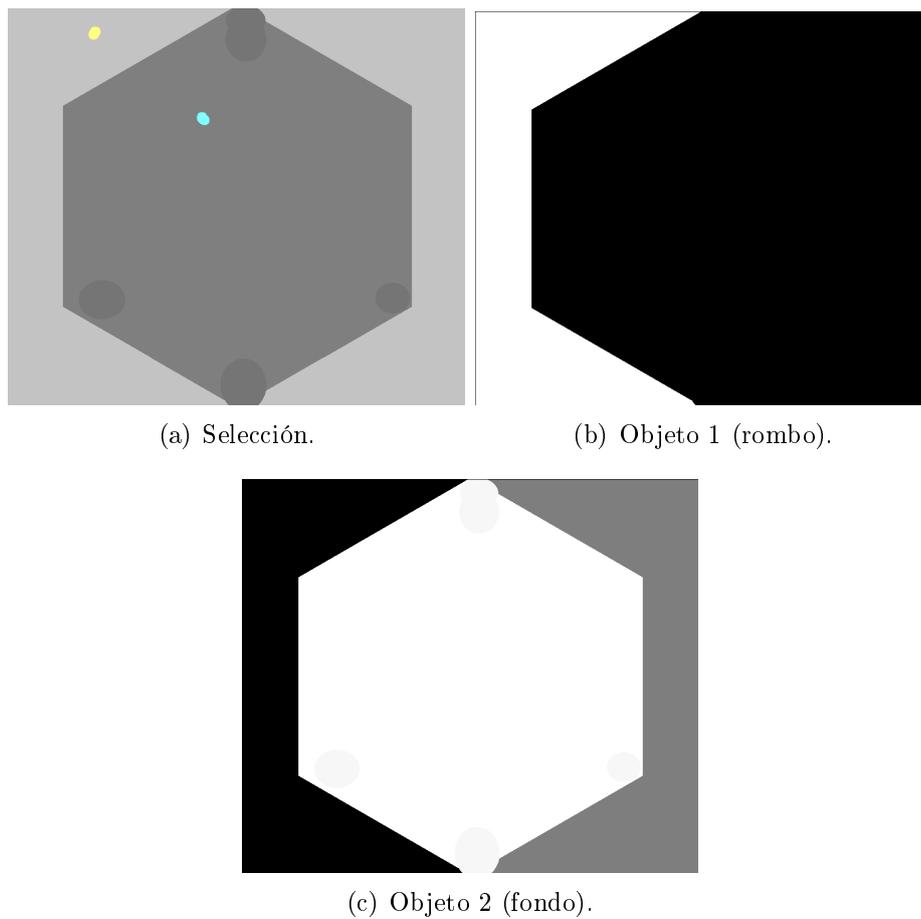


Figura 4.8: Como resultado de la segmentación aplicado a la imagen (a) utilizando una 8-adyacencia se obtiene las imágenes (b) y (c).

imagen correspondiente al primer objeto (4.9(b) o 4.10(b)) se distinguen ligeramente los bordes de los edificios. Parte del trípode de la cámara pertenece al objeto 2 (4.9(c) o 4.10(c)) y la parte superior del trípode se marca para el objeto 3 ya que la cámara es de color negro al igual que el saco del fotógrafo. En el objeto 3, Figura 4.9(d) o 4.10(d), el algoritmo asigna una pequeña sección del cielo a este objeto ya que está en un gris oscuro y aunque no es parte del fotógrafo, está cerca de este.



Figura 4.9: Segmentación de la imagen (a) utilizando una 4-adyacencia. Se seleccionaron 3 objetos y el resultado se puede observar en las imágenes (b), (c) y (d). El tiempo de ejecución es de 5.032 segundos.

4.5.4. Imagen por satélite

Las aplicaciones en el análisis de imágenes satelitales ha crecido vertiginosamente en los últimos años y abarcan diferentes campos como la agricultura, la cartografía, la meteorología y la forestación entre otros. Una imagen satelital contiene una gran

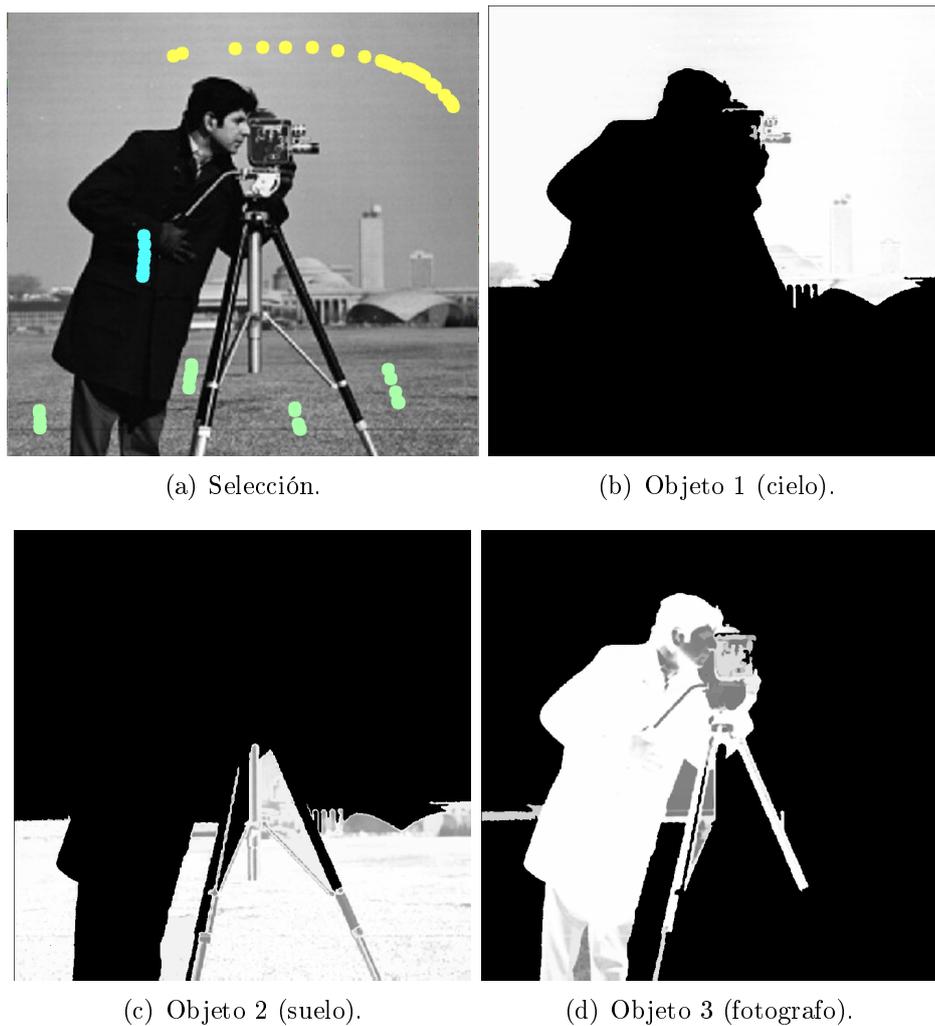


Figura 4.10: Se segmenta en 3 objetos a la imagen (a) obteniendo como resultado las imágenes de (b) a (d), en un tiempo de 7.292 segundos.

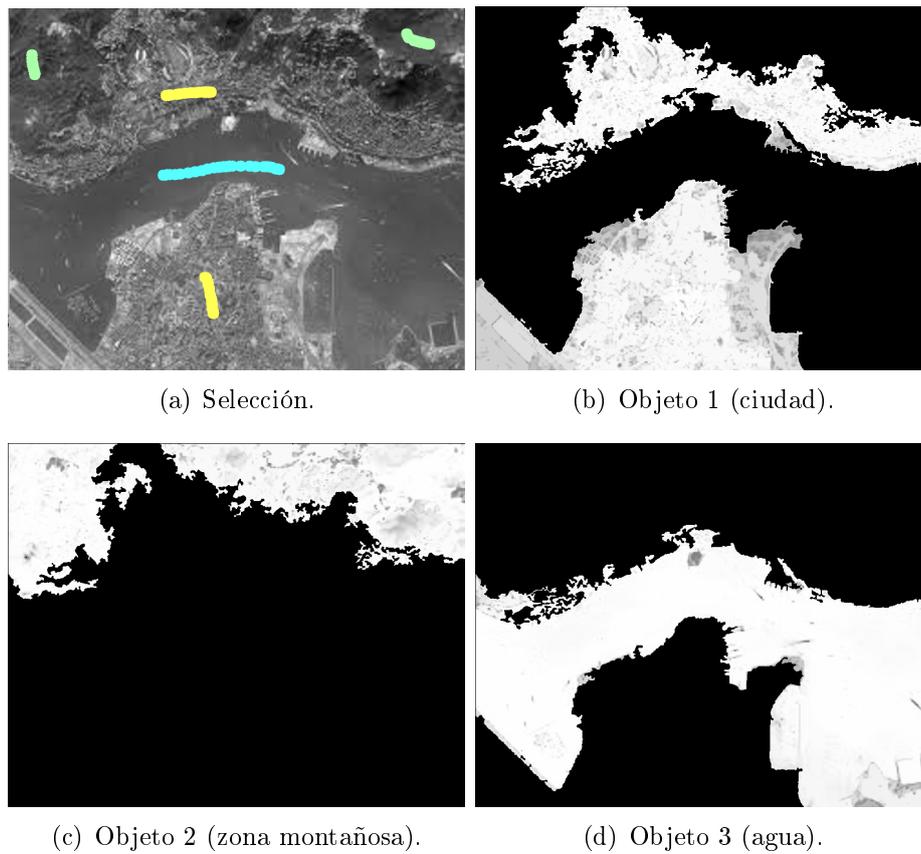


Figura 4.11: Imagen por satélite de la región de Hong Kong, la imagen se divide en 3 objetos y las imágenes (b), (c) y (d) son los resultados de la segmentación para una 4-adyacencia.

cantidad de información que no se puede obtener inmediatamente, pero se puede resaltar mediante técnicas de procesamiento de imágenes como es la segmentación. Dicha información puede ser la contaminación de ríos, zonas de interés turístico, incendios, enfermedades en cosechas, etc.

En las Figuras 4.11(a) y 4.12(a) se observa una imagen satelital de tamaño 807×641 que corresponde a la región de Hong Kong, la ciudad está dividida por una sección de agua y una parte de la ciudad que aún no está en construcción. Se seleccionaron 3 objetos (el agua, la ciudad, la zona montañosa) y se aplicó el algoritmo utilizando una 4-adyacencia y 8-adyacencia obteniendo las imágenes 4.11 y 4.12, respectivamente. Se puede observar que la segmentación es buena, salvo pequeños detalles. En el objeto 3 con una 4-adyacencia (ver Figura 4.11(d)) una porción de lo que es tierra lo reconoce como agua, en este caso se puede especificar al programa que esta sección no es agua marcándola como parte del objeto 2, sin embargo esto no sucede con una 8-adyacencia pero sí marca una pequeña sección de agua como parte del objeto 1 (ver Figuras 4.12(b) y 4.12(c)). El tiempo de ejecución es de 5.329 y 7.511 segundos, respectivamente. Dicho tiempo es mayor para una 8-adyacencia pero se puede observar una mejor segmentación.

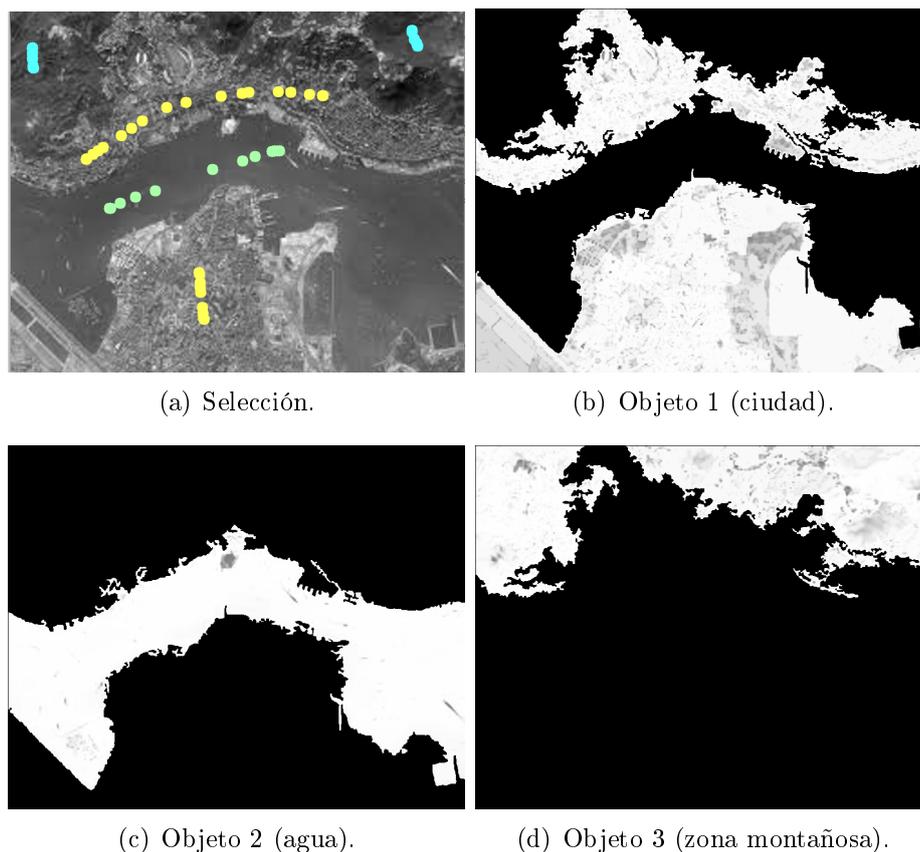


Figura 4.12: Segmentación que resulta de la imagen (a) al seleccionar 3 objetos con una 8-adyacencia. Se observa una mejor segmentación en comparación a una 4-adyacencia.

4.5.5. Resonancia magnética del cerebro humano

La segmentación de imágenes es un problema fundamental en el procesamiento de imágenes y visión por computadora, muy aplicado en el análisis de imágenes médicas. Las imágenes de resonancia magnética (MRI) proveen información importante para cirugías reparadoras, planificación del tratamiento de radioterapia, neurocirugía estereotáctica, entre otros. El segmentar la materia blanca, la materia gris y el fluido cerebro-espinal es de suma importancia en el análisis cuantitativo del volumen de diferentes partes del cerebro, con el propósito de evaluar el progreso o condición de diversas enfermedades como el Alzheimer y la epilepsia ([20]).

La segmentación de imágenes del cerebro no es una tarea fácil, entre las dificultades se encuentra que las intensidades de los píxeles no son necesariamente constantes para cada clase del tejido. Por lo tanto, los métodos tradicionales de la agrupación y basados en umbralización no producen los resultados deseados. Otra dificultad es debido a la presencia de ruido en los datos, lo que hace que los métodos de clasificación de píxeles den resultados poco realistas. Por esto, es importante implementar algoritmos que tomen en cuenta la estimación de la intensidad variable para cada clase, y el conocimiento previo acerca de la ubicación espacial de las regiones correspondientes ([20]). La segmentación de imágenes depende del experto en el tema, en este caso, la segmentación del cerebro depende de la información de los médicos, sin embargo en la actualidad los propios médicos no se han puesto de acuerdo con dicha segmentación, lo que muestra otro inconveniente.

Como se ha mencionado, el algoritmo del presente trabajo segmenta una imagen a partir de los objetos que el usuario indique, y esto depende de para qué se deseé utilizar. Tal vez a los médicos les interese distinguir entre materia blanca y gris, o entre el cráneo y el cerebro, o bien una segmentación más detallada. Si la imagen de resonancia magnética es de buena calidad se puede hacer una segmentación más detallada, pero por lo general no es el caso y para las siguientes imágenes menos, por lo que se decidió segmentar en pocos objetos. La Figura 4.13 es un corte coronal del cerebro humano obtenida de [16] y modificada al tamaño 518×512 . Se segmentó en 3 objetos, el primero corresponde al fondo de la imagen, el segundo es el cerebro y el tercer objeto el cráneo. La Figura 4.13 corresponde a una segmentación con una 4-adyacencia mientras que la imagen 4.14 corresponde a una 8-adyacencia, se puede observar que los resultados difieren por muy poco, por ejemplo en la parte inferior de la segmentación 4.13(c) agrega más píxeles en comparación a 4.14(c). El tiempo de ejecución es de 2.844 segundos para una 4-adyacencia y 4.073 segundos para una 8-adyacencia.

La imagen 4.15 es un corte axial del cerebro humano de tamaño 518×583 , se segmentó en 3 objetos, el primero corresponde al encéfalo, el segundo es el cráneo y el tercer objeto corresponde al fondo. La Figura 4.15 corresponde a una segmentación con una 4-adyacencia mientras que la Figura 4.16 corresponde a una 8-adyacencia, se puede observar que los resultados difieren por muy poco. El tiempo de ejecución es de 3.211 segundos para una 4-adyacencia y 4.547 para una 8-adyacencia.

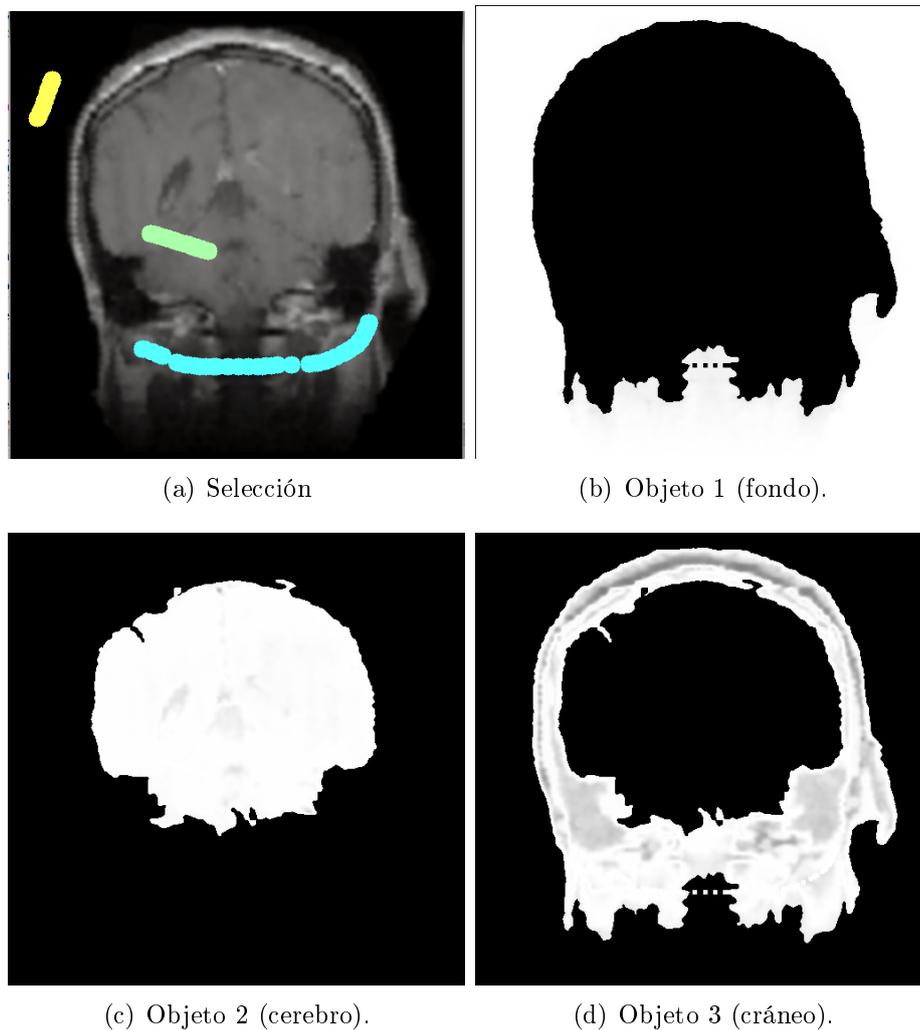


Figura 4.13: Segmentación de una imagen de resonancia magnética del cerebro tipo coronal utilizando una 4-adyacencia. La imagen se divide en 3 objetos, el cerebro, el cráneo y el fondo.

Imagen	4-adyacencia	8-adyacencia	Tamaño de la imagen
Figura geométrica	7.882	11.74	913×794
Fotógrafo	5.032	7.292	700×696
Imagen satelital	5.329	7.511	807×641
Corte coronal	2.844	4.073	518×512
Corte axial	3.211	4.547	518×583

Tabla 4.1: Resumen del tiempo de ejecución en segundos de los casos de prueba para su respectiva adyacencia.

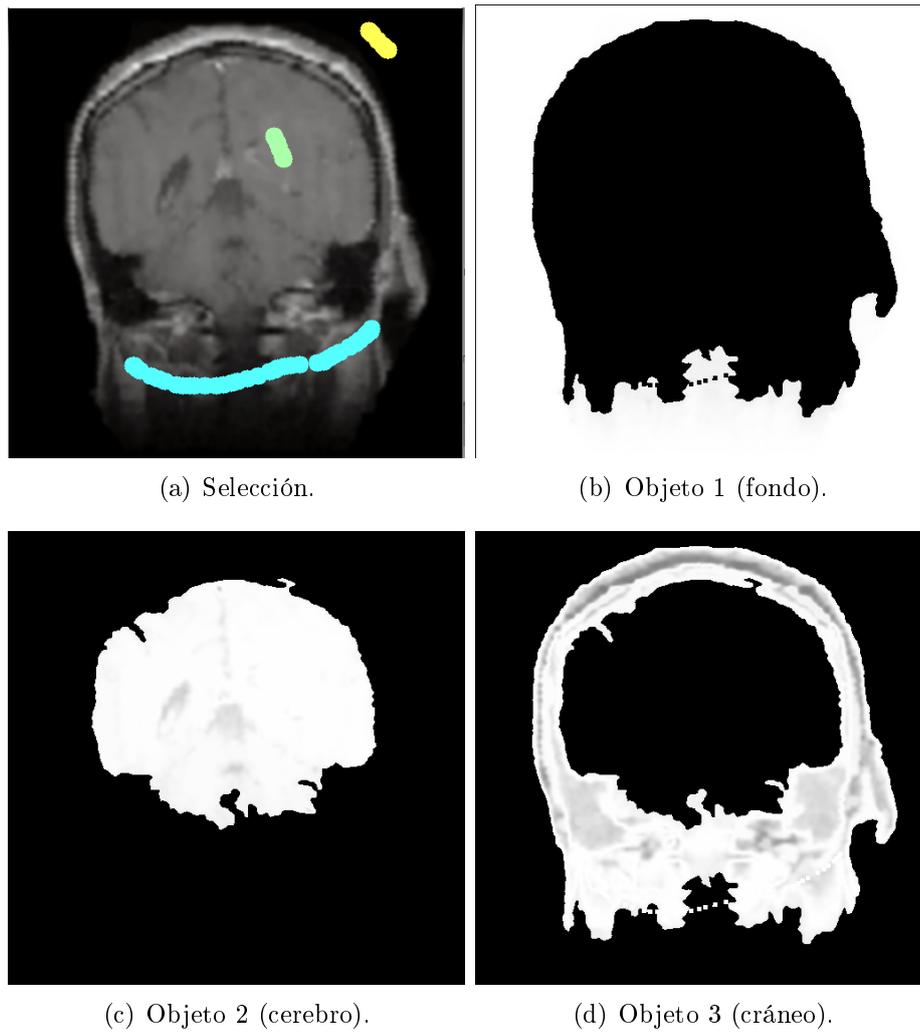


Figura 4.14: Imagen de resonancia magnética del cerebro tipo coronal, la imagen se segmentó en 3 objetos utilizando una 8-adyacencia.

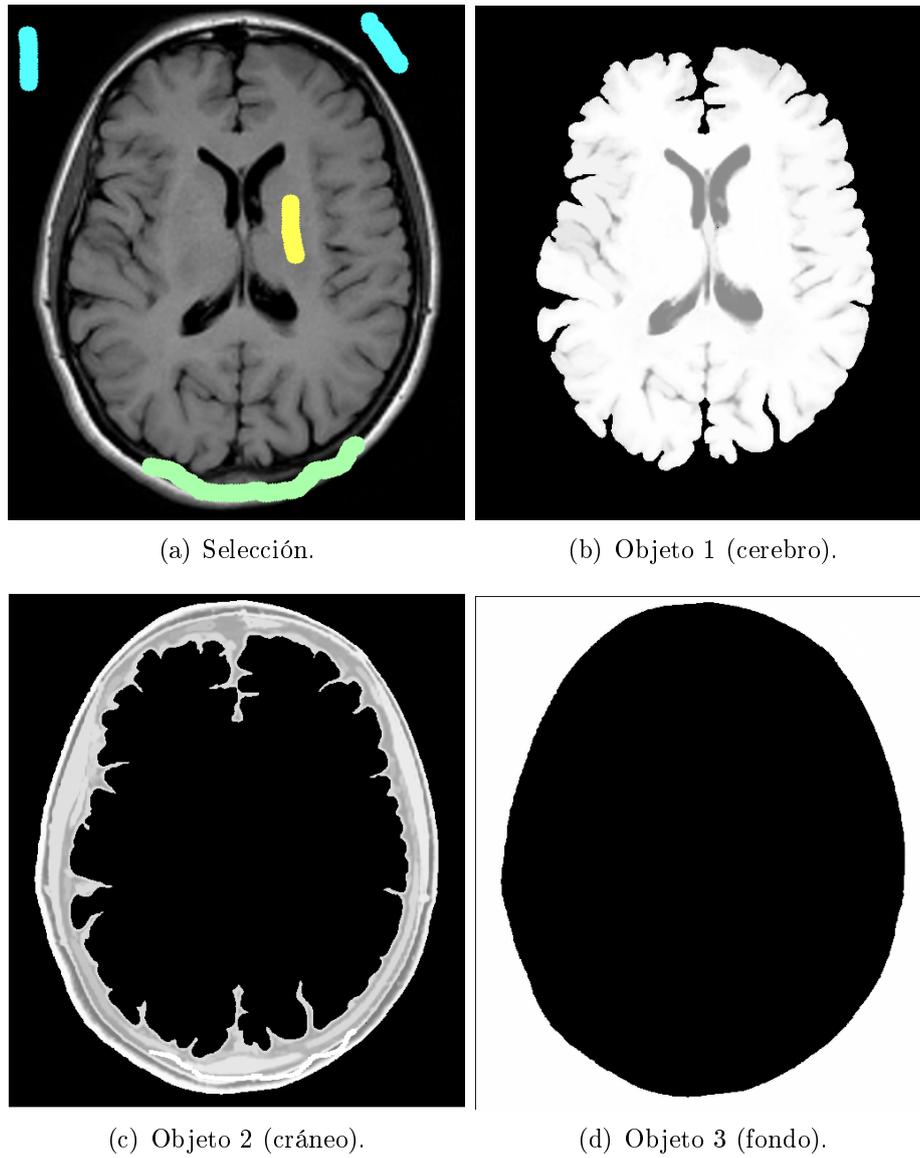


Figura 4.15: Imagen de resonancia magnética corte axial que se segmentó en 3 objetos utilizando una 4-adyacencia.

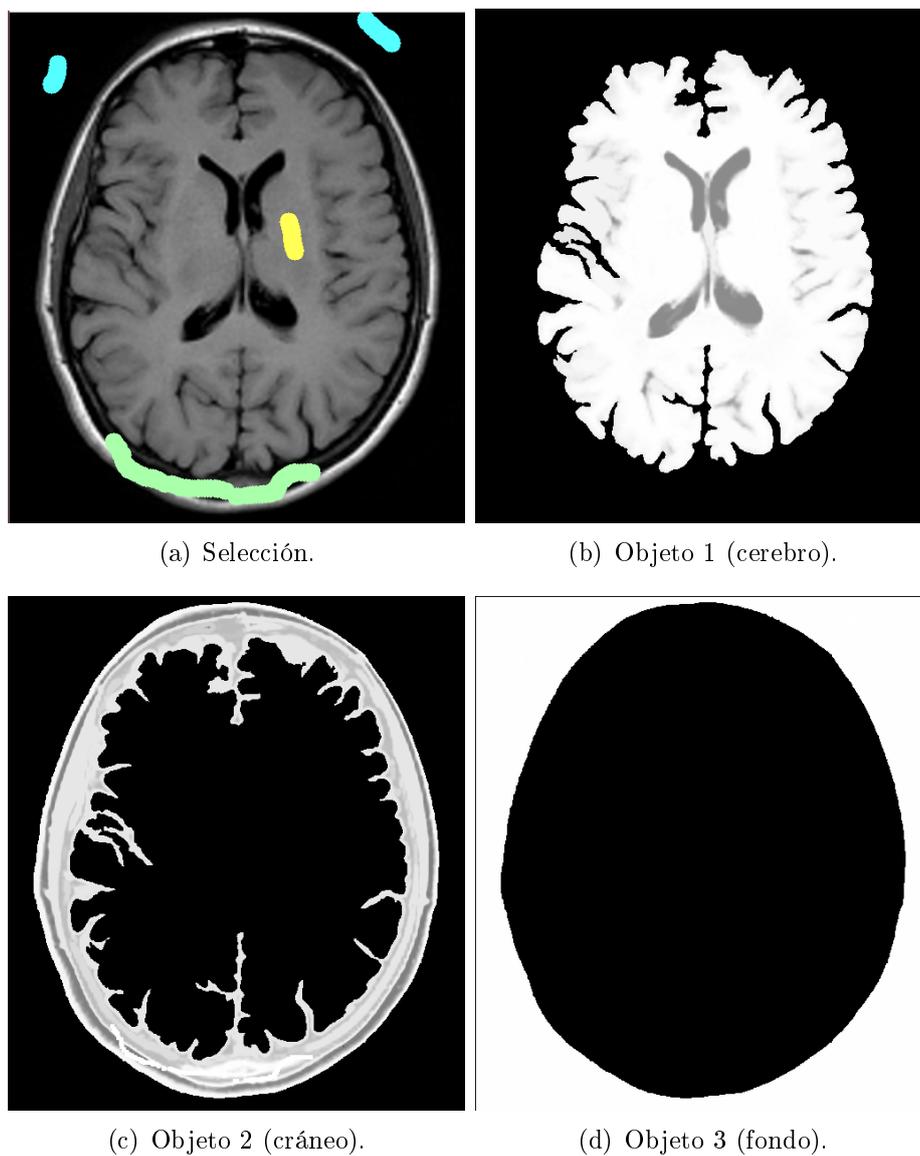


Figura 4.16: Segmentación de una MRI corte axial utilizando una 8-adyacencia y seleccionando 3 objetos. Lo que se desea es distinguir el área del encéfalo.

Conclusiones

El objetivo general de esta tesis fue presentar un algoritmo de segmentación de imágenes en tonos de gris a partir de puntos semilla, utilizando la teoría de conexidad difusa y las técnicas de programación voraz. Para esto fue importante definir las imágenes en el contexto difuso, lo que nos lleva a las definiciones de conjunto difuso y conexidad difusa. En el estudio del procesamiento de imágenes se han mostrado mejores resultados si se toma en cuenta la intensidad de los píxeles y su lugar espacial, que son los conceptos que combina una conexidad difusa: afinidad y adyacencia difusa. Al final, estos conceptos condujeron a la definición de un grafo M -difuso que permite segmentar una imagen digital en subconjuntos difusos.

Se logró observar que definir una afinidad difusa que produzca buenos resultados para la segmentación no es una tarea fácil, en este caso se utilizó la función (4.2) que se define en [16] y aporta la información suficiente para obtener buenos resultados. Esta función de afinidad considera la media y desviación estándar de las intensidades de los píxeles en los objetos, debido a que estos píxeles, por estar en un conjunto difuso, no necesariamente tienen la misma intensidad.

El algoritmo se implementó en C++ utilizando diferentes estructuras de datos, entre ellas una cola de prioridad. También se utilizó el paquete de OpenCV para mostrar la imagen y para que el usuario pueda seleccionar los objetos. La intervención del usuario es casi nula, solo se reduce a la selección de los puntos semilla.

Aunque el procedimiento de las imágenes digitales mediante objetos difusos involucra cálculos de complejidad combinatoria, con la ayuda de conceptos básicos de conjuntos difusos y al algoritmo de Dijkstra, se logra que el algoritmo ejecute las pruebas en segundos (ver Tabla 4.1), debido a que esta técnica evita explorar todos los caminos posibles entre cualesquiera dos elementos. Es por eso que el tiempo de ejecución del algoritmo es logarítmico $O(N(\log N + LM))$. Los resultados de la segmentación se pueden observar en las diferentes pruebas que se hicieron y con diferentes imágenes, desde algún programa de dibujo hasta aquellas que se pueden utilizar en casos reales. Las pruebas que se realizaron fueron con una 4 y 8-adyacencia. Los resultados obtenidos mostraron que no afecta mucho la adyacencia que se utilice, ya que se obtienen resultados similares. Sin embargo, sí afecta la elección de los puntos semilla, es por eso que en algunos casos es importante realizar la selección de puntos semilla más cuidado. En todos los casos de prueba que se realizaron, el tiempo de ejecución requerido fue muy pequeño. El programa solo tardó unos segundos para poder realizar la segmentación como se puede ver en la Tabla 4.1, pero se puede observar que la segmentación con

una 4-adyacencia requiere un menor tiempo de ejecución que una 8-adyacencia. Es razonable pensar esto debido a que con una 4-adyacencia se hacen menos operaciones que con una 8-adyacencia.

Como trabajo futuro se pueden proponer otras afinidades que proporcionen una segmentación más detallada. También se puede probar con otros tipos de imágenes, como imágenes digitales a color, o combinar esta técnica con detección o clasificación de objetos.

Bibliografía

- [1] ANNADURAI, S. *Fundamentals of Digital Image Processing*, first ed. Pearson, 2007.
- [2] ATKINS, M. S., AND MACKIEWICH, B. T. Fully automatic segmentation of the brain in mri. *IEEE transactions on medical imaging* 17, 1 (1998), 98–107.
- [3] AZUELA, J. H. S., AND MORALES, R. R. *Procesamiento y Análisis Digital de Imágenes*. RA-MA S.A. Editorial y Publicaciones, 2011.
- [4] BANDEMER, H., AND GOTTWALD, S. *Fuzzy sets, fuzzy logic, fuzzy methods with applications*, primera ed. J. Wiley, Michigan, 1995.
- [5] BELTING, H., AND ESPINOSA, G. *Antropología de la imagen*. Conocimiento (Katz). Katz, 2007.
- [6] BOUCHET, A., BRUN, M., AND BALLARIN, V. Morfología matemática difusa aplicada a la segmentación de angiografías retinales. *dimensión 3* (2010), 3. Revista Argentina de Bioingeniería.
- [7] BOVIK, A. C. *The Essential Guide to Image Processing*, primera ed. Elsevier Science, Austin, Texas, 2009.
- [8] CHONGFU HUANG, D. R. *Fuzzy sets and fuzzy information granulation theory*. No. v. 3. Beijing normal university press, 2000.
- [9] CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., AND STEIN, C. *Introduction to Algorithms*, tercera ed. MIT Press, Inglaterra, 2009.
- [10] CUEVAS, E., ZALDIVAR, D., AND ROJAS, R. Lvq color segmentation applied to face localization. In *1st International Conference on Electrical and Electronics Engineering (ICEEE)* (2004), pp. 142–146. <http://page.mi.fu-berlin.de/rojas/pub/icaifinal.pdf>, consultado en 2012.
- [11] GARCÍA, A. Y. Prprocesamiento de imágenes para la teledetección de aerosoles en el noroeste de México. Maestría en sistemas computacionales, Instituto Tecnológico de la Paz, México, Diciembre 2014.
- [12] GONZALEZ, R. C., AND WOODS, R. E. *Digital Image Processing*, tercera ed. Pearson Education, 2011.

- [13] GUTIÉRREZ, H. A. Conexidad difusa y segmentación de imágenes digitales, Julio 2008. http://jupiter.utm.mx/~tesis_dig/10632.pdf, consultado en 2012.
- [14] HALMOS, P. *Measure Theory*. Graduate Texts in Mathematics. Springer New York, 2013.
- [15] HAN, S., TAO, W., WANG, D., TAI, X.-C., AND WU, X. Image segmentation based on grabcut framework integrating multiscale nonlinear structure tensor. *IEEE Transactions on Image Processing* 18, 10 (2009), 2289–2302.
- [16] HERMAN, G. T., AND CARVALHO, B. M. Multiseeded segmentation using fuzzy connectedness. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 5 (2001), 460–474.
- [17] JESORSKY, O., KIRCHBERG, K. J., AND FRISCHHOLZ, R. W. Robust face detection using the hausdorff distance. In *International Conference on Audio- and Video-Based Biometric Person Authentication* (2001), Springer, pp. 90–95. <http://www.bioid.com/assets/files/AVBPA01BioID.pdf>, consultado en 2012.
- [18] KOSCHAN, A., AND ABIDI, M. *Digital Color Image Processing*. Wiley, 2008.
- [19] LÓPEZ, G. J. Algoritmo para la identificación de personas basado en huellas dactilares. Master's thesis, Universidad Politécnica de Cataluña, Barcelona, 2009.
- [20] MARROQUÍN, J. L., VEMURI, B. C., BOTELLO, S., CALDERON, E., AND FERNANDEZ-BOUZAS, A. An accurate and efficient bayesian method for automatic segmentation of brain mri. *IEEE Transactions on Medical Imaging* 21, 8 (2002), 934–945.
- [21] PÉREZ, R. P. *Procesado y Optimización de Espectros Raman mediante Técnicas de Lógica Difusa: Aplicación a la identificación de Materiales Pictóricos*. PhD thesis, Universidad Politécnica de Cataluña, España, 2005.
- [22] REINA, D. Fundamentos de matemática difusa, Junio 2008. http://www.konradlorenz.edu.co/images/stories/suma_digital_matematicas/EDICION_09_0I/trabajo_de_grado_daniel_reina.pdf, consultado en 2012.
- [23] ROSENFELD, A. Fuzzy digital topology. *Information and Control* 40, 1 (1979), 76–87.
- [24] SIMÓN, Z. D. *Reconocimiento Automático Mediante Patrones Biométricos de Huella Dactilar*. PhD thesis, Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Madrid, España, 2003. <http://oa.upm.es/79/1/09200327.pdf>, consultado en 2012.
- [25] WADE, N. J., AND FINGER, S. The eye as an optical instrument: From camera obscura to helmholtz's perspective. *Perception* 30, 10 (2001), 1157–1177.