



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA - PROCESAMIENTO DIGITAL DE
SEÑALES

SEGMENTACIÓN DE CRIO-REBANADAS DE RATÓN
USANDO PRINCIPIOS DE LÓGICA DIFUSA

TESIS

QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:

JEISON MÉNDEZ GARCÍA

TUTOR PRINCIPAL:

DR. EDGAR GARDUÑO ÁNGELES

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas - IIMAS

Ciudad Universitaria, CD. MX. - julio del 2017

JURADO ASIGNADO:

Presidente: **Dr. Boris Escalante Ramírez**

Secretario: **Dr. Fernando Arámbula Cosío**

1er. Vocal: **Dr. Edgar Garduño Ángeles**

2do. Vocal: **Dra. Nidiyare Hevia Montiel**

3er. Vocal: **Dr. Bruno Motta de Carvalho**

Lugar donde se realizó la tesis:

Ciudad Universitaria, CD. MX.

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas - IIMAS

TUTOR DE TESIS:

DR. EDGAR GARDUÑO ÁNGELES

FIRMA

Agradecimientos

Quiero agradecer al Posgrado en Ingeniería Eléctrica en su área de Procesamiento Digital de Señales por brindarme la oportunidad de realizar mis estudios. Un agradecimiento especial a todos los profesores con quienes tomé algún curso por ser partícipes de mi formación.

Agradezco también al Dr. Edgar Garduño Ángeles por brindarme la oportunidad de desarrollar este trabajo bajo su tutoría, por sus valiosos aportes y correcciones.

De igual manera, agradezco la participación de los jurados en el comité evaluador de este trabajo, el Dr. Boris Escalante, la Dra. Nidiyare Hevia, el Dr. Fernando Arámbula, el Dr. Bruno Motta de Carvalho y el Dr. Edgar Garduño.

Este trabajo contó con la participación del Instituto de Imagenología Biológica y Médica (*IBMI*, por sus siglas en inglés) ubicado en Múnich-Alemania. Agradezco especialmente al Dr. Vasilis Ntziachristos por su amable disposición y valiosos aportes al desarrollo del trabajo, al igual que a su equipo de trabajo por la asesoría técnica y evaluación de la calidad de las anotaciones manuales hechas a las imágenes realizadas por integrantes del grupo *VISP* a quienes de igual manera agradezco enormemente por su participación dentro del proyecto.

Quiero agradecer al Consejo Nacional de Ciencia y Tecnología (*CONACYT*) por el apoyo económico sin el cual hubiera sido imposible llevar a cabo mis estudios, mediante la beca asignada a través de la Convocatoria de Becas Nacionales 2015 segundo período identificada con el número 291025.

Nada de esto hubiese sido posible sin el apoyo de mi compañera de viaje en esta vida, mi esposa Maribel Vanegas, por su compañía incondicional, por su paciencia y consejos en los momentos difíciles, pero principalmente por ser el maravilloso ser que llena de alegría mi existencia.

Contenido

1	Introducción	1
2	Marco conceptual	7
2.1	Imágenes digitales	7
2.2	Color	9
2.2.1	Transformaciones lineales	10
2.2.2	Transformaciones no-lineales	12
2.2.3	Espacio híbrido de color	16
2.3	Segmentación de imágenes digitales	17
2.3.1	Técnicas basadas en puntos - Umbralización	17
2.3.2	Técnicas basadas en bordes	18
2.3.3	Técnicas basadas en regiones	19
2.4	Segmentación difusa	21
2.5	Función de afinidad	26
2.5.1	Incorporación de información de color	27
2.6	Algoritmo de segmentación multi-objeto	28
3	Trabajo realizado, experimentos y resultados	31
3.1	Conjunto de imágenes disponibles	31
3.2	Conjunto de datos <i>Ground-Truth</i>	32
3.3	Herramientas de software	34
3.3.1	Módulo para selección de semillas	34
3.3.2	Módulo para realizar segmentación	37
3.3.3	Módulo para visualización de objetos segmentados	39
3.3.4	Módulo para evaluación de resultados	39
3.4	Selección de umbral	41

3.5	Características de los conjuntos de semillas	43
3.6	Segmentación en diferentes espacios de color	44
3.6.1	Interior del estómago	46
3.6.2	Hígado	48
3.6.3	Páncreas	51
3.6.4	Bazo	52
3.6.5	Riñones	54
4	Conclusiones	57
4.1	Trabajo futuro	59
A	Renderizado directo de volumen	61
B	Ejemplo archivo de salida - Experimento varios espacios de color	67
	Referencias	71

Lista de Tablas

- 3.1 Lista de identificadores numéricos de componentes de color para formar espacio híbrido 38
- 3.2 Recomendación de colores estándar e híbrido realizada por el programa de selección de semillas para los diferentes conjuntos de semillas 46
- 3.3 Ejemplo de los resultados de segmentación para el hígado 47

Lista de figuras

2.1	Representación de espacio de color RGB [1].	9
2.2	Representación de espacio de color HSI [1].	13
2.3	Esquema funcional en una dimensión del método <i>watershed</i> . Se puede considerar que al agua fluye y llena las cuencas dando origen a las diferentes regiones u objetos segmentados. Obsérvese que las cuencas están separadas por líneas divisorias, conocidas como <i>watershed lines</i> , las cuales previenen que se “mezclen” las regiones.	20
2.4	Representación de una imagen de tamaño 3×3 como una gráfica donde cada <i>spel</i> corresponde a un nodo y las aristas representan un tipo de adyacencia entre los nodos (π_1 en este caso), cortesía de [2]	23
2.5	El algoritmo de segmentación difusa realiza un mapeo de la gráfica que representa la imagen a una gráfica difusa donde a cada nodo corresponde el valor σ_0^c y cada arista tiene un peso $\psi(\mathbf{c}, \mathbf{d})$, cortesía de [2]	24
3.1	Ejemplo de características no deseadas en el conjunto de datos	32
3.2	Ejemplo de construcción de volumen de datos a ser usado como <i>ground-truth</i>	33
3.3	Ventana de control de semillas del programa para selección de semillas.	35
3.4	Imagen de corte axial de abdomen de ratón con 6 conjuntos de semillas ubicadas	36
3.5	Resultado de segmentación con 6 clases llevada a cabo en el espacio <i>YCbCr</i> usando adyacencia π_1 . Cada clase es un una región de interés de la Figura 3.4.	40

3.6	Evaluación de resultado de segmentación. Se muestra un corte del hígado segmentado (verde) superpuesto a la misma rebanada del hígado de la imagen referencia (rojo). Para este ejemplo, con un <i>umbral de confianza</i> de 0.70, los valores de los coeficientes de <i>Sorensen-Dice</i> , <i>sensibilidad</i> y <i>especificidad</i> fueron respectivamente 0.6042, 0.5534 y 0.9658; lo cual es una indicación de que comparando las imágenes segmentada y referencia se presenta una baja tasa de <i>positivos verdaderos</i> y una tasa alta de <i>negativos verdaderos</i>	41
3.7	Curvas <i>ROC</i> para varios órganos y configuraciones de parámetros de segmentación.	42
3.8	Comparación de resultados de segmentación usando varios conjuntos de semillas y diferentes espacios de color	45
3.9	Resultados segmentación del interior del estómago con adyacencias (a) π_1 y (b) π_2	47
3.10	Anotación manual del interior del estómago.	48
3.11	Renderizado para el interior del estómago. a) Imagen referencia (<i>ground-truth</i>) y b) Superposición del resultado de la segmentación con la imagen referencia. La segmentación fue realizada en el espacio de color $i_1i_2i_3$ usando adyacencia π_2	49
3.12	Resultados segmentación del hígado con adyacencias (a) π_1 y (b) π_2 . . .	49
3.13	Anotación manual del hígado.	50
3.14	Renderizado para el hígado. a) Imagen referencia (<i>ground-truth</i>) y b) Superposición del resultado de la segmentación con la imagen referencia. La segmentación fue realizada en el espacio de color $YCbCr$ usando adyacencia π_1 . En las regiones blancas se presenta mayor coincidencia entre las imágenes.	50
3.15	Resultados segmentación del páncreas con adyacencias (a) π_1 y (b) π_2 . . .	51
3.16	Anotación manual del páncreas.	52
3.17	Renderizado para el páncreas. a) Imagen referencia (<i>ground-truth</i>) y b) Superposición del resultado de la segmentación con la imagen referencia. La segmentación fue realizada en el espacio de color $i_1i_2i_3$ usando adyacencia π_1	52
3.18	Resultados segmentación del bazo con adyacencias (a) π_1 y (b) π_2	53
3.19	Anotación manual del bazo.	53

3.20	Renderizado para el páncreas. a) Imagen referencia (<i>ground-truth</i>) y b) Superposición del resultado de la segmentación con la imagen referencia. La segmentación fue realizada en el espacio de color <i>CIELab</i> usando adyacencia π_2	54
3.21	Resultados segmentación de los riñones con adyacencias (a) π_1 y (b) π_2	55
3.22	Anotación manual de los riñones.	55
3.23	Renderizado para los riñones. a) Imagen referencia (<i>ground-truth</i>) y b) Superposición del resultado de la segmentación con la imagen referencia. La segmentación fue realizada en el espacio de color <i>HSV</i> usando adyacencia π_2	56
A.1	Intuición del proceso de <i>Volume rendering</i> , cortesía de [2]	62

Capítulo 1

Introducción

La técnica de *imagenología optoacústica* (conocida también como *imagenología fotoacústica*) ha presentado una capacidad notable para permitir visualizar tejido biológico en variadas aplicaciones tanto clínicas como en investigación en animales pequeños [3, 4, 5]. Esta técnica de imagenología está basada en los principios del llamado *efecto fotofónico* el cual fue descrito alrededor de 1880 por Alexander Graham Bell y su asistente [6], quienes reconocieron que ondas de sonido pueden ser generadas desde un objeto a partir de la absorción de fuentes de luz moduladas y su conversión en calor. Hoy en día los términos *optoacústico* y *fotoacústico* son usados de igual manera para describir el efecto de generación de ondas acústicas producidas a partir de la absorción de luz pulsada [7, 8].

El campo de aplicación de la imagenología optoacústica en la actualidad presenta un gran crecimiento debido a nuevos enfoques en la aplicación de la técnica, lo cual se debe en gran medida al desarrollo de tecnología de láser pulsado lo cual se presentaba como una limitación de la técnica en sus principios [9]. Otra característica que ha propiciado la diversificación de la técnica de imagenología es su naturaleza híbrida, ya que la excitación proviene de la energía portada por fotones mientras que la detección del efecto es acústica, esto hace entonces que la técnica se enriquezca tanto de la versatilidad del contraste óptico como de la alta resolución espacial asociada con la baja dispersión que se presenta en la propagación de las ondas ultrasónicas comparadas con la propagación de los fotones [10]. Lo anterior se suma al relativo bajo costo de implementación y reducidos efectos adversos al tejido bajo estudio, dado que se usa como fuente de excitación una radiación no ionizante de baja potencia.

Una de las aplicaciones importantes de la imagenología optoacústica se da en el

estudio de modelos animales, en particular de ratones de laboratorio. En estos casos los ratones son sometidos a modificaciones genéticas con el fin de causarles patologías que puedan ser detectadas mediante la aplicación de la técnica fotoacústica. Luego de ser sometidos a dicha técnica de imagenología, los ratones son congelados rápidamente y posteriormente rebanados para obtener imágenes digitales a color de alta resolución de su estructura interna (una imagen por cada uno de los cortes). Dado que el conjunto de imágenes a color permiten construir un volumen tridimensional discretizado, los hallazgos obtenidos mediante la técnica de imagenología pueden ser comparados con la representación digital de la anatomía del animal.

Sin embargo, la inspección de la anatomía del ratón en las imágenes a color de alta resolución se presenta como una tarea tediosa que demanda la aplicación de una delineación previa, o segmentación, de las estructuras de interés en cada una de las imágenes del conjunto donde éstas se encuentran presentes. Un hecho que dificulta la anotación manual de dichas estructuras es la falta de definición de los límites de cada órgano debido a la presencia de sangre en casi todos ellos. En este sentido, tener un algoritmo de segmentación que permita realizar la tarea sería muy deseable.

Los algoritmos de segmentación por computadora pueden ser divididos de acuerdo a cómo ellos resuelven el problema en las siguientes categorías: basados en puntos (*point-based*), basados en bordes (*edge-based*) y basados en regiones (*region-based*).

Los algoritmos basados en puntos hacen una decisión local acerca de la pertenencia de un determinado punto al objeto. Esta decisión puede estar basada únicamente en el valor de densidad del punto o los valores de densidad de un vecindario pequeño alrededor del punto. Un algoritmo basado en puntos que es ampliamente usado es el de *umbralización*. En éste, el usuario selecciona uno o dos valores de intensidad del objeto a ser segmentado y posteriormente todos los elementos de imagen cuyos valores se encuentren en el rango establecido por el usuario son considerados como pertenecientes al objeto. Sin embargo, los resultados arrojados por estos algoritmos son sensibles tanto al ruido como a la iluminación no-homogénea y no son apropiados para segmentar objetos con presencia de texturas [11].

Por su parte, los algoritmos basados en bordes usualmente trabajan en dos pasos, primero detectando los bordes en la imagen y después agrupándolos con base en un conocimiento previo de la forma del objeto esperado. Algunos procedimientos de este tipo incluyen el uso del operador gradiente, *Laplacianos* o el operador *Canny* [12]; por su parte, el enlace de los bordes puede ser implementado de manera local buscando vecindades pequeñas de los elementos de imagen o de manera global usando la

transformada de Hough, por ejemplo. Otros algoritmos basados en bordes usan modelos de contorno activo (*ACM*), como *Snakes* [13]. Los *Snakes* son *splines* que se modifican a partir de equilibrios de energías externas, relacionadas con los bordes presentes en la imagen, e internas, relacionadas con restricciones a la deformación del *spline*.

Finalmente, los algoritmos basados en regiones están divididos en algoritmos de crecimiento de regiones y separar-y-componer (*split-and-merge*). Los algoritmos basados en crecimiento de regiones, como su nombre lo sugiere, inician con unos puntos seleccionados por el usuario, conocidos como *semillas*, los cuales permiten identificar las regiones iniciales dentro de los límites de los objetos de interés a ser segmentados; posteriormente, a partir de tales *semillas* las regiones pertenecientes a cada objeto crecen basándose en algunas reglas pre-definidas hasta que todos los puntos de la imagen son asignados a una determinada clase u objeto. Mientras que los algoritmos separar-y-componer consisten en separar inicialmente la imagen en regiones no-articuladas, para posteriormente separar y/o componer dichas regiones de manera repetida hasta que algunas condiciones pre-definidas se satisfagan.

Adicionalmente, los algoritmos de segmentación por computadora pueden ser divididos también según el nivel de interacción con el usuario en las tres categorías siguientes: manual, semi-automático y automático.

En los algoritmos manuales se utiliza un programa con interfaz gráfica que asiste al usuario de una manera muy básica en la tarea de separar los objetos, permitiéndole a éste trazar líneas y rellenar las regiones de interés. La tarea de separación de objetos mediante este enfoque puede ser realizada de manera satisfactoria dado que el sistema humano de visión permite “distinguir” fácilmente entre diferentes objetos cuando, en lugar de los niveles de intensidad, la diferencia entre los objetos consiste en alguna propiedad de textura, o cuando la imagen se encuentra afectada por ruido y/o una iluminación no-homogénea [14]; sin embargo, el resultado también está sujeto al entrenamiento del usuario en la tarea de identificar objetos cuando se trata de tareas muy específicas, al igual que a la interactividad que ofrezca la herramienta. Además de lo mencionado, los algoritmos manuales resultan poco recomendables cuando el conjunto de datos es ruidoso y de un gran tamaño ya que puede resultar en un proceso lento, tedioso y sesgado.

En el caso de los algoritmos automáticos, el proceso de segmentación se lleva a cabo en su totalidad sin intervención del usuario. Sin embargo, debido a la falta de intervención por parte del usuario, esta clase de algoritmos aunque pueden ser útiles en algunos casos pueden fallar en muchos otros ya que dependen enteramente de la

información que se pueda obtener de la imagen misma o información incorporada al algoritmo antes de su ejecución.

Con un nivel de interacción media entre el usuario y el algoritmo, se presentan los algoritmos semi-automáticos. En el caso de algunos algoritmos de crecimiento de regiones, cuando operan en esta modalidad, la interacción del usuario consiste típicamente en proporcionar las “semillas”, las cuales corresponden a puntos en la imagen que a su juicio pertenecen al objeto de interés a ser segmentado. Con esta información el algoritmo continúa desde esos puntos por si solo ejecutando la tarea de segmentación. En este tipo de algoritmos, típicamente el usuario proporciona guía al algoritmo o puede proporcionarle también información necesaria para su ejecución.

Mencionado lo anterior, y dadas las características de los volúmenes de datos a procesar, se puede ver que los algoritmos semi-automáticos representan una solución apropiada al problema de segmentar las estructuras internas en las imágenes de cortes de ratón, dado que es conveniente hacer uso de la capacidad del usuario para identificar los objetos de interés a ser segmentados, pero dada la cantidad de datos se necesita que la tarea de asignar cada punto de la imagen a alguno de los objetos de interés esté a cargo de un algoritmo. Para llevar a cabo esta tarea se propone el uso de la técnica de segmentación de imágenes mediante un algoritmo que usa principios de *lógica difusa*. Inicialmente, Rosenfeld sugirió en [15] la aplicación de conceptos de *lógica difusa* en el campo de procesamiento digital de imágenes. Sin embargo, el uso de los conceptos de *conectividad difusa* y *afinidad difusa* aplicado a segmentación de imágenes basado en crecimiento de regiones fue presentado por Udupa y Samarasekera en [16]. El enfoque usado en el presente trabajo fue introducido por Herman y otros en [17] y está basado en la teoría presentada en [16], pero generalizada a espacios digitales arbitrarios [18]; además se tiene en cuenta también la propuesta de mejoramiento presentada en [19] y la aplicación de la técnica de segmentación a secuencias de imágenes a color presentada en [20]. En este algoritmo semi-automático de crecimiento de regiones, el usuario proporciona como información de entrada al algoritmo las coordenadas de puntos sobre la imagen los cuales considera con certeza que pertenecen al objeto de interés a segmentar. Esta información es usada en la llamada función de *afinidad difusa* para caracterizar los objetos a ser segmentados y a partir de allí realizar la tarea de segmentación sin más intervención por parte del usuario. El algoritmo permite desde las “semillas” y mediante el uso de *lógica difusa*, hacer crecer las regiones que pertenecen a los objetos a segmentar. Este algoritmo ha presentado buenos resultados en imágenes con diferentes características, entre ellas pobre relación señal a ruido, falta

de homogeneidad en la densidad, y presencia de texturas entre otras [19, 20, 21].

En este trabajo se plantea probar la utilidad del algoritmo de segmentación difusa para separar estructuras internas de ratón a partir de imágenes a color de alta resolución de crio-rebanadas.

Capítulo 2

Marco conceptual

2.1 Imágenes digitales

De manera que puedan ser almacenados y procesados por un computador, los datos provenientes de campos escalares ($f : \mathbb{R}^3 \rightarrow \mathbb{R}$) o campos vectoriales ($f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$) deben ser sometidos a un proceso de *discretización*. La discretización de imágenes implica dos procesos, a saber, *muestreo* y *cuantización* [11]. El primero de los procesos consiste en la *digitalización* de los valores de coordenadas espaciales, mientras el segundo consiste en la *digitalización* de los valores a la salida del campo escalar o vectorial.

La información presentada a continuación se hace para el caso de campos escalares, sin embargo puede ser extendida a espacios vectoriales.

Para llevar a cabo la *discretización* de una función f según [18, 22], dado la 3-tupla $\mathbf{k} = (k_1, k_2, k_3)^T$ con $\mathbf{k} \in \mathbb{Z}^3$, se hace uso de un conjunto o rejilla $\Gamma_\Delta \subset \mathbb{R}^3$ definido como sigue:

$$\Gamma_\Delta = \{(\Delta k_1, \Delta k_2, \Delta k_3)^T | k_i \in \mathbb{Z}, \text{ para } 1 \leq i \leq 3\}, \quad (2.1)$$

donde Δ es un número real positivo conocido como *intervalo de muestreo*.

Ahora bien, dado un punto $\mathbf{g} \in \Gamma_\Delta$ tal que $\mathbf{g} = \Delta \mathbf{k}$, el vecindario de *Voronoi* en Γ_Δ de cualquier elemento \mathbf{g} se denomina *vóxel cúbico* y puede ser definido como:

$$Voxel(\mathbf{g}) = \left\{ \mathbf{x} \in \mathbb{R}^3 \mid \text{para todo } \mathbf{h} \in \Gamma_\Delta, \|\mathbf{x} - \mathbf{g}\| \leq \|\mathbf{x} - \mathbf{h}\| \right\}. \quad (2.2)$$

Es decir, el vecindario de *Voronoi* del elemento \mathbf{g} consta de todos aquellos puntos que están más cerca de $\mathbf{g} \in \Gamma_\Delta$ que de cualquier otro punto \mathbf{h} también perteneciente

a la rejilla cúbica Γ_Δ . Por ejemplo, para $\Delta > 0$, el vecindario de *Voronoi* de $(\Delta k_1, \Delta k_2, \Delta k_3)^T$ es

$$\{(x_1, x_2, x_3)^T \in \mathbb{R}^3 \mid \max(|x_1 - \Delta k_1|, \max(|x_2 - \Delta k_2|, |x_3 - \Delta k_3|)) \leq \frac{\Delta}{2}\}.$$

Este conjunto representa un cubo cerrado, el cual contiene los bordes, cuyos lados tienen una longitud Δ y está centrado en el punto $(\Delta k_1, \Delta k_2, \Delta k_3)^T$ [18].

Luego, la *digitalización* de la función f sobre los puntos de la rejilla definida en (2.1) usando el intervalo de muestreo Δ , puede ser definida como:

$$p(\Delta \mathbf{k}) = \frac{1}{\Delta^3} \int_{Voxel(\Delta \mathbf{k})} f(\mathbf{x}) d\mathbf{x}. \quad (2.3)$$

Este proceso produce una *imagen digital* en 3 dimensiones o imagen 3D la cual puede ser definida por

$$v(\mathbf{x}) = \begin{cases} p(\Delta \mathbf{k}), & \text{si } \mathbf{x} \in Voxel(\Delta \mathbf{k}) \text{ para algún } \Delta \mathbf{k} \in \Gamma_\Delta, \\ 0, & \text{en cualquier otro caso.} \end{cases} \quad (2.4)$$

La *imagen digital* puede ser vista también como un *espacio digital* de la forma (V, π) , donde V es un conjunto (por ejemplo, de *vóxeles*) y π representa una relación binaria sobre V que determina la manera en que están conectados los elementos de este conjunto [14]. Además, se define $N(\mathbf{c})$ como el *vecindario* del vóxel $\mathbf{c} \in \mathbb{Z}^3$, el cual consta del mismo elemento \mathbf{c} y todos los elementos $\mathbf{d} \in V$ tales que $(\mathbf{c}, \mathbf{d}) \in \pi$; si esta última relación se cumple, entonces se puede decir que \mathbf{c} y \mathbf{d} son *adyacentes*.

En el presente trabajo, se explorarán los resultados con dos tipos de definiciones para la *adyacencia* de los *vóxeles* en un vecindario $N(\mathbf{c})$, así

$$(\mathbf{c}, \mathbf{d}) \in \pi_1 \Leftrightarrow |\mathbf{c} - \mathbf{d}| = 1 \quad (2.5)$$

y

$$(\mathbf{c}, \mathbf{d}) \in \pi_2 \Leftrightarrow 1 \leq |\mathbf{c} - \mathbf{d}| \leq \sqrt{2}. \quad (2.6)$$

Se puede ver que $\pi_1 \equiv$ “cara” y $\pi_2 \equiv$ “cara-borde”. Las adyacencias mencionadas constan de un vecindario $N(\mathbf{c})$ con 6 y 18 elementos respectivamente.

Dado que los *vóxeles* y *píxeles* (nombre dado al vecindario de *Voronoi* cuando la rejilla $\Gamma_\Delta \subset \mathbb{R}^2$) hacen parte de un conjunto más amplio que define los elementos

constituyentes de una imagen denominados *spels* (contracción en inglés para *spatial elements*), en el presente documento se hará referencia a éstos últimos cuando el algoritmo o proceso mencionado pueda ser aplicado tanto a *vóxeles* como a *píxeles*.

2.2 Color

El color es una percepción visual generada en el cerebro humano (y de otros animales) a partir de la combinación de señales nerviosas generadas en receptores ubicados en la retina (*conos*); éstos son de tres tipos y cada uno es más sensible a un cierto rango de longitudes de onda del espectro electromagnético visible. Los colores asociados a esas longitudes de onda son los llamados *colores aditivos primarios* (rojo, verde y azul) y a partir de su combinación es posible percibir una gran gama de colores en la banda visible del espectro.

Toda la gama de colores que se pueden obtener con la combinación de los colores primarios se encuentran representados en el llamado espacio de color *RGB* (siglas en inglés de los colores *rojo*, *verde* y *azul*) [1], ver Figura 2.1. Existen otras representaciones de color, por ejemplo, el substractivo *CMY* (siglas en inglés de los colores *cian*, *magenta* y *amarillo*), de manera que un punto en el espacio *RGB* puede transformarse a otro punto en alguno de los otros espacios mediante transformaciones lineales o no-lineales.

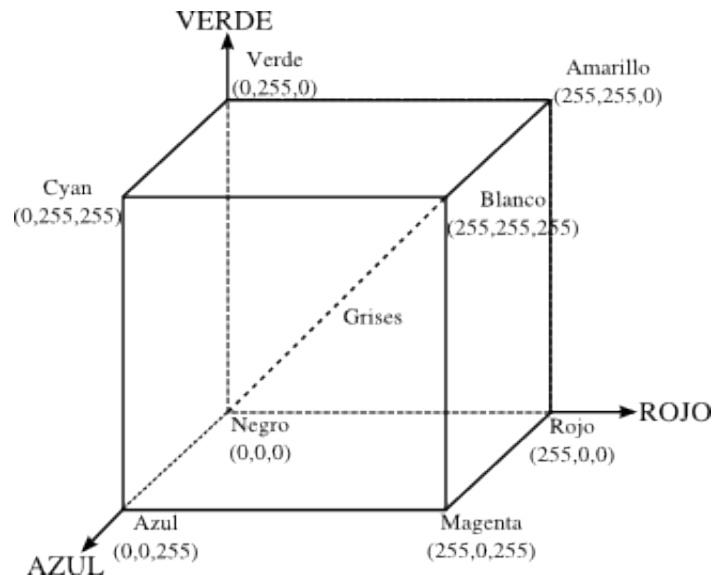


Figura 2.1: Representación de espacio de color RGB [1].

Mientras que el espacio de color *RGB* pareciese el espacio apropiado para

llevar a cabo tareas de procesamiento de imágenes, se ha demostrado que existen representaciones en otros espacios de color, por mencionar algunos: HSI , $YCbCr$ e $i_1i_2i_3$ entre otros, que pueden ofrecer mejores resultados dependiendo de las características de la imagen [23]; sin embargo, para llevar a cabo la tarea de segmentación de imágenes a color no existe una representación que sea dominante sobre los demás en todos los casos. La selección del mejor espacio de color es una de las dificultades en segmentación de imágenes a color [24].

Aunque la representación en el espacio RGB es recomendable para el despliegue de imágenes a color y es ampliamente usado en sistemas de televisión y cámaras digitales, no resulta tan útil en la mayoría de los casos para tareas de segmentación y análisis debido a la alta correlación entre sus componentes [25, 26]. Dado que es posible transformar un punto en el espacio RGB en otro punto en un espacio de color diferente, se presentan a continuación algunas de esas transformaciones.

2.2.1 Transformaciones lineales

Espacio de color YIQ (\mathcal{C}_{YIQ})

Este es un espacio de color usado para codificar información de señal de televisión en el sistema $NTSC$. Las componentes en esta representación pueden ser obtenidas a partir de los valores de color representados en el espacio rgb ($\mathcal{C}_{rgb} \subset \mathbb{R}^3$) como sigue según [27, 28]:

$$\begin{bmatrix} c_Y \\ c_I \\ c_Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.253 & -0.312 \end{bmatrix} \begin{bmatrix} c_r \\ c_g \\ c_b \end{bmatrix}, \quad (2.7)$$

donde el vector $\bar{c}_{YIQ} = (c_Y, c_I, c_Q)^T \in \mathcal{C}_{YIQ} \subset \mathbb{R}^3$ y el vector $\bar{c}_{rgb} = (c_r, c_g, c_b)^T \in \mathcal{C}_{rgb}$. Las componentes de \mathcal{C}_{rgb} se pueden obtener a partir del vector $\bar{c}_{RGB} \in \mathcal{C}_{RGB} \subset \mathbb{Z}^3$ como sigue:

$$c_r = \frac{c_R}{L-1}, \quad (2.8)$$

$$c_g = \frac{c_G}{L-1}, \quad (2.9)$$

$$c_b = \frac{c_B}{L-1}, \quad (2.10)$$

con las componentes $c_R, c_G, c_B \in [0, L-1]$; donde L representa el número de diferentes valores que se pueden representar según el ancho de palabra (número de bits) usado en la representación de los datos, por ejemplo $L = 256$ para una representación de 8 bits. En este espacio de color la componente c_Y representa la luminosidad, mientras que las componentes c_I y c_Q contienen la información del tono (*chrominancia*).

Espacio de color YUV (\mathcal{C}_{YUV})

Esta representación es usada en la codificación de color para sistemas de televisión en el sistema *PAL*. La transformación se obtiene también según [27, 28] a partir de la representación del color en el espacio *rgb* así:

$$\begin{bmatrix} c_Y \\ c_U \\ c_V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} c_r \\ c_g \\ c_b \end{bmatrix}, \quad (2.11)$$

donde el vector $\bar{c}_{YUV} = (c_Y, c_U, c_V)^T \in \mathcal{C}_{YUV} \subset \mathbb{R}^3$ y $\bar{c}_{rgb} \in \mathcal{C}_{rgb}$ es como se mostró en (2.8)-(2.10).

En este espacio de color, de manera similar que en el espacio *YIQ*, la componente c_Y representa la luminosidad, mientras las componentes c_U y c_V contienen información de la *chrominancia*.

Espacio de color YCbCr (\mathcal{C}_{YCbCr})

Éste es un espacio de color utilizado en sistemas de fotografía y video digital. Permite definir el color en términos de un componente de *luminosidad* (c_Y) y dos componentes de *chrominancia* (c_{Cb}, c_{Cr}). Los valores de color en este espacio según se muestra en [27, 29], se obtienen a partir de las componentes en el espacio *rgb* así:

$$\begin{bmatrix} c_Y \\ c_{Cb} \\ c_{Cr} \end{bmatrix} = \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} c_r \\ c_g \\ c_b \end{bmatrix} + \begin{bmatrix} L/16 \\ L/2 \\ L/2 \end{bmatrix}, \quad (2.12)$$

donde $\bar{c}_{YCbCr} = (c_Y, c_{Cb}, c_{Cr})^T \in \mathcal{C}_{YCbCr} \subset \mathbb{R}^3$ y \bar{c}_{rgb} es como se mostró en (2.8), (2.9) y (2.10); además con $L = 256$ para un tamaño de palabra de 8 bits.

Espacio de color $i_1i_2i_3$ ($\mathcal{C}_{i_1i_2i_3}$)

Este espacio fue reportado en [30] como un conjunto efectivo de características de color. Fue encontrado de manera iterativa a partir del método de *region-splitting* (ver Sección 2.3.3) y transformaciones *Karhunen-Loeve* a los componentes c_R , c_G y c_B . El conjunto reportado es como sigue:

$$c_{i_1} = \frac{c_R + c_G + c_B}{3}, \quad (2.13)$$

$$c_{i_2} = \frac{c_R - c_B}{2}, \quad (2.14)$$

$$c_{i_3} = \frac{2c_G - c_R - c_B}{4}. \quad (2.15)$$

En dicho trabajo fue afirmado que el espacio de color $i_1i_2i_3$ presentó menor complejidad respecto a las operaciones de la transformación, además de que fue más efectivo en términos de la calidad de la segmentación comparado con otros siete espacios de color (RGB , YIQ , HSI , rgb , CIE_{XYZ} , $CIEL^*u^*v^*$ y $CIEL^*a^*b^*$).

2.2.2 Transformaciones no-lineales

Espacio de color HSI (\mathcal{C}_{HSI})

Este espacio de color descrito por *tono-saturación-intensidad* (HSI , por sus siglas en inglés) es usado con frecuencia en el procesamiento de imágenes debido a que su representación resulta más acorde con respecto al sistema de visión humano [31, 23, 32]. Basadas en este mismo sistema, existen variaciones tales como el sistema HSB (siglas en inglés de tono-saturación-brillo), HSL (siglas de tono-saturación-luminosidad) y HSV (tono-saturación-valor) [33, 34, 35].

En el espacio HSI , la intensidad (relacionada con el brillo de la imagen) es determinada por la cantidad de luz en ésta, mientras la información del color es representada por los valores de tono y saturación. El tono representa los colores básicos y está relacionado con la distribución de las longitudes de onda del espectro visible. La saturación es una medida de la pureza del color, es decir, una medida de la cantidad de luz blanca mezclada con el tono puro. La representación geométrica de este espacio de color es como se muestra en la Figura 2.2.

Uno de los usos en el procesamiento de imágenes de este espacio de color, se

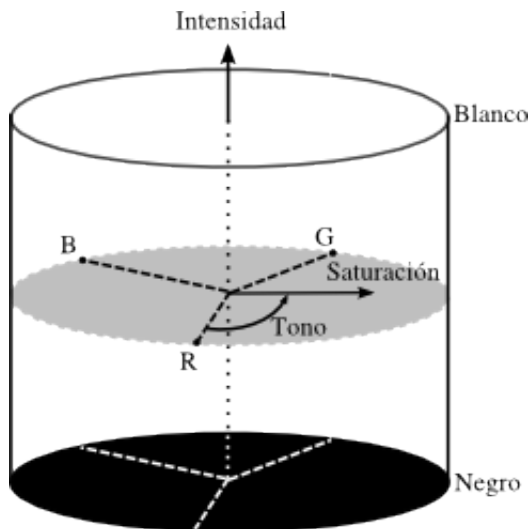


Figura 2.2: Representación de espacio de color HSI [1].

encuentra en que algoritmos de procesamiento construidos para operar sobre imágenes en niveles de gris pueden ser usados sobre el componente de intensidad. Por otra parte, para segmentar objetos de diferente color pueden ser aplicados algoritmos de segmentación a la componente del tono. Además, este espacio de color resulta particularmente eficiente cuando la imagen de interés presenta una iluminación no-uniforme debido a que el tono es independiente de los valores de intensidad [28]. Sin embargo, una de las desventajas de la componente del tono en esta representación es la existencia de una singularidad del tipo no-removible cerca al eje del cilindro (saturación cercana a cero), en donde pequeños cambios en los valores de las componentes c_r , c_g y c_b pueden causar grandes cambios en los valores transformados en este espacio de color [28].

Para los tres espacios de color (HSI , HSL , HSV) la componente de tono (c_H) se obtiene según:

$$c_H = \begin{cases} 60^\circ \times \text{mod}\left(\frac{c_g - c_b}{V_{max} - V_{min}}, 6\right), & \text{si } V_{max} = c_r, \\ 60^\circ \times \left(\frac{c_b - c_r}{V_{max} - V_{min}}\right) + 120^\circ, & \text{si } V_{max} = c_g, \\ 60^\circ \times \left(\frac{c_r - c_g}{V_{max} - V_{min}}\right) + 240^\circ, & \text{si } V_{max} = c_b, \end{cases} \quad (2.16)$$

donde \bar{c}_{rgb} es como se mostró en (2.8)-(2.10), mientras

$$V_{max} = \max(c_r, \max(c_g, c_b)) \quad (2.17)$$

y

$$V_{min} = \min(c_r, \min(c_g, c_b)); \quad (2.18)$$

además $\text{mod}(a, b)$ representa la operación módulo, esto es, una operación que da como resultado el resto de la división entre a y b .

En el caso de las componentes que proporcionan una medida de la pureza del tono (saturación) e intensidad lumínica, en los tres casos son calculados de manera diferente. En el caso del espacio *HSI* las componentes son calculadas como:

$$c_S = 1 - \frac{3V_{min}}{c_r + c_g + c_b} \quad (2.19)$$

y

$$c_I = \frac{c_r + c_g + c_b}{3} \quad (2.20)$$

donde $\bar{c}_{HSI} = (c_H, c_S, c_I)^T \in \mathcal{C}_{HSI} \subset \mathbb{R}^3$ y V_{min} es como se muestra en (2.18).

Por otra parte, en el caso del espacio *HSL* las componentes son calculadas como:

$$c_S = \frac{V_{max} - V_{min}}{1 - |2c_L - 1|} \quad (2.21)$$

y

$$c_L = \frac{V_{max} - V_{min}}{2} \quad (2.22)$$

donde $\bar{c}_{HSL} = (c_H, c_S, c_L)^T \in \mathcal{C}_{HSL} \subset \mathbb{R}^3$, mientras V_{max} y V_{min} son como se muestra en (2.17) y (2.18), respectivamente.

Finalmente, en el caso del espacio de color *HSV* las componentes son calculadas como:

$$c_S = \frac{V_{max} - V_{min}}{V_{max}} \quad (2.23)$$

y

$$c_V = V_{max} \quad (2.24)$$

donde $\bar{c}_{HSV} = (c_H, c_S, c_V)^T \in \mathcal{C}_{HSV} \subset \mathbb{R}^3$.

Espacios *CIE*

Este sistema, presentado en 1931, fue el primero diseñado de manera que se presentara un enlace entre las longitudes de onda del espectro electromagnético visible y la percepción humana del color. El sistema *CIE* (siglas en francés de la Comisión Internacional de la Iluminación) en su forma estándar CIE_{XYZ} ($\mathcal{C}_{XYZ} \subset \mathbb{R}^3$) cuenta con tres componentes denotadas como c_X , c_Y y c_Z las cuales pueden ser obtenidas a partir de los valores en el espacio *RGB* [27, 28]. Existen variedad de transformaciones que se pueden aplicar a puntos en el espacio *RGB* para obtener puntos en el espacio CIE_{XYZ} . Algunos ejemplos son las transformaciones aplicadas en los estándares de codificación *PAL/SECAM*, *NSTC* y *sRGB* (estándar RGB). Adicionalmente, estas transformaciones presentan variaciones según el tipo de estándar de iluminación (a saber *CIE-A*, *CIE-C*, *CIE-D55* y *CIE-D65*, entre otros). Los diferentes estándares mencionados buscan simular situaciones típicas de iluminación correlacionadas a la temperatura a partir de diferentes localizaciones del punto de referencia (color blanco) y los colores primarios [27, 36]. En particular, en el caso del *sRGB* para una iluminación tipo *CIE-D65*, según se presentó en [36], la matriz de transformación es como se muestra:

$$\begin{bmatrix} c_X \\ c_Y \\ c_Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1804 \\ 0.2127 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9503 \end{bmatrix} \begin{bmatrix} c_r \\ c_g \\ c_b \end{bmatrix}, \quad (2.25)$$

donde $\bar{c}_{XYZ} = (c_X, c_Y, c_Z)^T \in \mathcal{C}_{XYZ}$ y \bar{c}_{rgb} es como se mostró en (2.8)-(2.10).

Existen dos variaciones del sistema conocidas como $CIEL^*a^*b^*$ ($\mathcal{C}_{Lab} \subset \mathbb{R}^3$) y $CIEL^*u^*v^*$ ($\mathcal{C}_{Luv} \subset \mathbb{R}^3$) cuyas componentes pueden ser obtenidas a partir de transformaciones no-lineales de las coordenadas en el espacio CIE_{XYZ} [27, 28].

Las componentes en el espacio $CIEL^*a^*b^*$ están definidos según:

$$c_L = 116 \sqrt[3]{\frac{c_Y}{c_{Y_0}}} - 16, \quad (2.26)$$

$$c_a = 500 \left(\sqrt[3]{\frac{c_X}{c_{X_0}}} - \sqrt[3]{\frac{c_Y}{c_{Y_0}}} \right), \quad (2.27)$$

$$c_b = 200 \left(\sqrt[3]{\frac{c_Y}{c_{Y_0}}} - \sqrt[3]{\frac{c_Z}{c_{Z_0}}} \right), \quad (2.28)$$

donde el punto $(c_{X_0}, c_{Y_0}, c_{Z_0})^T$ corresponde a la coordenada del blanco según el estándar

de iluminación a usar.

Por otra parte, las componentes de color en el espacio $CIEL^*u^*v^*$ están definidas según:

$$c_L = 116 \sqrt[3]{\frac{c_Y}{c_{Y_0}}} - 16, \quad (2.29)$$

$$c_u = 13c_L(c_{u'} - c_{u_0}), \quad (2.30)$$

$$c_v = 13c_L(c_{v'} - c_{v_0}), \quad (2.31)$$

donde el valor c_{Y_0} es la segunda componente de la coordenada del blanco $(c_{X_0}, c_{Y_0}, c_{Z_0})^T$ según el estándar de iluminación, mientras los valores de $c_{u'}$ y $c_{v'}$ se calculan como sigue:

$$c_{u'} = \frac{4c_X}{c_X + 15c_Y + 3c_Z}, \quad (2.32)$$

y

$$c_{v'} = \frac{9c_Y}{c_X + 15c_Y + 3c_Z}; \quad (2.33)$$

mientras los valores de c_{u_0} y c_{v_0} se obtienen evaluando (2.32) y (2.33) en el punto $(c_{X_0}, c_{Y_0}, c_{Z_0})^T$.

2.2.3 Espacio híbrido de color

En [37] fue propuesto un método para clasificar *píxeles* en un espacio híbrido de color (*HSC* por sus siglas en inglés). Allí fueron tomadas las componentes de color de 14 diferentes espacios ($c_R, c_G, c_B, c_H, c_S, c_I, c_r, c_g, c_b$, entre otros) y usando un criterio de separación, eran seleccionadas las 3 que mejor permitían discriminar entre los objetos presentes en la imagen.

El enfoque usado para determinar el espacio híbrido de color fue un esquema de aprendizaje de máquinas supervisado (*supervised machine learning*), en el que cada *píxel* de la imagen de entrada es representado como un vector K -dimensional ($K = 14$) y a partir de éstos fue maximizada una función de costo. La función de costo mencionada fue definida a partir de medidas de dispersión intra-clase (intra-objeto) y dispersión inter-clase (inter-objeto), de manera que las 3 componentes de color que aumentaban la dispersión inter-clase a la vez que disminuían la dispersión intra-clase eran seleccionadas como las componentes del espacio híbrido de color.

2.3 Segmentación de imágenes digitales

Como se mencionó en la Introducción, una de las etapas importantes en el análisis de imágenes es la segmentación de zonas de interés. A continuación son presentados algunos de los métodos usados con frecuencia en el análisis de imágenes de diversas naturalezas. Los métodos se presentan dentro de tres grandes grupos: técnicas basadas en puntos, basadas en bordes y basadas en regiones.

2.3.1 Técnicas basadas en puntos - Umbralización

Este conjunto de técnicas consiste en segmentar las imágenes directamente en regiones, basándose en los valores de intensidad o las propiedades de $v(\mathbf{k})$. Básicamente, en estos algoritmos se hace uso de un valor denominado *umbral* para convertir una imagen en escala de grises a una imagen binaria. En los casos en que es utilizado un solo umbral se asume que los valores de la imagen siguen una distribución bi-modal, es decir que los valores de intensidad se encuentran distribuidos al rededor de dos valores (*modas*) [11]. Sin embargo, la idea puede ser extendida de manera que el resultado sea una imagen con múltiples intensidades, definiendo varios *umbrales* y suponiendo que los valores de la imagen siguen distribuciones multi-modales [11].

En general, los resultados entregados por algoritmos que hacen uso de algún método de umbralización son sensibles al ruido, a iluminación no-homogénea y no son apropiados para segmentar objetos con presencia de texturas [11].

El principio básico de la *umbralización* consiste en separar la imagen definida por $v(\mathbf{k})$ en dos conjuntos de puntos, un conjunto para los puntos pertenecientes al *objeto* y otro conjunto para los que no pertenecen al objeto (*fondo*) [11].

Una manera sencilla de llevar a cabo esta separación es mediante el uso de un *umbral* τ . De esta manera si $v(\mathbf{k}) > \tau$ entonces el *spel* dado por el punto \mathbf{k} pertenece al objeto, en caso contrario pertenece al *fondo*. El resultado, o imagen segmentada se define entonces como

$$v'(\mathbf{k}) = \begin{cases} 1, & \text{si } v(\mathbf{k}) > \tau, \\ 0, & \text{cualquier otro caso.} \end{cases} \quad (2.34)$$

El proceso *umbralización* puede ser *global* (τ constante) o *dinámico*. Una forma de implementar *umbralización dinámica* consiste en modificar el valor de τ para cada *spel* en la imagen, según los *spels* que se encuentren alrededor (por ejemplo según la

intensidad promedio de este conjunto).

Puede también presentarse el caso en el que el interés sea segmentar dos objetos y el fondo. En este caso es necesario definir dos umbrales (τ_1, τ_2) . Entonces, cada *spel* en la imagen definido en el punto \mathbf{k} recibirá una etiqueta (objeto A, objeto B o fondo) asignando valores de intensidad a, b o c , según:

$$v'(\mathbf{k}) = \begin{cases} a, & \text{si } v(\mathbf{k}) > \tau_2, \\ b, & \text{si } \tau_1 < v(\mathbf{k}) \leq \tau_2 \text{ y} \\ c, & \text{si } v(\mathbf{k}) \leq \tau_1. \end{cases} \quad (2.35)$$

Para aplicar *umbralización* a una imagen existen varios métodos populares, entre ellos el método de máxima entropía y el método propuesto por *Otsu*. Uno de los más populares es el método propuesto por *Otsu* en 1979 [38]. En el trabajo mencionado, *Otsu* presentó un método para calcular un valor óptimo de umbral usando información estadística [11]. En este enfoque, el problema de segmentación se aborda como un problema de decisión estadística, en donde el objetivo es minimizar el error promedio al asignar un punto de la imagen a alguna *clase*.

Por otra parte, otros métodos más recientes sugieren el uso de umbrales no-lineales usando reglas difusas multi-dimensionales. En estos trabajos la pertenencia de un punto de la imagen a un objeto de interés dentro de la imagen se basa en reglas derivadas de la *lógica difusa* y algoritmos evolutivos [39].

2.3.2 Técnicas basadas en bordes

Los objetos de interés dentro de una imagen digital $v(\mathbf{k})$ pueden ser segmentados utilizando primero una etapa de detección de bordes. La detección de bordes es un área muy estudiada dentro del procesamiento digital de imágenes. Algunos algoritmos dentro de esta técnica están basados en el cálculo de derivadas de v a una cierta escala espacial (relacionada con el tamaño de los objetos de interés), con el objetivo de detectar diferencias significativas que sean útiles en etapas posteriores en la identificación de los objetos. Dentro de este grupo se pueden encontrar: *Canny*, *Prewit*, *Sobel* y *Laplaciano de Gaussiana* entre otros. Sin embargo, para que estos algoritmos puedan ser utilizados como base en tareas de segmentación se hace necesario, la mayoría de las veces, que la tarea de detección de bordes sea complementada por una etapa donde se apliquen técnicas que permitan construir, de la mejor manera posible, contornos cerrados de

los objetos de interés. Por otra parte, existen enfoques que hacen uso de modelos superpuestos en el espacio de la imagen los cuales pueden cambiar su forma para ajustarse a los contornos de los objetos. Un ejemplo de dichos enfoques son los llamados *Modelos de forma activa* (*ASM* por sus siglas en inglés) presentados por Cootes en [40], los cuales consisten en curvas o superficies definidas en el dominio de $v(\mathbf{k})$ que pueden ser deformadas bajo el efecto de fuerzas internas y externas. Las fuerzas internas definen propiedades de la curva o superficie (por ejemplo, suavidad de la curva o superficie), mientras que las fuerzas externas están relacionadas con los *píxeles* dentro de la imagen y son éstas las que permiten mover y deformar el modelo para que este se ajuste al objeto de interés [13]. Usando esta técnica, se puede, de manera automática, segmentar el contorno de uno o varios objetos de interés con exactitud y rapidez, sin embargo los resultados dependen críticamente de la inicialización de los modelos, ya que si esta no es adecuada, los modelos pueden no converger a una solución correcta.

2.3.3 Técnicas basadas en regiones

Estas técnicas hacen uso de las características de la imagen digital para mapear los *spels* en el conjunto de entrada a una serie de conjuntos llamados *regiones*. Dentro de estas técnicas se encuentran los métodos de agrupamiento, métodos de *separar/componer* y métodos basados en teoría de gráficas. Existen variedad de criterios a los cuales las imágenes de entrada son sometidos de manera que se puedan construir las diferentes regiones que componen la imagen, sin embargo, la selección del criterio, o los criterios de similitud dependen no solo del problema sino también de las características de las imágenes sometidas al proceso.

Uno de los ejemplo clásicos de esta familia de algoritmos lo representa el método *watershed* presentado en [41, 42], en éste se considera el complemento o “negativo” de la imagen $v(\mathbf{k})$, en el caso de imágenes 2D, como una superficie topológica construida a partir de dos coordenadas espaciales y una altura asociada a cada coordenada (relacionada con el valor de intensidad en ese punto). Usando este principio, las regiones homogéneas en $v(\mathbf{k})$ pueden ser consideradas *cuencas* rodeadas por puntos elevados, ver Figura 2.3. El método hace uso del gradiente de la imagen para hallar mínimos, máximos y puntos de inflexión.

Como se ha mencionado antes, existe otro conjunto de métodos basados en separación y composición de regiones. Estos métodos toman una aproximación opuesta a las de crecimiento de regiones. Inician con la imagen completa, posteriormente,

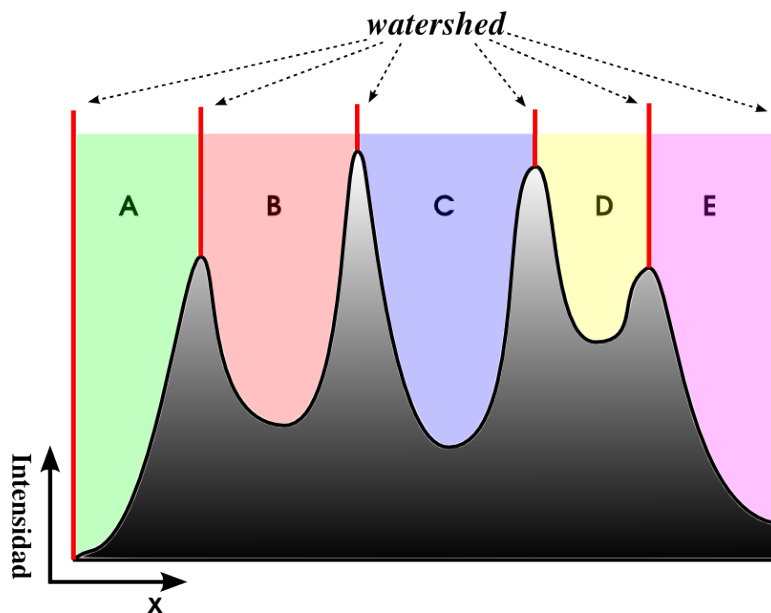


Figura 2.3: Esquema funcional en una dimensión del método *watershed*. Se puede considerar que al agua fluye y llena las cuencas dando origen a las diferentes regiones u objetos segmentados. Obsérvese que las cuencas están separadas por líneas divisorias, conocidas como *watershed lines*, las cuales previenen que se “mezclen” las regiones.

la dividen en sub-imágenes en caso de que ésta no cumpla con las restricciones de homogeneidad (por ejemplo, intensidad dentro de un determinado rango). El procedimiento de separación es aplicado a cada sub-imagen de manera recursiva hasta que todas cumplen con el criterio de homogeneidad [11]. Hecho esto, la imagen puede ser representada como un *quadtrees* u *octrees*. Posteriormente, son evaluadas las regiones adyacentes y son unidas si se da el caso que poseen propiedades similares. La desventaja principal de la separación iterativa de regiones, además de la efectividad de los criterios usados en el proceso de separación y el criterio de parada de este mismo, es también que diferentes regiones contiguas pueden tener propiedades similares. Sin embargo, una solución propuesta es incluir una etapa de composición dentro del algoritmo que lleva a cabo la separación [11].

Más recientemente, fue propuesto en [43] un método de segmentación semi-automático basado en *autómatas celulares*. Como es usual en los métodos de crecimiento de regiones, la intervención del usuario consiste en seleccionar *semillas* que pertenezcan a los objetos de interés para ser segmentados, la diferencia con los otros métodos, consiste en el uso *autómatas celulares* para llevar a cabo de manera automática la tarea

de etiquetado de todos los demás *spels* en la imagen. En este trabajo se afirma que el desempeño del método propuesto es competitivo frente a otros algoritmos basados en teoría de grafos ya que integra capacidad de segmentación multi-objeto, segmentación en varias dimensiones y velocidad suficiente para llevar a cabo segmentación interactiva; además de que afirman que la técnica propuesta ofrece mayor “extensibilidad” del algoritmo a través de diferentes reglas que pueden dirigir la evolución del *autómata*, mayor interactividad y control del usuario en el proceso de segmentación.

Finalmente, otra conjunto de técnicas para llevar a cabo segmentación de imágenes son aquellas basadas en principios de *lógica difusa*. Este tipo de técnicas han demostrado eficacia bajo diferentes condiciones de ruido, presencia de textura y artefactos [17, 14, 44, 45, 46]. Una característica de este conjunto de algoritmos, es que en ellos se supone que las propiedades que definen la pertenencia de un elemento a un objeto o clase dentro de la imagen, cambian suavemente. De esta manera, puede ser asignado a cada elemento un valor real en el intervalo $[0, 1]$ que defina su grado de *pertenencia* a una determinada clase; donde el valor de 1 indica con certeza que el punto pertenece al objeto y un valor de 0 indica que el punto no hace parte de la región que define el objeto. Este conjunto de algoritmos, forma parte de los algoritmos semi-automáticos donde es requerida la intervención del usuario. El usuario debe seleccionar puntos sobre la imagen, denominados *semillas*, dentro de la región de la imagen donde identifica que se encuentra el objeto de interés a ser segmentado. Esta información es usada para calcular que tan similares son todos los puntos de la imagen respecto al conjunto de *semillas* establecidas por el usuario. El grado de similitud entre cualquier par de puntos de la imagen es calculado mediante una función de *conectividad difusa* (*conectividad* de aquí en adelante), la cual es una función global que depende de una relación de *afinidad difusa* (*afinidad* de aquí en adelante). Esta última permite medir de manera local la *similitud* entre dos puntos adyacentes. Típicamente, una relación de *afinidad* consta de dos componentes, uno de ellos permite medir las características de los objetos, mientras el otro proporciona una medida de homogeneidad, ver sección 2.5.

2.4 Segmentación difusa

Como fue presentado en [17], el algoritmo de segmentación simultánea de múltiples objetos presentes en una imagen basado en el concepto de *conectividad difusa*, puede

ser entendido a través de una analogía con un ejercicio militar de expansión de territorio como sigue: Hay M ejércitos compitiendo por el control de N terrenos conectados entre ellos por caminos. En este ejercicio, M corresponde al número de objetos de interés a segmentar en la imagen 3D y N corresponde al número de *vóxeles* que la constituyen.

Al inicio del ejercicio, todos los ejércitos tienen su máximo de fuerza (representada por el valor 1) y ocupan sus respectivos terrenos (*semillas*).

Posteriormente todos los ejércitos tratan de incrementar su territorio, pero como resultado del desplazamiento de un terreno a otro, se ve reducida la fuerza de los soldados a un valor que equivale al mínimo entre la fuerza que tenían en el terreno anterior y la *afinidad* para ese ejército en particular respecto al camino que une los dos terrenos. Las *afinidades* de los caminos para los distintos ejércitos es fija durante el ejercicio y está relacionada con la facilidad que tienen para desplazarse de un lugar a otro.

A lo largo del ejercicio, cada terreno también tiene una *resistencia* asignada, la cual corresponde a un valor real en el rango $[0, 1]$ y está directamente relacionada con la fuerza del ejército más fuerte que ocupa el terreno. La resistencia de cada terreno puede aumentar a medida que los ejércitos avancen y en cualquier momento del ejercicio cada terreno puede estar ocupado por uno o más ejércitos si éstos tienen la misma fuerza.

El objetivo del ejercicio de expansión militar, el cual puede ser usado como un acercamiento para entender el algoritmo presentado en [17], es determinar cómo la configuración final de los terrenos ocupados depende de la asignación inicial de los terrenos a los distintos ejércitos.

Aplicado a imágenes definidas en tres dimensiones, la segmentación difusa asume que el valor de densidad (o color) de un *vóxel* del conjunto de datos permite definir si este es miembro o no de un objeto (o clase). Esta clasificación se da a partir de una medida de similitud entre los *vóxeles* adyacentes que se obtiene a partir de la función de *afinidad*. Ésta a su vez, depende de estadísticas de primer orden de los conjuntos de *semillas* ubicados en la imagen por el usuarios antes de iniciar la segmentación.

El algoritmo de segmentación difusa presentado en [17] y aplicado a una imagen 3D permite construir una función σ que para M objetos mapea cada *vóxel* $\mathbf{c} \in \mathbb{Z}^3$ en la imagen, a un vector $(M + 1)$ -dimensional $\sigma^{\mathbf{c}} = (\sigma_0^{\mathbf{c}}, \sigma_1^{\mathbf{c}}, \dots, \sigma_M^{\mathbf{c}})$ donde cada elemento de este arreglo se encuentra en el rango $[0, 1]$. El concepto básico a partir del cual se construye $\sigma^{\mathbf{c}}$, es el de *conectividad difusa* (*Fuzzy connectedness*), el cual como se ha mencionado antes, permite asignar a cada par de elementos adyacentes del conjunto de datos un valor real en el rango $[0, 1]$. En términos de cómo están definidos los

problemas de *lógica difusa* en la literatura, para este ejercicio, el conjunto difuso en cuestión es el de los “pares de elementos conectados” y el *grado de membresía* de cada par de elementos es su *conectividad difusa* [14].

Una manera de entender el problema abordado es representar la imagen a la entrada del algoritmo de segmentación como una gráfica de nodos y aristas, así como se muestra en la Figura 2.4. El algoritmo de segmentación difusa permite calcular un *mapa de conectividad* a partir de esta representación, el cual se define como la *conectividad* de cada elemento en la gráfica de nodos, es decir cada *vóxel* en la imagen, respecto al conjunto de nodos *semilla* (ver Figura 2.5). Debido al costo computacional alto que implica la tarea mencionada anteriormente, en [19] fue sugerido un algoritmo que usa una discretización del rango de números $[0, 1]$ el cual representó una aceleración de aproximadamente siete veces respecto al propuesto en [17].

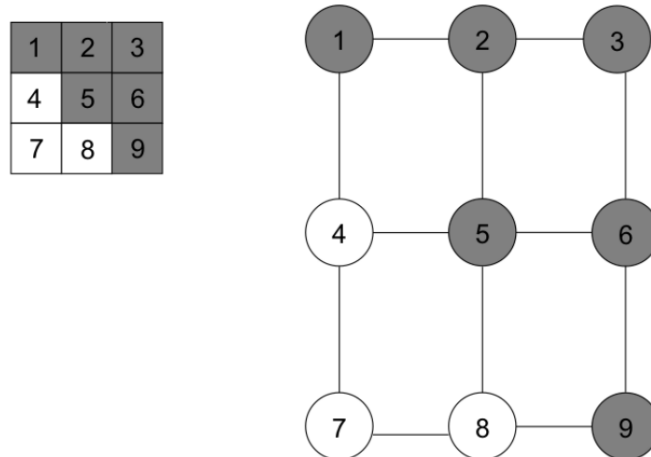


Figura 2.4: Representación de una imagen de tamaño 3×3 como una gráfica donde cada *spel* corresponde a un nodo y las aristas representan un tipo de adyacencia entre los nodos (π_1 en este caso), cortesía de [2]

La implementación del algoritmo, permite segmentar M objetos con la posibilidad de que cada objeto (o clase) tenga su propia definición para la función de *afinidad* ψ y su propio conjunto de *semillas*. El resultado final define a un objeto (conjunto conexo), como el conjunto de *vóxeles* que están conectados de una manera más fuerte al conjunto de *semillas* de ese objeto que a cualquier conjunto de *semillas* de otro objeto.

Hasta este punto han sido presentados elementos de la teoría de una manera intuitiva, sin embargo, a partir de este punto serán presentados de una manera formal los elementos de la teoría de donde se desprendió la implementación del algoritmo de segmentación basado en principios de *lógica difusa* usado en el presente trabajo. Aunque

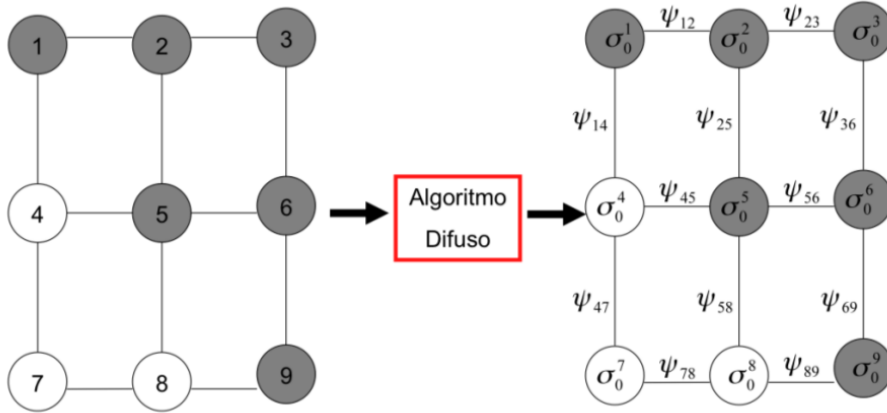


Figura 2.5: El algoritmo de segmentación difusa realiza un mapeo de la gráfica que representa la imagen a una gráfica difusa donde a cada nodo corresponde el valor σ_0^c y cada arista tiene un peso $\psi(\mathbf{c}, \mathbf{d})$, cortesía de [2]

será presentado el teorema que rige el proceso, se invita al lector revisar las referencias [19, 14] para mayor profundidad en los conceptos presentados.

Se define una M -semisegmentación del conjunto V (imagen 3D) como la función σ que para M objetos mapea cada *vóxel* en el conjunto de datos $\mathbf{c} \in \mathbb{Z}^3$, a un vector $(M + 1)$ -dimensional $\sigma^c = (\sigma_0^c, \sigma_1^c, \sigma_2^c, \dots, \sigma_M^c)$ de tal forma que:

1. $\sigma_0^c \in [0, 1]$;
2. para cada m en $1 \leq m \leq M$, el valor de σ_m^c puede ser 0 ó σ_0^c ; y
3. $\sigma_m^c = \sigma_0^c$ para al menos un m en $1 \leq m \leq M$.

Del anterior conjunto de características del vector σ^c , se puede ver que el valor de $\sigma_0^c = \max(\sigma_m^c)$, para $1 \leq m \leq M$. Por otra parte, en términos de los problemas planteados en lógica difusa, el valor de σ_0^c representa el *grado de pertenencia* del *vóxel* \mathbf{c} al m -ésimo objeto. Respecto al ejercicio de expansión territorial de los ejércitos presentado antes, σ_0^c es proporcional a la *resistencia* del terreno \mathbf{c} y el m -ésimo ejército puede ocupar éste si, y sólo si, $\sigma_m^c = \sigma_0^c > 0$.

Esta definición de M -semisegmentación permite al *vóxel* pertenecer a más de un objeto siempre y cuando tenga el mismo *grado de pertenencia* respecto a todos ellos.

Se dice que una M -semisegmentación es una M -segmentación si se cumple que σ_0^c es positivo para todo $\mathbf{c} \in V$.

Las M -semisegmentaciones según los elementos presentados hasta ahora son determinados por *mapas de conectividad* (ver Figura 2.5). En esta representación se

puede definir una *cadena* como una secuencia $\langle \mathbf{c}^0, \mathbf{c}^1, \dots, \mathbf{c}^P \rangle$ de $(P+1)$ nodos, donde sus *enlaces* están representados por nodos consecutivos $(\mathbf{c}^{p-1}, \mathbf{c}^p)$. Se define $\psi : V^2 \rightarrow [0, 1]$ como la función de *afinidad difusa* la cual permite asignar la *fuerza* de cada *enlace*, es decir, asigna un valor real entre 0 y 1 a cada par de elementos en V . La *fuerza* de la *cadena* se define como la *fuerza* del *enlace* más débil presente en ésta. Se dice que un conjunto $U \subseteq V$ está *conectado* bajo la función ψ , si para cada par de *vóxeles* en U , hay una *cadena* de *fuerza* positiva que vaya desde el primer elemento de cada par hasta el segundo.

La *conectividad* $(\mu : V^2 \rightarrow [0, 1])$ entre dos *vóxeles* \mathbf{c} y \mathbf{d} se define como:

$$\mu(\mathbf{c}, \mathbf{d}) = \max_{\substack{\langle \mathbf{c}^0, \mathbf{c}^1, \dots, \mathbf{c}^P \rangle \\ \mathbf{c} = \mathbf{c}^0, \mathbf{d} = \mathbf{c}^P}} \left[\min_{1 \leq p \leq P} \psi(\mathbf{c}^{p-1}, \mathbf{c}^p) \right]. \quad (2.36)$$

Ahora bien, si el objetivo es segmentar múltiples objetos, resulta razonable que cada uno de ellos pueda tener su propia definición de *afinidad* y su propio conjunto de *semillas*. De esta manera, el *mapa de conectividad con semillas* es una tripleta (V, Ψ, \mathcal{V}) , donde V es la imagen de entrada, $\Psi = \langle \psi_1, \psi_2, \dots, \psi_M \rangle$ son las definiciones de afinidad para los M objetos; y $\mathcal{V} = \langle V_1, V_2, \dots, V_M \rangle$ contiene los conjuntos de *semillas* (uno para cada objeto, y $V_m \subseteq V$ para $1 \leq m \leq M$). Esta representación es *conectable* si se cumple que:

1. El conjunto de entrada V está conectado bajo la función $\phi_\Psi(\mathbf{c}, \mathbf{d}) = \min(\psi_m(\mathbf{c}, \mathbf{d}))$ para todo $\mathbf{c}, \mathbf{d} \in V$ y $1 \leq m \leq M$,
2. $V_m \neq \emptyset$, para al menos un conjunto semilla en $1 \leq m \leq M$.

Antes de presentar el teorema, es definida la cadena $\langle \mathbf{c}^0, \mathbf{c}^1, \dots, \mathbf{c}^P \rangle$ como una σ_m -cadena si para $0 \leq p \leq P$ se cumple $\sigma_m^{\mathbf{c}^p} > 0$. Adicionalmente, para $W \subseteq V$, es usado el término $\mu_{\sigma, m, W}(\mathbf{c})$ para denotar la máxima *fuerza* de una σ_m -cadena desde un *vóxel* en W hasta \mathbf{c} .

Teorema 1. *Si (V, Ψ, \mathcal{V}) es un mapa de conectividad con semillas (donde $\Psi = \langle \psi_1, \psi_2, \dots, \psi_M \rangle$ y $\mathcal{V} = \langle V_1, V_2, \dots, V_M \rangle$), entonces*

- (i) *Existe una M -semisegmentación σ de V con las siguientes propiedades: para cada $\mathbf{c} \in V$, si para $1 \leq n \leq M$*

$$s_n^{\mathbf{c}} = \begin{cases} 1, & \text{si } \mathbf{c} \in V_n, \\ \max_{\mathbf{d} \in V} (\min(\mu_{\sigma, n, V_n}(\mathbf{d}), \psi_n(\mathbf{d}, \mathbf{c}))), & \text{c.o.c.} \end{cases} \quad (2.37)$$

Luego para $1 \leq m \leq M$

$$\sigma_m^c = \begin{cases} s_m^c, & \text{si } s_m^c \geq s_n^c, \text{ para } 1 \leq n \leq M, \\ 0, & \text{c.o.c.} \end{cases} \quad (2.38)$$

(ii) Esta M -semisegmentación es única, y

(iii) además es una M -segmentación, dado que (V, Ψ, \mathcal{V}) es conectable.

2.5 Función de afinidad

Una mirada a (2.36) revela la importancia de la función de *afinidad* ψ ya que de la definición de ésta pueden depender de manera crítica los resultados entregados por el algoritmo de segmentación.

La idea del uso de la función de *afinidad* es que el usuario no tenga la tarea de definir de forma matemática las características del objeto a segmentar, sino que solo seleccionando un conjunto de puntos que pertenezcan a éste, el programa esté en capacidad de calcular algunas estadísticas que permitan llevar a cabo la caracterización del objeto basado en el vecindario de cada elemento *semilla* seleccionado.

En la literatura se menciona que para definir la función ψ de *afinidad* solo es necesario que ésta sea monotónica y normal [47], además de que sean consideradas en su definición las características y homogeneidad de los objetos (aunque también afirman que además debe ser simétrica [48]).

Según [16, 46] la función de *afinidad* tiene como forma general

$$\psi(\mathbf{c}, \mathbf{d}) = \psi_A(\mathbf{c}, \mathbf{d}) \left[w_H \psi_H(\mathbf{c}, \mathbf{d}) + w_G \psi_G(\mathbf{c}, \mathbf{d}) \right], \quad (2.39)$$

donde ψ_A es una relación de adyacencia entre los *vóxeles* \mathbf{c} y \mathbf{d} la cual tiene la forma

$$\psi_A(\mathbf{c}, \mathbf{d}) = \begin{cases} 1, & \text{si } (\mathbf{c}, \mathbf{d}) \in \pi \quad (\mathbf{c} \text{ y } \mathbf{d} \text{ son adyacentes)}, \\ 0, & \text{en cualquier otro caso.} \end{cases} \quad (2.40)$$

El término ψ_G en (2.39) permite medir el grado de similitud de las características de intensidad de los objetos a segmentar, mientras el término ψ_H permite medir el grado de homogeneidad de los *vóxeles* \mathbf{c} y \mathbf{d} . Las funciones ψ_G y ψ_H han sido definidas en

[16, 19, 45] como

$$\psi_G(\mathbf{c}, \mathbf{d}) = e^{-\frac{(v(\mathbf{c}) + v(\mathbf{d}) - \eta_G)^2}{2\zeta_G^2}} \quad (2.41)$$

y

$$\psi_H(\mathbf{c}, \mathbf{d}) = e^{-\frac{(|v(\mathbf{c}) - v(\mathbf{d})| - \eta_H)^2}{2\zeta_H^2}}, \quad (2.42)$$

donde η_G y ζ_G son la media y la desviación estándar, respectivamente, de la suma $v(\mathbf{u}) + v(\mathbf{v})$ de cada par de *vóxeles* adyacentes pertenecientes a los vecindarios $N(\mathbf{u})$ construidos para el conjunto de semillas V_m , esto es, η_G y ζ_G son diferentes para cada conjunto V_m de semillas, con $1 \leq m \leq M$ (M representa el número de objetos a ser segmentados). Lo anterior aplica de manera similar para los valores η_H y ζ_H , la diferencia ahora es que éstos representan la media y la desviación estándar de la diferencia absoluta $|v(\mathbf{u}) - v(\mathbf{v})|$.

Se puede observar en (2.39) que las funciones ψ_G y ψ_H están ponderadas por las constantes w_G y w_H , cuyo valor asignado normalmente es 0.5, aunque han sido investigados los efectos de estas constantes y las formas de ajustarlas durante el proceso de segmentación [49].

2.5.1 Incorporación de información de color

Dado que las imágenes a usar en el presente trabajo están representadas en un espacio de color, las funciones de *afinidad* deben estar acorde con esta característica. En [20, 50] fueron construidas funciones de *afinidad* de forma similar a las usadas para segmentar imágenes en escala de grises, las cuales permiten incorporar información de color. Dichas funciones de *afinidad* tienen la siguiente forma, para $1 \leq i \leq 3$ (canales de color) y $1 \leq m \leq M$ (M objetos)

$$\psi_m(\mathbf{c}, \mathbf{d})_{color} = \frac{\sum_{i=1}^3 (\rho_{\eta_{G_{i,m}}, \zeta_{G_{i,m}}}(G_i) + \rho_{\eta_{H_{i,m}}, \zeta_{H_{i,m}}}(H_i))}{3}, \quad (2.43)$$

con

$$\rho_{\lambda, \epsilon}(x) = e^{-\frac{(x - \lambda)^2}{2\epsilon^2}}, \quad (2.44)$$

donde $\eta_{G_{i,m}}$ y $\zeta_{G_{i,m}}$ representan la media y la desviación estándar, respectivamente, del promedio de los valores del i -ésimo canal de color ($1 \leq i \leq 3$) para todos los pares de

vóxeles adyacentes en el conjunto de semillas V_m ; $\eta_{H_{i,m}}$ y $\zeta_{H_{i,m}}$ representan la media y la desviación estándar, respectivamente, de la diferencia absoluta de los valores del i -ésimo canal de color ($1 \leq i \leq 3$) para todos los pares de *vóxeles* adyacentes en el conjunto de semillas V_m ; mientras G_i y H_i son respectivamente el promedio y la diferencia absoluta de los *vóxeles* \mathbf{c} y \mathbf{d} en el canal de color i .

2.6 Algoritmo de segmentación multi-objeto

De manera que el presente documento presente, en la mayor medida posible, la característica de ser auto-contenido, se presenta a continuación el algoritmo *fast-MOFS* (*MOFS* proviene de las siglas en inglés de segmentación multi-objeto basada en lógica difusa) propuesto en [19] el cual ha sido elegido para ser utilizado en el presente trabajo. Las entradas del algoritmo son: la imagen V , el conjunto de funciones de *afinidad* $\Psi = \{\psi_m\}$ y el conjunto de semillas $\mathcal{V} = \{V_m\}$ para $1 \leq m \leq M$, donde M es el número de objetos a segmentar. La función $\sigma_m^{\mathbf{c}}$ representa el grado de pertenencia del *vóxel* \mathbf{c} respecto a la clase m (el cual es proporcional a la fuerza de conexión del *vóxel* \mathbf{c} a la clase m). La clase m puede “reclamar” el *vóxel* \mathbf{c} si, y sólo si, $\sigma_m^{\mathbf{c}} = \sigma_0^{\mathbf{c}} > 0$.

Para hacer más rápido el algoritmo, se hace la suposición de que el conjunto de *afinidades* $\{\psi_m\}$ es un subconjunto de A , donde K es la cardinalidad de este conjunto y $1 = a_1 > a_2 > \dots > a_K > 0$ son sus elementos. Lo anterior se basa en que en muchas aplicaciones la calidad de la segmentación no se ve significativamente afectada cuando los valores de *afinidad* se redondean a tres cifras decimales. De esta manera el conjunto $A = \{0.001, 0.002, \dots, 0.999, 1.000\}$ cuando $K = 1000$ y los elementos $a_k = 1.001 - k/K$. Adicionalmente, se dispone de un arreglo U de tamaño $M \times K$ donde cada elemento $U[m][k]$ contiene el conjunto de *vóxeles* que están ocupados por la clase del objeto m y cuya fuerza actual es $K + 1 - k$.

El algoritmo es como se presenta a continuación:

Algoritmo *fast-MOFS*

1. **para** $\mathbf{c} \in V$ **hacer**
2. **para** $m \leftarrow 0$ **a** M **hacer**
3. $\sigma_m^{\mathbf{c}} \leftarrow 0$
4. **para** $m \leftarrow 1$ **a** M **hacer**
5. **para** $k \leftarrow 1$ **a** K **hacer**
6. $U[m][k] \leftarrow 0$

```

7.     para  $\mathbf{c} \in V_m$  hacer
8.          $\sigma_0^{\mathbf{c}} \leftarrow \sigma_m^{\mathbf{c}} \leftarrow 1$ 
9.      $U[m][1] \leftarrow V_m$ 
10.    para  $k \leftarrow 1$  a  $K$  hacer
11.        para  $m \leftarrow 1$  a  $M$  hacer
12.            mientras  $U[m][k] \neq \emptyset$  hacer
13.                remove un vóxel  $\mathbf{d}$  de  $U[m][k]$ 
14.                 $C \leftarrow \{\mathbf{c} \in V \mid \sigma_m^{\mathbf{c}} < \min(a_k, \psi_m(\mathbf{d}, \mathbf{c}))$ 
                    and  $\sigma_0^{\mathbf{c}} < \min(a_k, \psi_m(\mathbf{d}, \mathbf{c}))\}$ 
15.            mientras  $C \neq \emptyset$  hacer
16.                remove un vóxel  $\mathbf{c}$  de  $C$ 
17.                 $t \leftarrow \min(a_k, \psi_m(\mathbf{d}, \mathbf{c}))$ 
18.                si  $\sigma_0^{\mathbf{c}} < t$  entonces hacer
19.                    remove  $\mathbf{c}$  de cada conjunto en  $U$  que lo contiene
20.                    para  $n \leftarrow 1$  a  $M$  hacer
21.                         $\sigma_n^{\mathbf{c}} \leftarrow 0$ 
22.                         $\sigma_0^{\mathbf{c}} \leftarrow \sigma_m^{\mathbf{c}} \leftarrow t$ 
23.                    insertar  $\mathbf{c}$  en el conjunto  $U[m][l]$ ,
                        donde  $l$  es el entero tal que  $a_l = t$ 

```

El proceso inicializa en las líneas 1-9, donde se asigna a cada $\sigma_m^{\mathbf{c}}$ el valor de 0 para cada *vóxel* \mathbf{c} y para $0 \leq m \leq M$, además de asegurarse que los conjuntos $U[m][k]$ estén vacíos. Luego, cada *vóxel* $\mathbf{c} \in V_m$ es incluido en $U[m][1]$ y los valores $\sigma_0^{\mathbf{c}} = \sigma_m^{\mathbf{c}} = 1$ (para ese *vóxel* \mathbf{c}), siendo esta última asignación equivalente a proporcionar una *fuerza* de 1 a los ejércitos y sus bases iniciales en el ejercicio de expansión de territorio (Sección 2.4).

Una vez realizado el proceso de inicialización se ejecuta el ciclo principal (líneas 11-23), donde para cada iteración k hasta K y para cada objeto m hasta el número total M , son actualizados los valores de $\sigma_m^{\mathbf{c}}$. Un valor actual es reemplazado por uno mayor si se da que existe una cadena del objeto m que parte de una *semilla* en el conjunto V_m hasta el *vóxel* \mathbf{c} con una *fuerza* mayor a la anterior; y es reemplazado por el valor de 0 si se da que, para todo $n \neq m$, hay una cadena desde una *semilla* en V_n hasta \mathbf{c} con una *fuerza* mayor al valor anterior de $\sigma_m^{\mathbf{c}}$.

Al inicio de cada iteración k , para cada índice l en el rango $k \leq l \leq K$, el conjunto $U[m][l]$ contiene todos aquellos *vóxeles* que están ocupados por la clase m y cuya fuerza actual es $K+1-l$. Durante la iteración k -ésima, exploradores de la clase m son enviados

desde cada *vóxel* en $U[m][k]$ a todos los otros *vóxeles* adyacentes (líneas 11-14) para tratar de ocuparlos. Cuando un explorador de la clase m tiene éxito en ocupar un *vóxel* \mathbf{c} , y la *fuerza* resultante de \mathbf{c} es $K + 1 - l$, entonces \mathbf{c} es insertado en el conjunto $U[m][l]$ (línea 23). Sin embargo, si \mathbf{c} es tomado porque la fuerza anterior era menor a $K + 1 - l$, entonces es necesario primero que \mathbf{c} sea removido de cualquier otro conjunto que lo contenga (línea 19).

Finalmente, pueden obtenerse como resultado las segmentaciones $\sigma_1, \sigma_2, \dots, \sigma_M$ recorriendo los elementos en los conjuntos $U[1], U[2], \dots, U[M]$.

Capítulo 3

Trabajo realizado, experimentos y resultados

3.1 Conjunto de imágenes disponibles

Para el desarrollo del presente trabajo en el que se pretende explorar la utilidad del algoritmo de segmentación basado en lógica difusa aplicado a imágenes a color, los datos de prueba constan de conjuntos de imágenes bidimensionales que representan secciones transversales de ratón. La imagen de cada corte axial tiene una dimensión de 1002×1004 píxeles y la zona abdominal, donde se encuentran los órganos de interés a ser segmentados, está compuesta por aproximadamente 40 cortes, de manera que se cuenta con imágenes 3D a color con una dimensión de $1002 \times 1004 \times 40$ vóxeles. El volumen representado por cada vóxel es aproximadamente de $0.05 \times 0.05 \times 0.90$ mm³. Las imágenes tomadas en el Instituto de Imagenología Biológica y Médica (*IBMI* por sus siglas en inglés) ubicado en Múnich-Alemania están almacenadas en un formato TIFF sin compresión, usando una representación *RGB*. Para este conjunto de imágenes, cada dato de 32 bits está dividido en bloques de 8 bits en donde se almacenan las componentes de color rojo, verde y azul; además de la componente de opacidad la cual en todos los casos tiene un valor de 255. Probablemente, debido a que no eran digitalizados todos los cortes correspondientes a un mismo espécimen en una sola sesión, las imágenes presentan diferentes niveles de iluminación en varios cortes contiguos, además de que no en todos los casos se presenta una buena alineación o registro entre éstos, ver Figura 3.1.

El objetivo de la aplicación del método de segmentación a este conjunto de imágenes, es evaluar el desempeño de éste a la hora de segmentar de manera simultánea los órganos

del interior del abdomen del ratón: riñones, páncreas, bazo, interior del estómago e hígado.

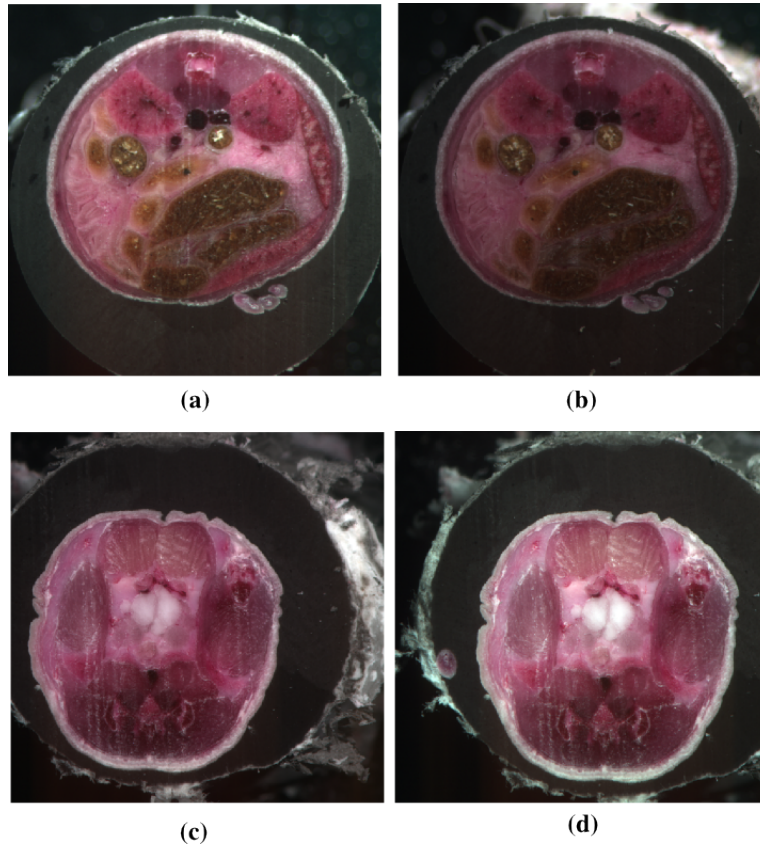


Figura 3.1: Ejemplo de características no deseadas en el conjunto de datos. Figuras (a) y (b) cortes contiguos con diferente iluminación. Figuras (c) y (d) cortes contiguos con falta de alineación.

Para el desarrollo de los experimentos fue recomendado el uso de una imagen 3D del conjunto disponible, el cual presentaba el menor número de problemas de alineación y cambios en la iluminación entre cortes contiguos.

3.2 Conjunto de datos *Ground-Truth*

De manera que los resultados de las segmentaciones entregadas por el *software* de segmentación pudieran ser sometidas a una evaluación objetiva, fueron construidas imágenes 3D, una por cada órgano de interés a ser segmentado, para ser usadas como referencia. Para esto, inicialmente se descargó un atlas de órganos de ratón disponible de manera pública en la red, el cual hace parte del proyecto *DigiMouse* ejecutado por

el grupo *Biomedical Image Group* de Universidad del sur de California (USC) [51, 52]. Luego, cuatro integrantes del grupo de investigación *VISP* (Visualización, imágenes, señales y patrones - UNAM), haciendo uso del atlas, anotaron manualmente, corte por corte, cada órgano de interés a ser segmentado posteriormente con el *software* de segmentación. Las anotaciones hechas por cada participante fueron usadas para construir imágenes binarias 3D de los órganos de interés, es decir, un volumen por cada órgano de interés. Posteriormente, las imágenes a ser usadas como *ground-truth* fueron calculadas como el promedio, para cada órgano de interés, de las imágenes binarias correspondientes (ver ejemplo de construcción en la Figura 3.2).

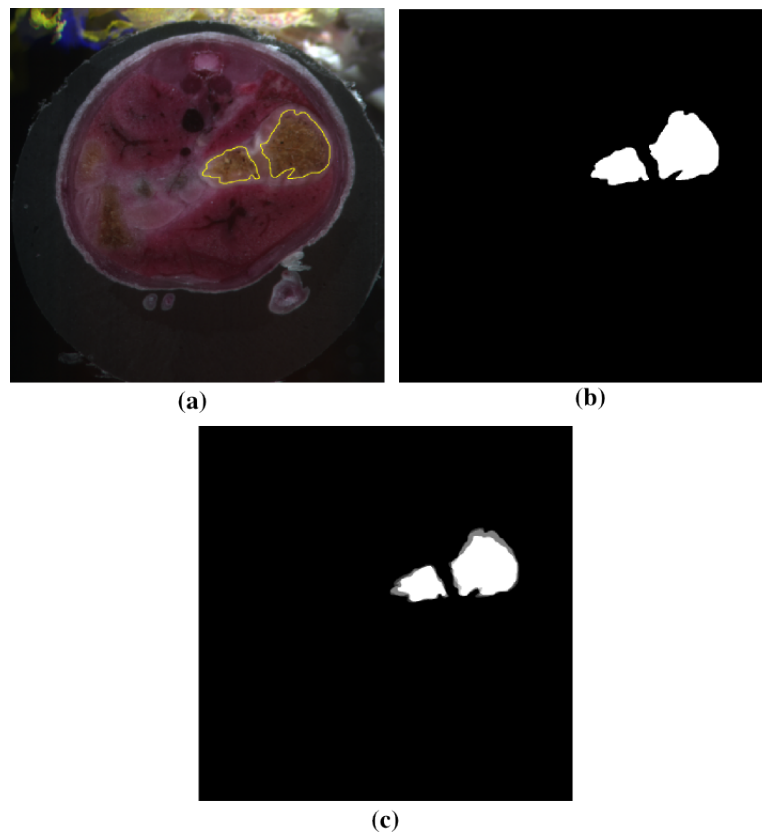


Figura 3.2: Ejemplo de construcción de volumen de datos a ser usado como *ground-truth*. (a) Anotación manual del interior del estómago. (b) Máscara binaria a partir de la anotación manual mostrada en (a). (c) Promedio de las cuatro máscaras binarias.

3.3 Herramientas de software

Se presenta a continuación una descripción breve de las herramientas de software construidas o modificadas para llevar a cabo la prueba de la hipótesis del presente trabajo. Dado que se cuenta con un conjunto de imágenes a color de alta resolución de secciones transversales de ratón, la parte de la construcción de las herramientas de software es una parte muy importante del trabajo.

Las implementaciones de código necesarias fueron elaboradas en lenguaje C/C++ y usaron como base implementaciones de algoritmos realizados por integrantes del grupo *VISP* y otros colaboradores, especialmente el grupo de investigación del Dr. Carvalho [17, 50]. Las herramientas fueron programadas en forma modular y consta de cuatro partes que permiten al usuario: definir el número de objetos a segmentar así como el conjunto de *semillas* para cada uno, segmentar los objetos de interés en un determinado espacio de color (haciendo uso del algoritmo *MOFS*), visualizar y evaluar los resultados.

3.3.1 Módulo para selección de semillas

Basado en trabajos previos llevados a cabo por integrantes y colaboradores del grupo *VISP*, fue construido un programa haciendo uso de la biblioteca *Qt* en la versión 4.8 [53], la cual permite desarrollar programas que utilizan interfaz gráfica de usuario. El programa construido permite al usuario mediante una interfaz intuitiva ubicar para cada objeto o clase a segmentar, su respectivo conjunto de semillas en la imagen a color original (2D-3D).

El programa consta de 3 ventanas: una principal, una de control de semillas y clases y una que permite desplegar en pantalla cada corte que compone la imagen 3D.

La ventana principal contiene menús que permiten cargar la imagen de interés, guardar en un archivo las semillas seleccionadas por el usuario, así como la clase a la que pertenecen, sus respectivas adyacencias y afinidades; además de que permite cargar un archivo de semillas creado en una sesión anterior.

La ventana de control (ver Figura 3.3) permite:

- Cambiar la imagen desplegada mediante un control que permite seleccionar el número del *plano* de interés.
- Según la posición del *puntero* sobre la imagen desplegada, conocer las coordenadas del punto, así como sus componentes de color (RGB).

- Adicionar y borrar clases; así como editar la adyacencia, la afinidad y el color que se usa para ilustrar su conjunto de semillas.
- Visualizar un identificador numérico de la clase activa y el número de semillas que tiene ésta.
- Borrar semillas que pertenecen a una determinada clase.
- Recomendar, una vez que el usuario esté satisfecho con el número de clases y semillas, dos espacios de color del conjunto incluido en el programa, además de un espacio híbrido de color, los cuales se espera que permitan una mejor segmentación de los objetos de interés.

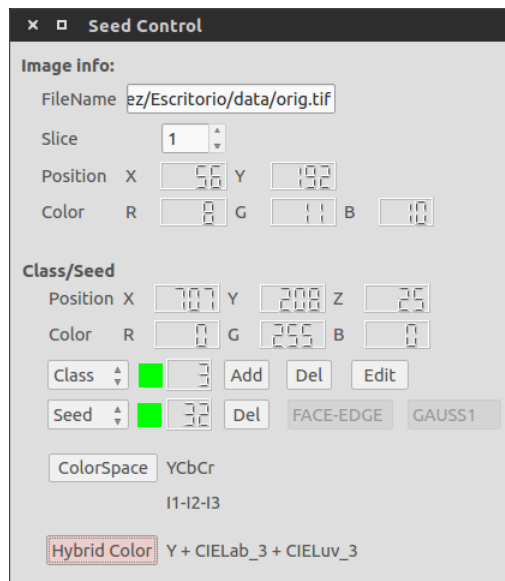


Figura 3.3: Ventana de control de semillas del programa para selección de semillas.

Por otra parte, la ventana de despliegue de la imagen permite:

- Ubicar marcas circulares de color sobre la imagen desplegada en las coordenadas donde el usuario esté interesado en ubicar semillas que pertenezcan a una determinada clase.
- Seleccionar, sobre la imagen, semillas ubicadas con anterioridad en caso de que el usuario esté interesado en descartarla del conjunto, obtener información sobre la posición de ésta o sus componentes de color.

- Mediante el uso del teclado: cambiar de *plano*, agregar clases, borrar semillas seleccionadas por el usuario y seleccionar la clase activa (para adicionar semillas a ésta u obtener información sobre la cantidad de semillas que contiene).

La salida de este programa es un archivo de texto, el cual contiene toda la información sobre los conjuntos de semillas que necesita el programa que lleva a cabo el proceso de segmentación: número de objeto o clases a segmentar, tipo de adyacencia y afinidad para cada objeto así como las coordenadas de su conjunto de semillas.

Un ejemplo de ubicación de semillas en una imagen se muestra en la Figura 3.4. Este conjunto busca separar órganos ubicados en la zona abdominal del ratón y se usa un color para cada zona de interés: riñones, páncreas, bazo, interior del estómago e hígado con semillas de color rojo, cian, verde, magenta y azul, respectivamente. Es usada una clase adicional con semillas de color amarillo para describir la región exterior del ratón.

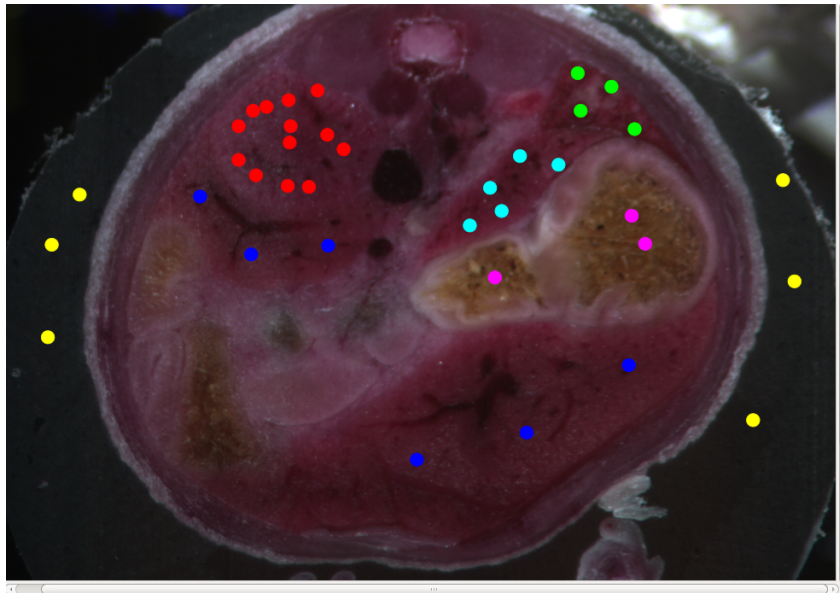


Figura 3.4: Imagen de corte axial de abdomen de ratón con 6 conjuntos de semillas ubicadas en el exterior del espécimen y los órganos de interés: riñón, páncreas, bazo, interior del estómago e hígado. Los colores de cada grupo de semillas son usados para diferenciar a qué objeto de interés a segmentar pertenecen. El tamaño de los círculos que representan las semillas ha sido incrementado para propósitos visuales.

Métrica usada para recomendar diferentes espacios de color

La recomendación de los espacios de color, estándar o híbrido, que puede proporcionar el *software* de selección de semillas para llevar a cabo la segmentación y los cuales se espera que entreguen buenos resultados, se basan en la distancia *Euclídeana* entre hiper-esferas *Gaussianas*, una por cada objeto a segmentar, definidas a partir de las medias y las desviaciones estándares de los valores de las *semillas* en un determinado espacio de color [50].

3.3.2 Módulo para realizar segmentación

En el grupo de investigación *VISP* se cuenta con una versión del programa de segmentación el cual trabaja sobre imágenes en niveles gris en varios formatos incluyendo el formato TIFF. El módulo de segmentación del presente trabajo se construyó a partir de la versión descrita anteriormente y el trabajo consistió en agregar a este programa la capacidad de leer imágenes en formato TIFF a color (*RGB*) y llevar a cabo el proceso de segmentación en los siguientes 10 diferentes espacios de color: *RGB*, $i_1i_2i_3$, *HSI*, *HSL*, *HSV*, *YCbCr*, *YUV*, *CIELab* (estándar 1931, iluminación D65), *CIELuv* (estándar 1931, iluminación D65) y *escala de grises*; así como en un espacio híbrido que puede ser definido por el usuario a partir de las componentes de los espacios estándar de color ya mencionados. Como se mencionó en la Sección 3.3.1, el usuario puede usar alguno de los espacios de color, ya sea alguno de los espacios estándar o el híbrido, recomendados por la herramienta de selección de semillas.

La estructura del programa permite agregar nuevas clases que definan las funciones de afinidad. En el presente trabajo, cuando la imagen de entrada está representada en el espacio *RGB*, el programa la transforma al espacio de color definido por el usuario para llevar a cabo la segmentación. Posteriormente, es usada la clase que se agregó a la implementación anterior del programa la cual permite definir la función de afinidad para imágenes con 3 componentes de color y calcular así la conectividad de todos los vóxeles de la imagen con respecto a los conjuntos de semillas de los objetos de interés a segmentar.

De manera que se pudiera conservar la estructura de almacenamiento de los datos, provenientes de las imágenes de entrada, incluida en la implementación original, se usó un tipo de dato *entero de 32 bits sin signo* para almacenar las 3 componentes de color del espacio donde se esté llevando a cabo el proceso de segmentación así como la componente de opacidad. Dado que en los cálculos intermedios es necesario extraer

las componentes de color del espacio en el que se lleva a cabo la segmentación, son aplicadas máscaras de bits para extraerlas, realizar las operaciones y después almacenar los resultados de nuevo en los datos de 32 bits sin signo.

El programa recibe como parámetro de entrada la ruta y el nombre de la imagen a segmentar, la ruta y el nombre base de las imágenes a ser entregadas por el programa como resultado de la segmentación (una por cada objeto definido por el usuario), la ruta y el nombre del archivo que contiene la información de las semillas así como el espacio de color a ser usado en el proceso de segmentación. Dado el caso que sea seleccionado el espacio híbrido de color para llevar a cabo la segmentación, el programa pide al usuario seleccionar 3 componentes de color de la lista usando identificadores numéricos, ver Tabla 3.1.

0 \rightarrow $(\bar{c}_{RGB})_1$	13 \rightarrow $(\bar{c}_{YCbCr})_1$
1 \rightarrow $(\bar{c}_{RGB})_2$	14 \rightarrow $(\bar{c}_{YCbCr})_2$
2 \rightarrow $(\bar{c}_{RGB})_3$	15 \rightarrow $(\bar{c}_{YCbCr})_3$
3 \rightarrow $(\bar{c}_{i_1i_2i_3})_1$	16 \rightarrow $(\bar{c}_{YUV})_1$
4 \rightarrow $(\bar{c}_{i_1i_2i_3})_2$	17 \rightarrow $(\bar{c}_{YUV})_2$
5 \rightarrow $(\bar{c}_{i_1i_2i_3})_3$	18 \rightarrow $(\bar{c}_{YUV})_3$
6 \rightarrow $(\bar{c}_{HSI})_1$	19 \rightarrow $(\bar{c}_{Lab})_1$
7 \rightarrow $(\bar{c}_{HSI})_2$	20 \rightarrow $(\bar{c}_{Lab})_2$
8 \rightarrow $(\bar{c}_{HSI})_3$	21 \rightarrow $(\bar{c}_{Lab})_3$
9 \rightarrow $(\bar{c}_{HSL})_2$	22 \rightarrow $(\bar{c}_{Luv})_1$
10 \rightarrow $(\bar{c}_{HSL})_3$	23 \rightarrow $(\bar{c}_{Luv})_2$
11 \rightarrow $(\bar{c}_{HSV})_2$	24 \rightarrow $(\bar{c}_{Luv})_3$
12 \rightarrow $(\bar{c}_{HSV})_3$	

Tabla 3.1: Lista de identificadores numéricos de componentes de color para formar espacio híbrido

Debido a que el proceso de segmentación produce una gráfica difusa que representa varias clases (ver Sección 2.4) el programa crea una imagen por cada clase segmentada. En cada imagen el valor de los *vóxeles* (valor real en el rango $[0, 1]$) representa la *conectividad* de ese punto respecto a las semillas que definen esa clase. Los puntos que tienen un valor de cero en la imagen que representa una clase es porque han sido asignados a otras clases.

3.3.3 Módulo para visualización de objetos segmentados

Para visualizar los objetos segmentados fue construido un programa con interfaz de usuario el cual permite cargar y desplegar el resultado de la segmentación, en el caso de imágenes 3D la herramienta permite desplegar un corte a la vez; un ejemplo se muestra en en la Figura 3.5. Es importante mencionar que el nivel de pertenencia a una determinada clase está dada por la intensidad del punto, es decir, un *píxel* de color azul oscuro, por ejemplo $\bar{c}_{RGB} = (0, 0, 120)$, indica una menor pertenencia al objeto identificado con el color azul que un *píxel* cuyo color se acerque más al tono puro, es decir $\bar{c}_{RGB} = (0, 0, 255)$. Adicionalmente, de manera que el usuario pueda visualizar de mejor manera la relación espacial de los objetos segmentados, fue agregado a este módulo la capacidad de aplicar a la imagen de interés la técnica de *Renderizado directo de volumen* (*Volume Rendering*) usando el resultado de la segmentación como *función de transferencia* tri-dimensional, ver Apéndice A.

Con respecto a la técnica de *Volume Rendering*, en el presente documento se hace uso de términos derivados de dicha técnica como “renderizar”, “renderizado” o “renderización”; los cuales si bien no existen en el idioma español si son ampliamente usados en el área de graficación y visualización por computador, debido a que no existe un término en este idioma que por si solo englobe todo lo que implica la “renderización”. El término en cuestión implica el proceso de generar una imagen a partir de un modelo 3D en que generalmente es tomada en cuenta información de materiales, colores e iluminación.

3.3.4 Módulo para evaluación de resultados

Para evaluar los resultados fue construido un módulo que permite calcular, para cada objeto segmentado, según un *umbral de confianza* establecido por el usuario, los siguientes coeficientes: *Sorensen-Dice*, *sensibilidad* y *especificidad*.

Los datos de entrada del programa son la imagen segmentada para un determinado objeto a ser evaluado y una imagen referencia. La información sobre la construcción de la imagen referencia se presentó en la Sección 3.2.

Además de los coeficientes, el programa despliega una superposición de las imágenes segmentada y referencia de manera que se pueda llevar a cabo también una inspección visual de los resultados, ver Figura 3.6. La imagen desplegada del volumen segmentado no hace uso del umbral definido por el usuario sino que presenta todos los puntos cuya pertenencia al objeto de interés es mayor a cero.

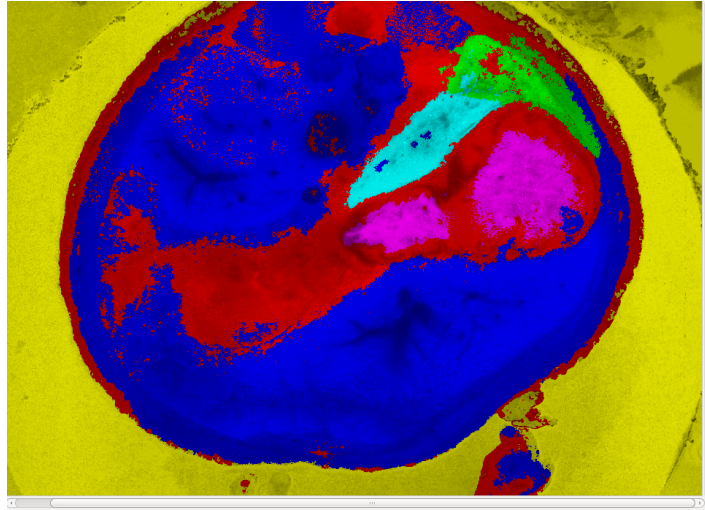


Figura 3.5: Resultado de segmentación con 6 clases llevada a cabo en el espacio $YCbCr$ usando adyacencia π_1 . Cada clase es una región de interés de la Figura 3.4. Las clases etiquetadas con los colores rojo, cian, verde, magenta, azul y amarillo; representan los riñones, el páncreas, el bazo, el interior del estómago, el hígado y el exterior del ratón respectivamente.

Dados los conjuntos X e Y representado las imágenes segmentada y referencia respectivamente, el coeficiente de *Sorensen-Dice* según [54, 55] es calculado como sigue

$$d = \frac{2|X \cap Y|}{|X| + |Y|}, \quad (3.1)$$

donde X e Y contienen los puntos de ambos conjuntos de datos que superan el umbral establecido por el usuario como nivel de confianza.

Por otra parte, los coeficientes de *sensibilidad* (S) y *especificidad* (E) son calculados, según [55] como sigue:

$$S = \frac{PV}{PV + FN}, \quad (3.2)$$

$$E = \frac{NV}{NV + FP}, \quad (3.3)$$

donde PV representa la porción de los puntos clasificados como *positivos verdaderos*, NV los *negativos verdaderos*, FP los *falsos positivos* y FN los *falsos negativos*. Para calcular cada uno de los valores mencionados se tiene en cuenta el valor del umbral establecido por el usuario.

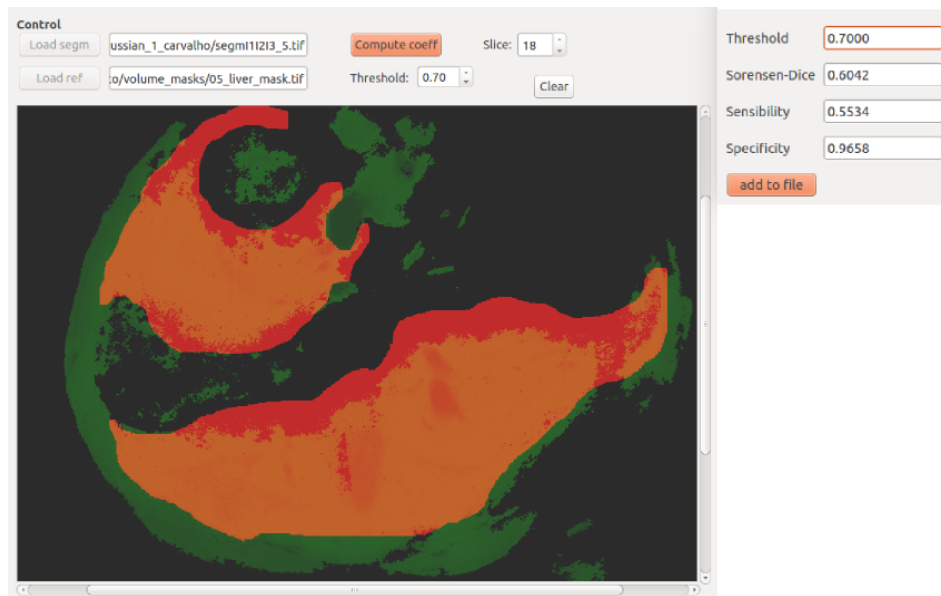


Figura 3.6: Evaluación de resultado de segmentación. Se muestra un corte del hígado segmentado (verde) superpuesto a la misma rebanada del hígado de la imagen referencia (rojo). Para este ejemplo, con un *umbral de confianza* de 0.70, los valores de los coeficientes de *Sorensen-Dice*, *sensibilidad* y *especificidad* fueron respectivamente 0.6042, 0.5534 y 0.9658; lo cual es una indicación de que comparando las imágenes segmentada y referencia se presenta una baja tasa de *positivos verdaderos* y una tasa alta de *negativos verdaderos*.

3.4 Selección de umbral

Como se mencionó en las Secciones 2.4 y 3.3.2, el algoritmo de segmentación produce una gráfica difusa que representa varias clases y a partir de ésta una imagen por cada objeto de interés segmentado. Cada imagen resultado contiene valores reales en el rango $[0, 1]$ indicando la pertenencia de cada punto al objeto en cuestión.

En principio, el algoritmo de segmentación permite segmentar los objetos de la mejor manera posible a partir de la información que obtiene a partir de las *semillas* indicadas por el usuario, es decir, en una “buena” segmentación solo los *vóxeles* que se encuentran dentro de los límites del objeto deberían tener valores “altos” o cercanos a 1 mientras el resto deberían tener valores “bajos” o cercanos a 0; sin embargo, debido a las características de las imágenes del presente trabajo (ver Sección 3.1) se presentan en muchos casos un crecimiento descontrolado de las regiones (con valores de pertenencia “no tan bajos”) más allá de los límites reales del objeto en la imagen digital, fenómeno que de ahora en adelante será referido como “fuga”.

Debido al problema mencionado, se hizo necesario proponer el uso de un *umbral* para determinar de manera general, para el conjunto de datos disponibles, un criterio de parada para el algoritmo de segmentación de manera que se redujeran las “fugas”, es decir que se detuviera el crecimiento de las regiones de los diferentes objetos para niveles de pertenencia menores a éste.

De manera que pudiera ser determinado un valor *umbral*, fueron construidas curvas *ROC* (siglas en inglés de Característica Operativa del Receptor) a partir de los coeficientes de *sensibilidad* y *especificidad* obtenidos a partir diferentes configuraciones de la aplicación encargada de la segmentación (un ejemplo se muestra en la Figura 3.7), dado que este tipo de curvas permiten determinar el comportamiento de un sistema binario de clasificación, específicamente cómo varían la porción de *positivos verdaderos* (*PV*) frente a la porción de *negativos verdaderos* (*NV*) según se varía un *umbral de confianza* [56].

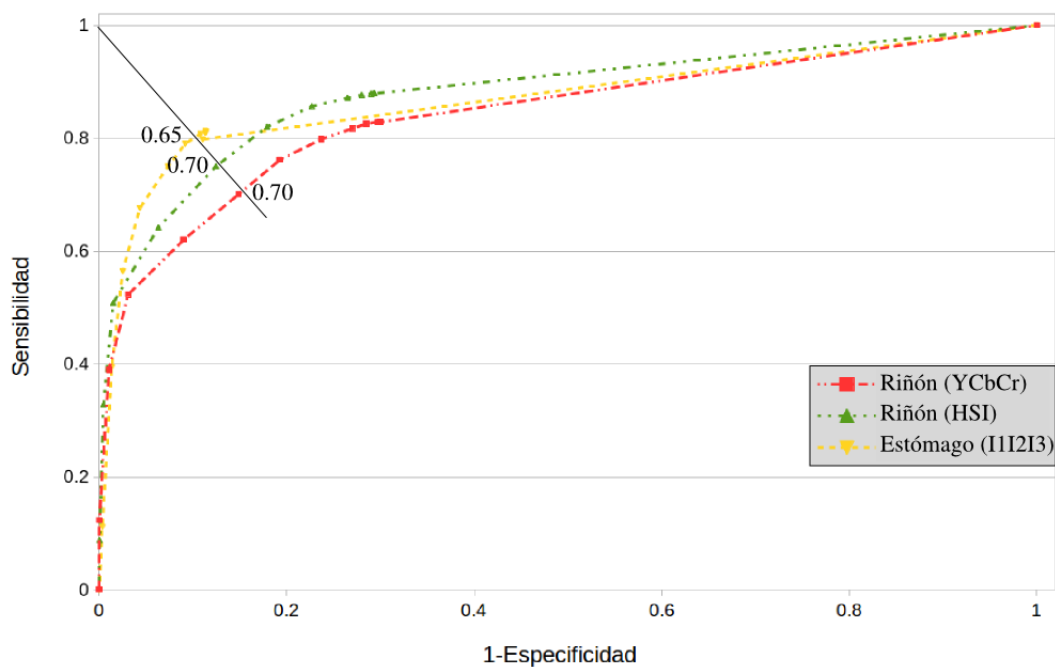


Figura 3.7: Curvas *ROC* para varios órganos y configuraciones de parámetros de segmentación.

La Figura 3.7 muestra varias curvas *ROC* obtenidas para diferentes órganos y configuraciones de parámetros de segmentación. La figura mostrada es un ejemplo representativo de los experimentos realizados y permitió determinar que un *umbral* apropiado para detener el crecimiento de las regiones en la segmentación puede ser un

valor entre 0.65 y 0.70 (0.70 para el presente trabajo).

3.5 Características de los conjuntos de semillas

Como se mencionó en el Capítulo 1, el uso de los conceptos de *conectividad difusa* y *afinidad difusa* aplicados a segmentación de imágenes fue presentado por Udupa y Samarasekera en [16]. En el trabajo mencionado, los resultados de las segmentaciones no dependen de la ubicación de las semillas dentro del objeto a segmentar, es decir, con este enfoque mientras las semillas se encuentren dentro de los límites del objeto de interés, el objeto difuso generado por el algoritmo siempre es el mismo. A diferencia del enfoque mencionado, el introducido por Herman y otros en [17] el cual es el usado en el presente trabajo, aunque su teoría se basa en la presentada en [16], tiene como característica que los objetos difusos generados por el algoritmo de segmentación pueden cambiar según la ubicación y número de semillas.

Dado que los resultados entregados por el algoritmo de segmentación usado en el presente trabajo pueden depender críticamente de la ubicación de las semillas, fueron realizados experimentos usando varios conjuntos de semillas con diferentes características. El objetivo de los experimentos fue determinar las características que deben tener los conjuntos de semillas, de manera que los resultados del proceso de segmentación estuvieran cercanos al rendimiento óptimo de la herramienta.

Para el análisis se propuso el uso de los siguientes conjuntos de semillas: 10 semillas ubicadas en varios cortes, 20 semillas ubicadas en un solo corte, 20-35 semillas ubicadas en diferentes cortes y un conjunto con más de 40 semillas para cada objeto u órgano de interés a ser segmentado, ver Figuras 3.4 y 3.5. Fueron elegidos cuatro espacios de color (*RGB*, *HSI*, *YCbCr* y *CIELab*) para llevar a cabo la segmentación de los órganos de interés. El espacio *RGB* fue elegido por ser el espacio en el cual se encuentran representadas las imágenes originalmente, el espacio *HSI* fue elegido de manera aleatoria entre los espacios *HSL*, *HSV* y *HSI*; el espacio *YCbCr* fue elegido también de forma aleatoria entre los espacios *YUV*, $i_1i_2i_3$ e *YCbCr*; mientras el espacio *CIELab* fue elegido aleatoriamente entre los espacios *CIELab* y *CIELuv*. Los grupos de espacios de color, a partir de los cuales fueron elegidos los cuatro espacios de color del experimento en cuestión, fueron formados a partir de identificar similitudes entre los resultados de segmentación de las imágenes disponibles. Los resultados de cada segmentación fueron evaluados usando la herramienta descrita en la Sección 3.3.4.

Los resultados de los experimentos se muestran en la Figura 3.8. Se muestran cuatro “cubos”, uno por cada espacio de color, en donde los ejes principales corresponden a los coeficientes de *Sorensen-Dice*, *sensibilidad* y *especificidad*. En cada “cubo” se comparan los resultados de la segmentación de los cinco órganos de interés usando los diferentes conjuntos de semillas. Aunque los resultados no muestran una tendencia clara, se puede observar que en general presentan menor desempeño las segmentaciones que hacen uso de los conjuntos de 10 semillas en varios cortes y 20 semillas en un solo corte. En lo relacionado con los conjuntos restantes se puede observar que las segmentaciones presentan desempeños similares.

Es importante mencionar en este punto que el algoritmo de segmentación no debería depender tanto de la cantidad de semillas y de si estas son ubicadas en uno o varios cortes; sin embargo, creemos que debido a que las imágenes axiales que componen el volumen de datos no presentan siempre una iluminación homogénea y que en muchos casos se presenta falta de alineación entre cortes contiguos, estos factores efectivamente afectan el desempeño del algoritmo.

De esta manera, se puede afirmar que por lo menos para este conjunto de datos puede ser usado un conjunto de semillas que tenga entre 20 y 35 elementos en cada objeto de interés a ser segmentado.

3.6 Segmentación en diferentes espacios de color

Una vez definidas las características que debían tener los conjuntos de semillas a usar en el proceso de segmentación así como el umbral a ser usado en la evaluación de los resultados, el siguiente paso fue evaluar la herramienta de segmentación en todos los espacios estándar de color disponibles en la aplicación descrita en la Sección 3.3.2, además de un espacio híbrido de color seleccionado para cada conjunto de semillas diferente.

Para esto se pidió a cinco personas que generaran un conjunto de semillas para el mismo volumen, usando la herramienta descrita en la Sección 3.3.1. El conjunto de semillas generado por cada usuario debía contener seis clases de manera que se pudiera llevar a cabo la segmentación de los órganos de interés como se muestra en la Figura 3.4. A partir de cada conjunto de semillas fueron generados dos archivos, uno por cada adyacencia a utilizar en el proceso de segmentación, es decir π_1 y π_2 . Adicionalmente, la herramienta para selección de semillas sugirió para cada conjunto

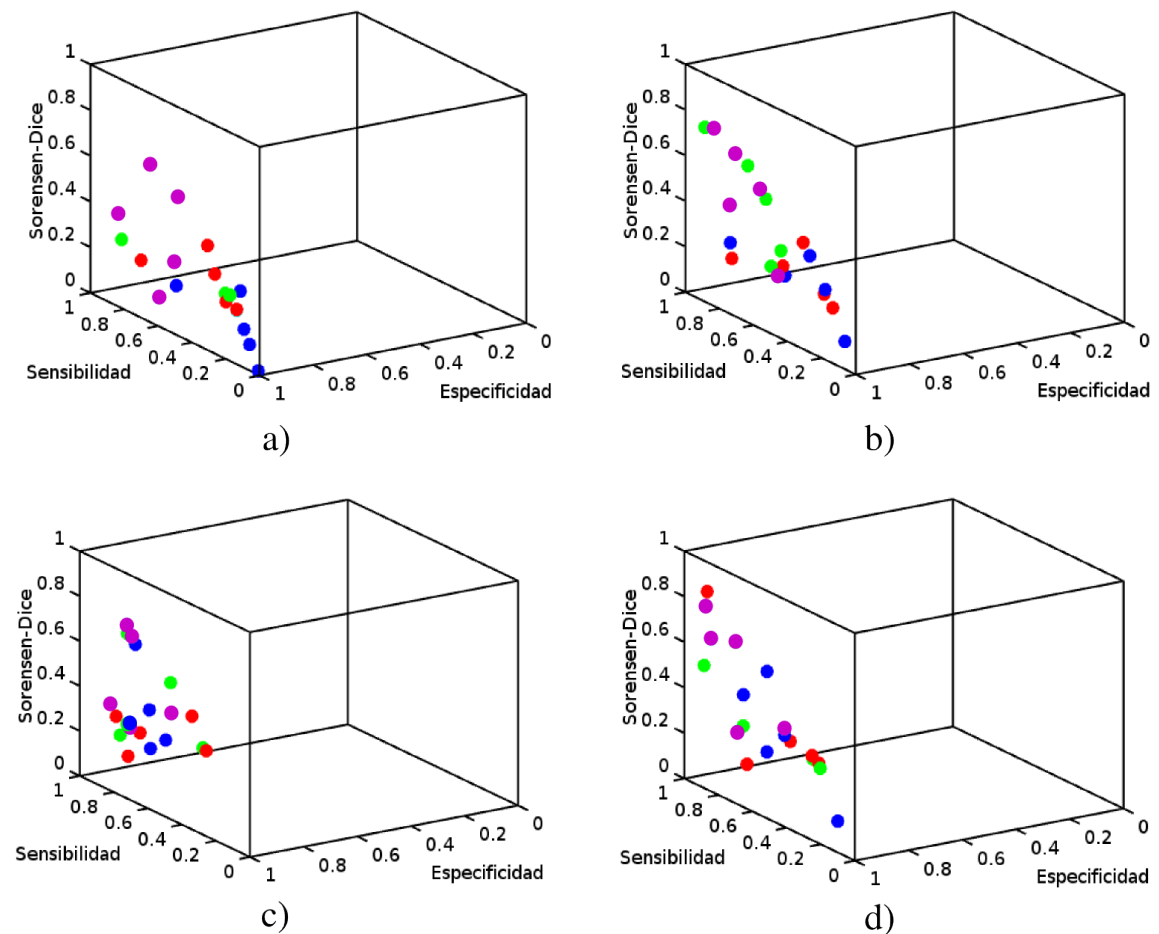


Figura 3.8: Comparación de resultados de segmentación usando los espacios de color a) *RGB*, b) *HSI*, c) *CIELab* y d) *YCbCr*. Fueron usados varios conjuntos de semillas en cada caso: 10 semillas en varios cortes, 20-35 semillas en varios cortes, 20 semillas en un solo corte y un conjunto con 40 o más semillas en varios cortes; los conjuntos están identificados en las gráficas con los colores rojo, verde, azul y magenta, respectivamente.

de semillas dos espacios de color en donde la métrica usada daba indicación de mejor separación entre los objetos seleccionados al igual que un espacio híbrido de color, ver Tabla 3.2.

Posteriormente, cada conjunto de semillas, con sus definiciones de adyacencia y afinidad, generado por los diferentes usuarios fue usado como entrada al programa encargado de llevar a cabo la tarea de segmentación. Los resultados fueron evaluados usando la herramienta que permite calcular los índices de *Sorensen-Dice*, *sensibilidad* y *especificidad* descrita en la Sección 3.3.4. Aunque los programas fueron construidos en forma modular, fue construido un *script* que permitiera automatizar la tarea de

Usuario	Color ₁	Color ₂	Híbrido
J	\bar{c}_{YCbCr}	$\bar{c}_{i_1i_2i_3}$	$(\bar{c}_{HSI})_2 + (\bar{c}_{YUV})_2 + (\bar{c}_{Luv})_3$
C	\bar{c}_{YCbCr}	$\bar{c}_{i_1i_2i_3}$	$(\bar{c}_{YCbCr})_1 + (\bar{c}_{Lab})_3 + (\bar{c}_{Luv})_3$
R	\bar{c}_{YCbCr}	\bar{c}_{YUV}	$(\bar{c}_{i_1i_2i_3})_2 + (\bar{c}_{HSI})_2 + (\bar{c}_{Luv})_3$
M	\bar{c}_{YCbCr}	\bar{c}_{YUV}	$(\bar{c}_{YCbCr})_1 + (\bar{c}_{Lab})_3 + (\bar{c}_{Luv})_3$
L	\bar{c}_{YCbCr}	$\bar{c}_{i_1i_2i_3}$	$(\bar{c}_{i_1i_2i_3})_3 + (\bar{c}_{YCbCr})_1 + (\bar{c}_{Lab})_3$

Tabla 3.2: Recomendación de colores estándar e híbrido realizada por el programa de selección de semillas para los diferentes conjuntos de semillas

segmentar y evaluar los resultados para cada espacio de color comparando con las respectivas imágenes usadas como *ground-truth*. El resultado de la ejecución del *script* para cada conjunto de semillas generado por los usuarios es un archivo de texto donde se listan, para cada espacio de color y un mismo conjunto de semillas, los coeficientes que permiten evaluar la calidad de la segmentación. Un ejemplo del archivo de salida generado a partir de un conjunto de semillas se muestra en el Apéndice B.

Posteriormente, para cada órgano de interés a ser segmentado y para cada adyacencia π_1 y π_2 , fueron construidas tablas como la que se muestra en la Tabla 3.3. Dado que el valor del coeficiente *especificidad* es mayor a 0.90 en la mayoría de los casos, las tablas fueron construidas con los valores medios de los coeficientes de *Dice* y *sensibilidad* y su respectivo valor de ϵ , el cual está relacionado con la variabilidad de los resultados a partir de los diferentes conjuntos de semillas. A partir de dichas tablas fueron construidos gráficos de barra de error como los que se muestran en las Secciones desde la 3.6.1 hasta la 3.6.5 en donde se presentan los resultados para cada órgano de interés.

3.6.1 Interior del estómago

En la Figura 3.9 se pueden observar los resultados de la segmentación del interior del estómago usando todos los espacios de color para las adyacencias π_1 (Figura 3.9a) y π_2 (Figura 3.9b). Los gráficos de barra de error muestran los valores medios para cada espacio de color de los coeficientes de *Dice* y *sensibilidad* para este órgano así como su respectivo error. La longitud de cada barra está relacionada con la variabilidad en los resultados al usar en la segmentación los distintos archivos de semillas generados por los usuarios.

En el caso de los espacios de color *RGB* y *escala de grises* se puede observar mayor variabilidad que en los otros casos. Ambos resultados para el espacio en *escala de grises*

Color	$media_{Dice}$	$media_{sensibilidad}$	σ_{Dice}	$\sigma_{Sensibilidad}$	$\epsilon = \sqrt{\sigma_{Dice}^2 + \sigma_{Sensibilidad}^2}$
<i>RGB</i>	0.4901	0.6522	0.0629	0.2733	0.2804
<i>i₁i₂i₃</i>	0.5707	0.7181	0.0557	0.1073	0.1209
<i>HSI</i>	0.5722	0.8006	0.0356	0.1654	0.1692
<i>HSL</i>	0.5762	0.8007	0.0483	0.1638	0.1708
<i>HSV</i>	0.5358	0.8217	0.0352	0.1583	0.1622
<i>YCbCr</i>	0.6595	0.7892	0.0737	0.1544	0.1711
<i>YUV</i>	0.6587	0.7917	0.0723	0.1525	0.1688
<i>CIELab</i>	0.6871	0.7508	0.0640	0.1238	0.1394
<i>CIELuv</i>	0.6319	0.7868	0.0739	0.1403	0.1585
<i>Gris</i>	0.3449	0.5478	0.1408	0.3046	0.3355
<i>Híbrido</i>	0.5748	0.6956	0.0862	0.1801	0.1996

Tabla 3.3: Ejemplo de los resultados de segmentación para el hígado usando adyacencia π_2 en todos los espacios de color disponibles.

muestran los valores de coeficientes más bajos en comparación con los demás. Se puede observar que los espacios de color que presentan un mejor desempeño son *i₁i₂i₃*, *HSI*, *HSL*, *HSV* y los espacios híbridos de color.

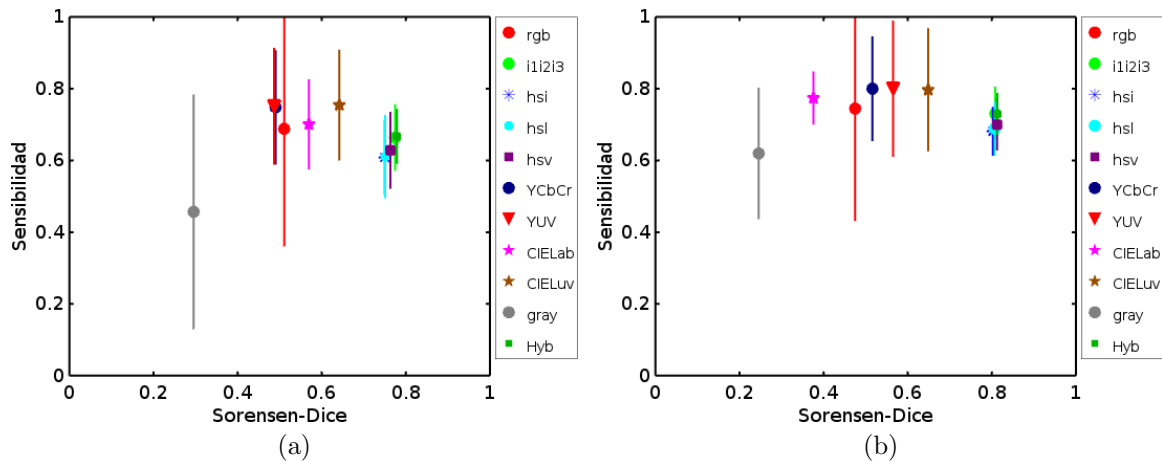


Figura 3.9: Resultados segmentación del interior del estómago con adyacencias (a) π_1 y (b) π_2 .

Este órgano presenta los valores de coeficientes más altos en comparación con los demás órganos de interés posiblemente porque presenta un mayor contraste respecto a los demás órganos para los diferentes espacios de color incluidos en el trabajo, ver Figura 3.10. Los bajos valores de coeficiente en algunos espacios de color pueden ser

debidos a que dichos espacios de color no permiten al algoritmo “diferenciar” el interior del estómago del interior de los intestinos dado que se presenta poco contraste.

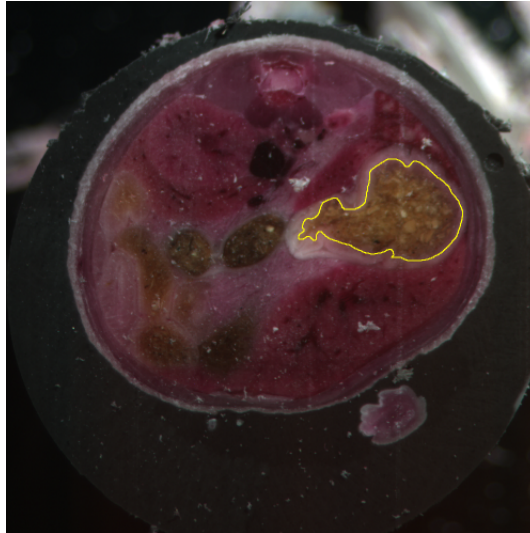


Figura 3.10: Anotación manual del interior del estómago.

En la Figura 3.11b se presenta un renderizado que muestra la superposición de las imágenes segmentada (*cian*) y referencia (*rojo*), mientras en la Figura 3.11a se muestra únicamente la imagen referencia. La segmentación de este ejemplo fue realizada en el espacio de color $i_1i_2i_3$, con una adyacencia π_2 y para un valor de umbral de 0.7 los coeficientes de *Sorensen-Dice*, *sensibilidad* y *especificidad* son 0.8495, 0.8296 y 0.9986, respectivamente.

3.6.2 Hígado

En el caso del hígado los resultados se pueden observar en la Figura 3.12. Así como en el caso anterior, los espacios de color que presentan valores más bajos en los coeficientes y una mayor variabilidad son los espacios *RGB* y *escala de grises*. Por otra parte, para este órgano los espacios de color que presentan mejores resultados son: *CIELab*, *CIELuv* y *YUV* en ambos casos. En el caso de los espacios híbridos de color el desempeño es apenas inferior a los espacios mencionados.

Este órgano presenta un tono más uniforme a pesar de la falta de homogeneidad en la iluminación en varios cortes contiguos, presenta también menos textura comparado con los otros órganos de interés; sin embargo, se puede observar que su tono presenta similitud con el que se presenta en los riñones y el tejido muscular que rodea la

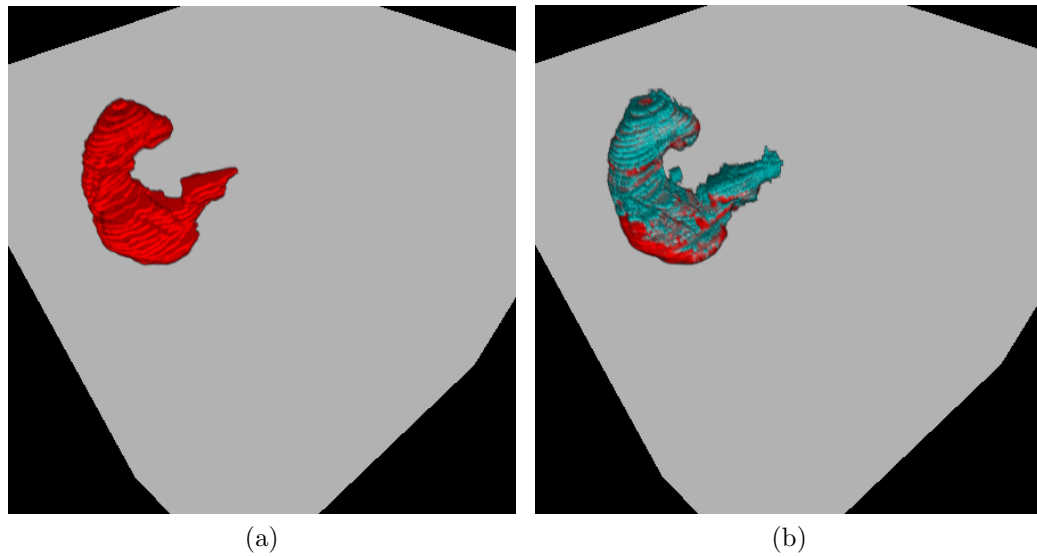


Figura 3.11: Renderizado para el interior del estómago. a) Imagen referencia (*ground-truth*) y b) Superposición del resultado de la segmentación con la imagen referencia. La segmentación fue realizada en el espacio de color $i_1i_2i_3$ usando adyacencia π_2 .

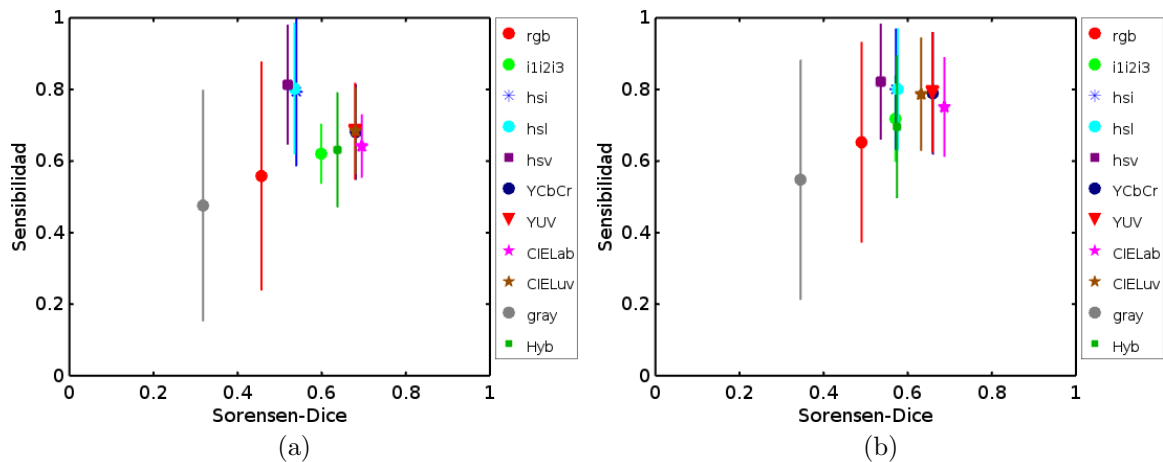


Figura 3.12: Resultados segmentación del hígado con adyacencias (a) π_1 y (b) π_2 .

zona abdominal, lo cual puede ser una causa que explique los valores de coeficientes calculados, sumado también lo anterior a la presencia de vascularidad en la región que corresponde al hígado, ver Figura 3.13.

En la Figura 3.14b se presenta un renderizado que muestra la superposición de las imágenes segmentada y referencia, mientras en la Figura 3.14a se muestra únicamente la imagen referencia. La segmentación de este ejemplo fue realizada en el espacio de

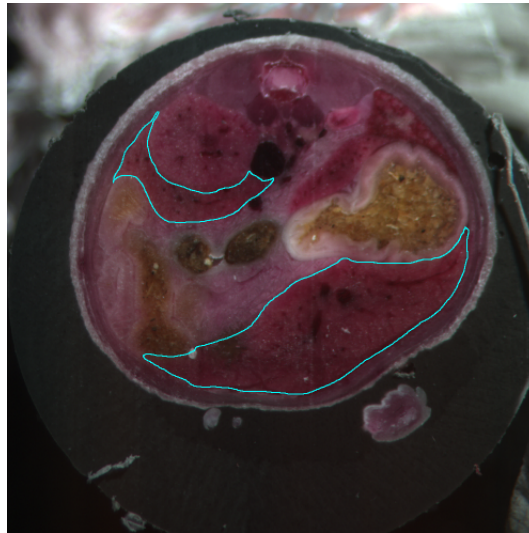


Figura 3.13: Anotación manual del hígado.

color $YCbCr$, con una adyacencia π_1 y para un valor de umbral de 0.7 los coeficientes de *Sorensen-Dice*, *sensibilidad* y *especificidad* son 0.7356, 0.7622 y 0.9624 respectivamente.

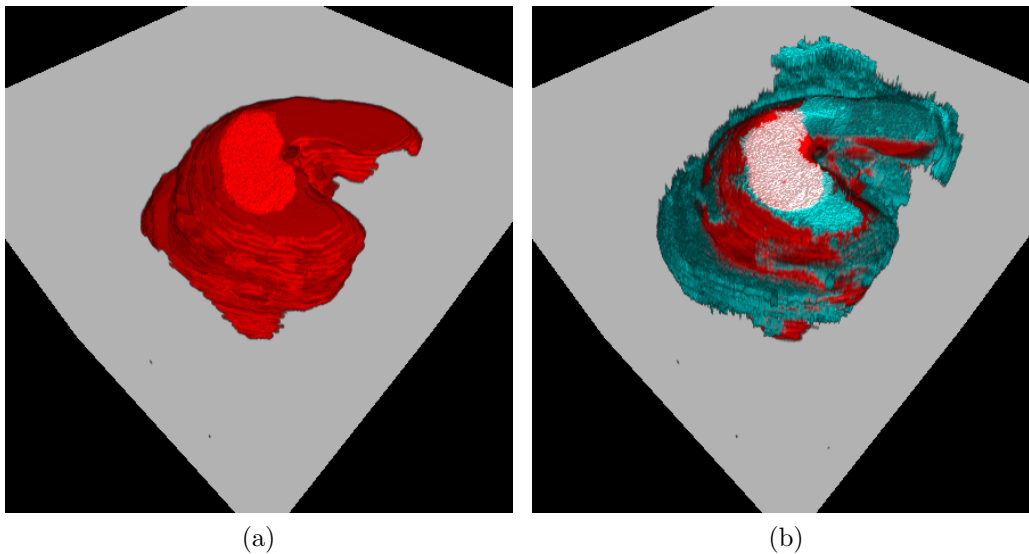


Figura 3.14: Renderizado para el hígado. a) Imagen referencia (*ground-truth*) y b) Superposición del resultado de la segmentación con la imagen referencia. La segmentación fue realizada en el espacio de color $YCbCr$ usando adyacencia π_1 . En las regiones blancas se presenta mayor coincidencia entre las imágenes.

3.6.3 Páncreas

En el caso de este órgano los resultados se pueden observar en la Figura 3.15. Así como en el caso anterior los espacios de color que presentan valores más bajos en los coeficientes y una mayor variabilidad son los espacios *RGB* y *escala de grises*. Por otra parte, para este órgano los espacios de color que presentan mejores resultados son: $i_1i_2i_3$ e *YCbCr* en ambos casos. Los resultados presentan una variabilidad considerable en casi todos los espacios de color. Otra característica que se puede resaltar es que en este caso a diferencia de los resultados descritos para los anteriores órganos es que se presentan, aunque no en todos los casos, menores valores de coeficientes para la adyacencia π_2 . Este órgano es el más pequeño de los órganos de interés en el presente

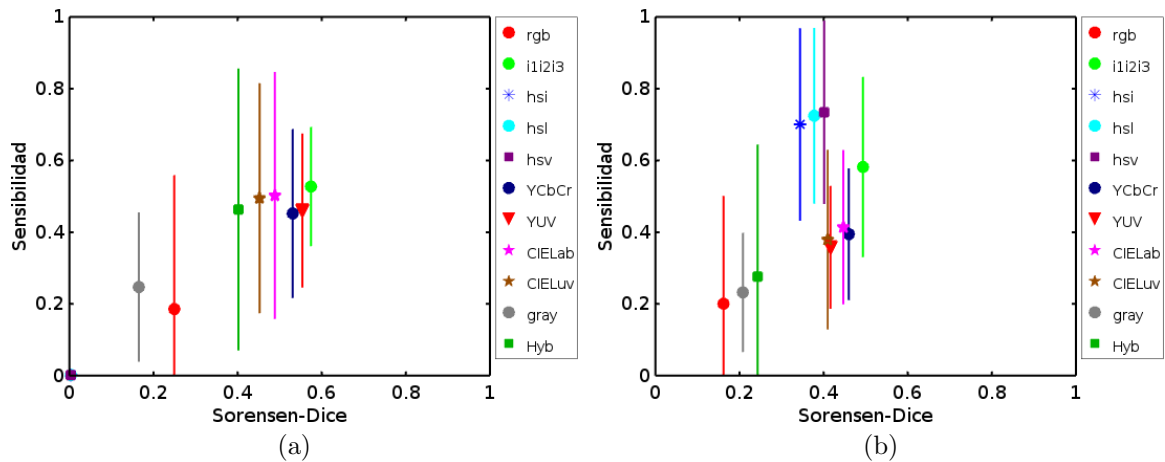


Figura 3.15: Resultados segmentación del páncreas con adyacencias (a) π_1 y (b) π_2 .

trabajo, con una presencia considerable de textura, además de que su cercanía con el estómago no permite definir bien sus límites en varios de los cortes, ver Figura 3.16.

En la Figura 3.17b se presenta un renderizado que muestra la superposición de las imágenes segmentada y referencia, mientras en la Figura 3.17a se muestra únicamente la imagen referencia. La segmentación de este ejemplo fue realizada en el espacio de color $i_1i_2i_3$, con una adyacencia π_1 y para un valor de umbral de 0.7 los coeficientes de *Sorensen-Dice*, *sensibilidad* y *especificidad* son 0.6770, 0.66134 y 0.9989, respectivamente.

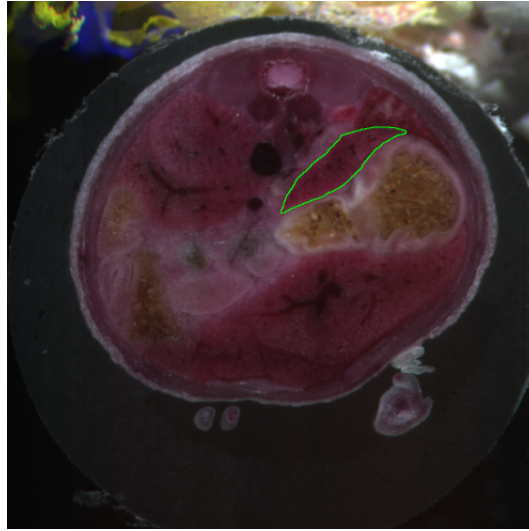


Figura 3.16: Anotación manual del páncreas.

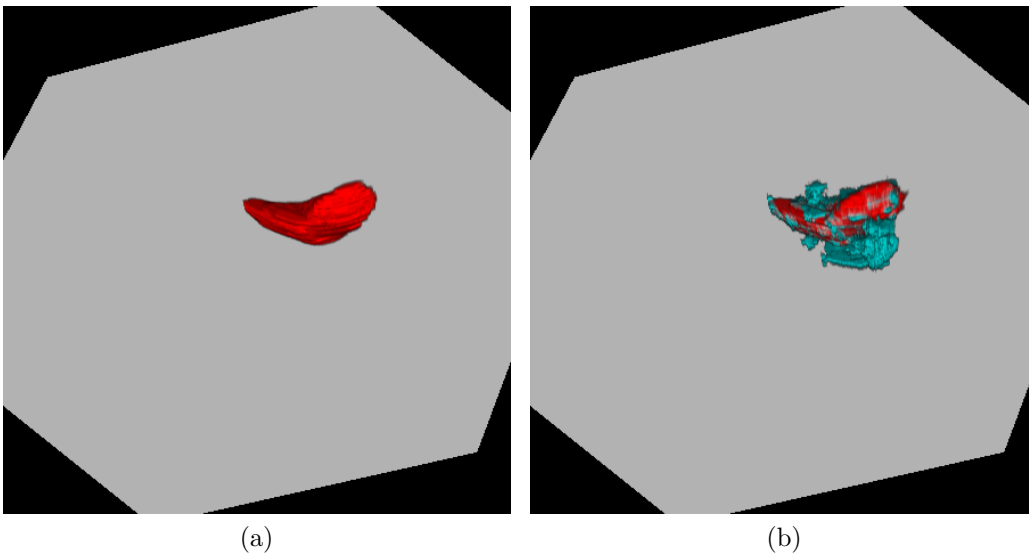


Figura 3.17: Renderizado para el páncreas. a) Imagen referencia (*ground-truth*) y b) Superposición del resultado de la segmentación con la imagen referencia. La segmentación fue realizada en el espacio de color $i_1i_2i_3$ usando adyacencia π_1 .

3.6.4 Bazo

Los resultados de los valores medios de los coeficientes y su respectivo error se muestra en la Figura 3.18. Se puede observar un desempeño similar y hasta menor en las segmentaciones de realizadas en los espacios: *HSI*, *HSL* y *HSV* en relación con aquellas realizadas en *niveles de gris* y *RGB* (espacios de color que hasta ahora habían

presentado los valores más bajos). En ambos casos se observa un desempeño similar para los espacios de color: *YCbCr*, *YUV*, *CIELab*, *CIEluv* e híbridos. Éste es un órgano

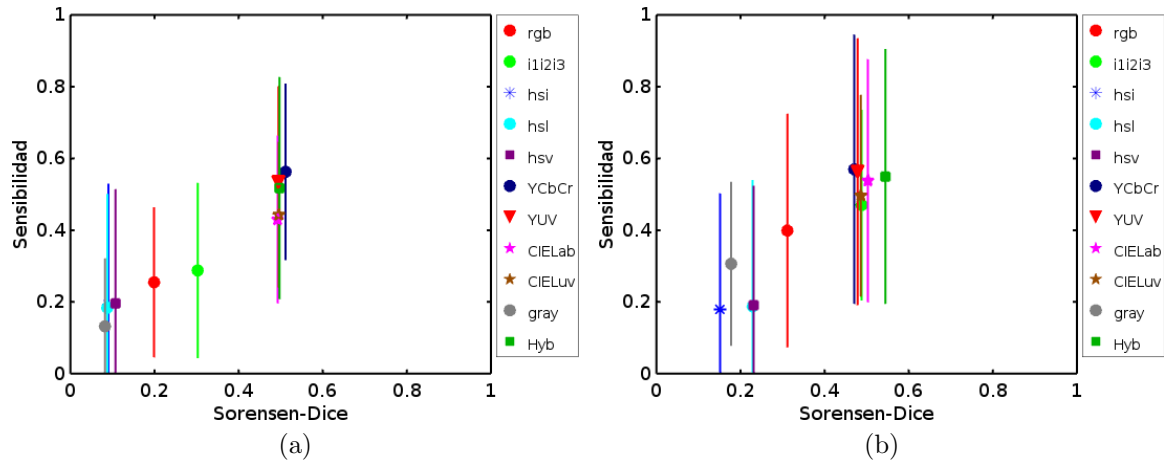


Figura 3.18: Resultados segmentación del bazo con adyacencias (a) π_1 y (b) π_2 .

relativamente pequeño aunque su tamaño supera al del páncreas, con una presencia considerable de textura lo cual afecta de forma negativa el desempeño del algoritmo de segmentación, ver Figura 3.19.

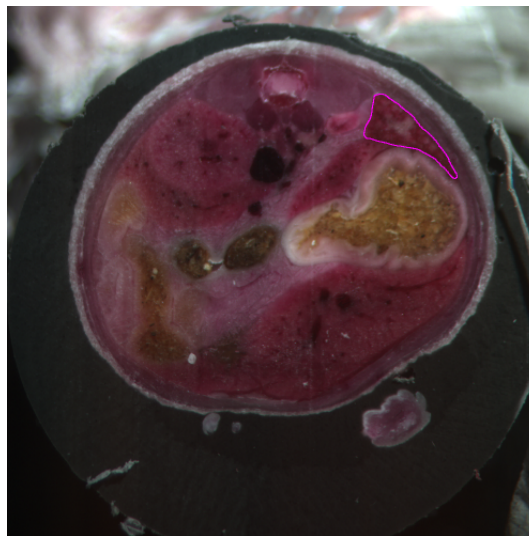


Figura 3.19: Anotación manual del bazo.

En la Figura 3.20b se presenta un renderizado que muestra la superposición de las imágenes segmentada y referencia, mientras en la Figura 3.20a se muestra únicamente la imagen referencia. La segmentación de este ejemplo fue realizada en el espacio

de color *CIELab*, con una adyacencia π_2 y para un valor de umbral de 0.7 los coeficientes de *Sorensen-Dice*, *sensibilidad* y *especificidad* son 0.6820, 0.7474 y 0.9970, respectivamente: .

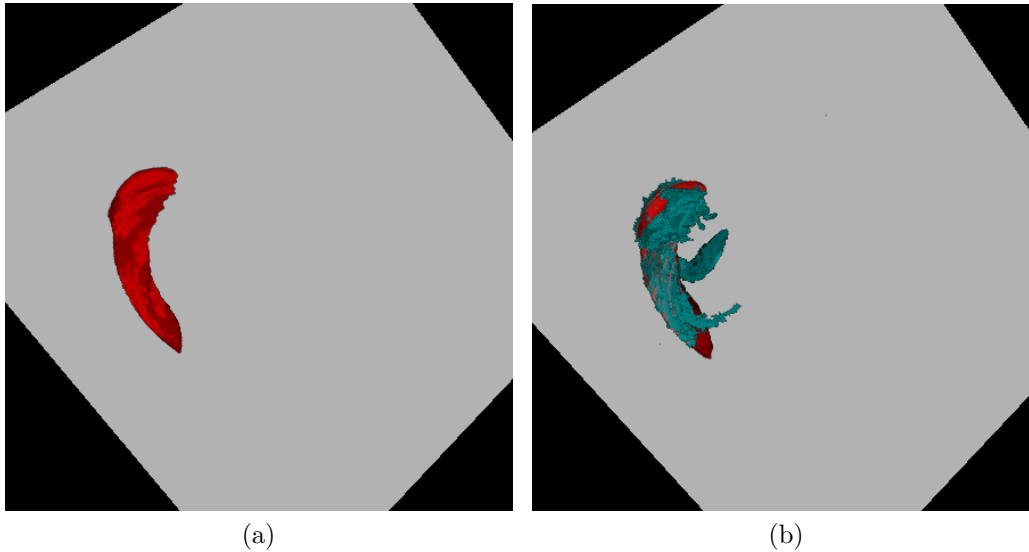


Figura 3.20: Renderizado para el páncreas. a) Imagen referencia (*ground-truth*) y b) Superposición del resultado de la segmentación con la imagen referencia. La segmentación fue realizada en el espacio de color *CIELab* usando adyacencia π_2 .

3.6.5 Riñones

Los resultados de las segmentaciones para este órgano se muestran en la figura 3.21. A diferencia de los órganos anteriormente mencionados, en este caso no se presenta una tendencia similar en el caso de las dos adyacencias. Mientras en el caso de la adyacencia π_1 (Figura 3.21a) los espacios de color *HSI*, *HSL* y *HSV* presentan los valores más bajos de coeficientes; en el caso de la adyacencia π_2 éstos presentan los mejores valores de coeficientes. Este órgano presenta los menores valores de coeficientes en comparación con los obtenidos para los demás órganos de interés. Se presenta en mayor medida el crecimiento descontrolado de la región que el algoritmo reportado como perteneciente a este órgano (“fugas”), especialmente hacia la región del tejido muscular que se ubica al rededor de la región abdominal, ver Figura 3.22.

En la Figura 3.23b se presenta un renderizado que muestra la superposición de las imágenes segmentada y referencia, mientras en la Figura 3.23a se muestra únicamente la imagen referencia. La segmentación de este ejemplo fue realizada en

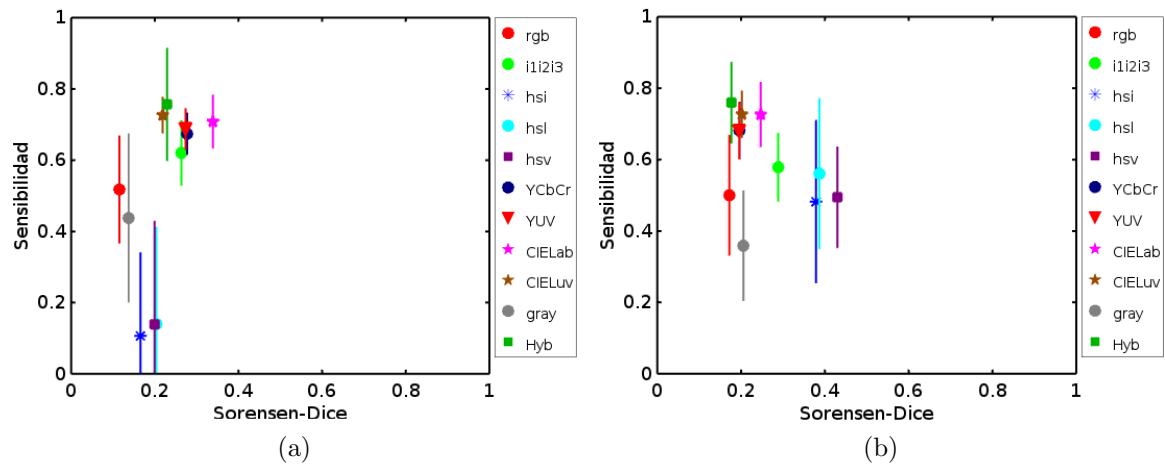


Figura 3.21: Resultados segmentación de los riñones con adyacencias (a) π_1 y (b) π_2 .

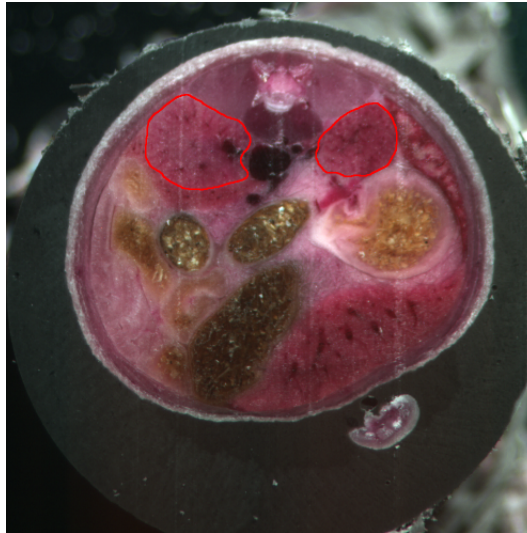


Figura 3.22: Anotación manual de los riñones.

el espacio de color *HSV*, con una adyacencia π_2 y para un valor de umbral de 0.7 los coeficientes de *Sorensen-Dice*, *sensibilidad* y *especificidad* son 0.4437, 0.5741 y 0.9797, respectivamente.

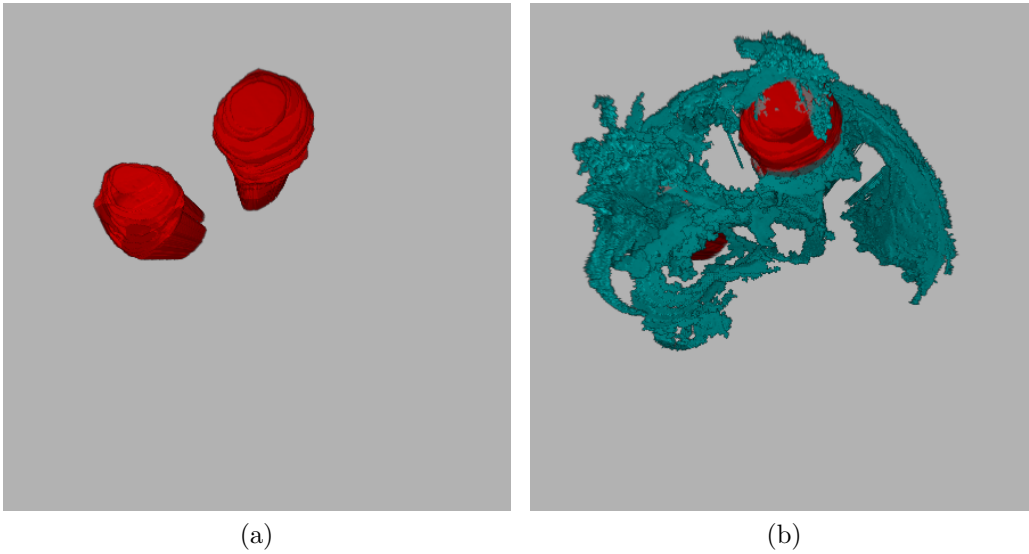


Figura 3.23: Renderizado para los riñones. a) Imagen referencia (*ground-truth*) y b) Superposición del resultado de la segmentación con la imagen referencia. La segmentación fue realizada en el espacio de color *HSV* usando adyacencia π_2 .

Capítulo 4

Conclusiones

En el presente trabajo fue puesta a prueba la utilidad del método de segmentación basado en principios de *lógica difusa* usando una imagen 3D construida a partir de un conjunto de imágenes a color de alta resolución de cortes axiales de ratón. El trabajo se enfocó en segmentar de manera simultánea los órganos del interior del abdomen de uno de los especímenes seleccionados. La selección del conjunto de datos consistió en evaluar el conjunto de imágenes que presentara el menor número de problemas de registro entre cortes contiguos.

De manera que los resultados de las segmentaciones pudieran ser evaluados cuantitativamente fue construido un conjunto de datos a ser usado como *ground-truth*, es decir, un conjunto de imágenes que pudieran ser usadas como referencia, una por cada órgano de interés. Este conjunto fue construido a partir de anotaciones manuales hechas por cuatro diferentes personas, quienes hicieron uso del atlas disponible de manera pública en la red el cual hace parte del proyecto *DigiMouse* [52].

En cuanto al programa para llevar a cabo la segmentación de las imágenes a color, éste fue obtenido a partir de la modificación de una implementación disponible en el grupo de trabajo. Las modificaciones permitieron a éste tener la capacidad de llevar a cabo el proceso de segmentación en imágenes representadas en diferentes espacios de color. La herramienta modificada puede llevar a cabo la segmentación en diez diferentes espacios estándar de color: *RGB*, $i_1i_2i_3$, *HSI*, *HSL*, *HSV*, *YCbCr*, *YUV*, *CIELab*, *CIELuv* y *escala de grises*; así como en un espacio *híbrido* de color el cual se forma a partir de 3 de las componentes de color de los espacios estándar mencionados antes. Dadas las características de las imágenes disponibles, falta de homogeneidad en la iluminación entre cortes contiguos así como falta de alineación entre algunos de

ellos, se llevó a cabo un experimento que permitió determinar que, por lo menos para el conjunto de datos seleccionado, cada objeto se encontraba bien representado por un conjunto de semillas con un número de elementos entre 25 y 30 semillas.

Como parte del trabajo también fueron construidos herramientas de software que permiten: seleccionar semillas, visualizar y evaluar los resultados de la segmentación. La herramienta de selección de semillas, como se mostró en la Sección 3.3.1, permite al usuario mediante un programa con interfaz de usuario definir los objetos de interés a ser segmentados a partir de la ubicación de conjuntos de semillas que definen cada objeto, así como la función de afinidad y el tipo de adyacencia a ser usada durante la segmentación del mismo. La herramienta construida para visualizar los resultados permite cargar las imágenes resultado de la segmentación, las cuales se obtienen a partir del mapa de conectividad a la salida del programa que ejecuta la segmentación, asignando un color diferente a cada objeto segmentado. La utilidad de la herramienta de visualización es más de carácter cualitativo que cuantitativo, aunque también permite obtener información de la pertenencia de los puntos de la imagen a sus respectivas clases. A este programa de visualización, fue agregado un módulo que permite construir un modelo tri-dimensional de los datos de entrada haciendo uso de la técnica de graficación por computador conocida como *Volume rendering*. El resultado de las segmentaciones, dentro de este módulo, son usados como *función de transferencia multi-dimensional* para asignar propiedades de color y opacidad a la imagen original. Dentro del trabajo, la herramienta de renderizado 3D fue implementada para visualizar los resultados de la segmentación, así como para comparar cualitativamente los resultados de la segmentación respecto a los datos construidos como *ground-truth*. Finalmente, la herramienta construida para la evaluación cuantitativa de las segmentaciones permite calcular los coeficientes de *Sorensen-Dice*, *sensibilidad* y *especificidad* comparando la segmentación para cada objeto respecto a la imagen referencia para el mismo objeto según un umbral de confianza. En este trabajo fue propuesto el uso de curvas *ROC* para encontrar el valor de umbral óptimo para ser usado en la evaluación cuantitativa de los resultados de las segmentaciones. El valor hallado en el presente trabajo para el conjunto de datos utilizado fue de 0.70. Este valor fue también utilizado como criterio de parada en el algoritmo de segmentación de manera que se presentara una reducción del crecimiento sin control de las regiones más allá de los límites reales del objeto segmentado.

Para explorar la utilidad de las herramientas construidas y/o modificadas fue propuesto un experimento en el que, haciendo uso de diferentes conjuntos de semillas

generados por 5 voluntarios, se llevaron a cabo las segmentaciones de los siguientes órganos: riñones, páncreas, bazo, interior del estómago e hígado; usando todos los espacios de color disponibles en la herramienta de segmentación, así como en un espacio híbrido de color recomendado por la herramienta de selección de semillas a partir del conjunto de semillas generado por cada usuario. Las medidas de desempeño utilizadas fueron los coeficientes de *Sorensen-Dice*, *sensibilidad* y *especificidad*. Sin embargo, dado que para el experimento planteado en el 92% de los casos el coeficiente de *especificidad* presentó un valor mayor a 0.90, fueron usados únicamente los coeficientes de *Sorensen-Dice* y *sensibilidad* para construir gráficos de barra de error que permitieron evaluar cuantitativamente las segmentaciones obtenidas en los diferentes espacios de color. Posiblemente debido a que en el proceso de segmentación son usadas estadísticas de primer orden de las componentes de color en los diferentes espacios, las cuales permiten hacer medidas de intensidad media y homogeneidad de los objetos mediante el uso de la función de afinidad, se encontró que la herramienta presenta mejor desempeño cuando las imágenes contienen menos textura además de un mayor contraste del objeto de interés frente a los otros presentes en la imagen. Se pudo observar en la mayoría de los resultados de segmentaciones obtenidos con los diferentes conjuntos de semillas generados por los voluntarios, que los espacios estándares de color así como el espacio híbrido de color recomendados por la herramienta de selección de semillas, efectivamente proporcionan un buen desempeño en comparación con los otros espacios de color utilizados. Lo anterior contrasta con los resultados reportados a partir de las segmentaciones realizadas en los espacios *RGB* y *escala de grises* los cuales en la mayoría de los casos presentaron el menor desempeño.

4.1 Trabajo futuro

Una vez analizados los resultados, se puede observar que una de las primeras tareas que se desprende del presente trabajo es la inclusión en el algoritmo de segmentación de medidas de textura presente los objetos de interés. Ya sea que éstas sean realizadas de manera tradicional en alguna medida de intensidad de la representación *RGB* o se proponga una medida de textura directamente en el espacio de color donde se pretenda llevar a cabo el proceso de segmentación.

Dado que uno de los puntos débiles de la herramienta de segmentación usada en el presente trabajo frente a los datos disponibles, es el crecimiento sin control más

allá de los límites reales del objeto de interés por parte de las regiones resultado de la segmentación, se propone como estrategia para tratar de mejorar esta situación: explorar las posibilidades de incorporar a la función de afinidad la capacidad de hacer uso de medidas locales relacionadas con la presencia de bordes. Dichas medidas pueden realizarse en “ventanas” cuadradas, por ejemplo de tamaño 5×5 o 7×7 , las cuales contengan el *spel* a partir del cual la región pretende crecer en cada iteración. Por otra parte, otro enfoque que podría ser explorado es el de llevar a cabo un procesamiento previo de las imágenes que permita generar contornos a diferentes escalas de los diferentes objetos en éstas. La información de los contornos hallados puede ser usada en el algoritmo para tratar de prevenir también el crecimiento sin control de las regiones pertenecientes a cada clase.

Finalmente, un trabajo importante para realizar es la actualización de la herramienta de segmentación para que ésta funcione con el algoritmo *fast-MOFS2*. Este algoritmo fue reportado en [50] y allí se afirma que éste presentó mejor desempeño computacional frente al *fast-MOFS* el cual es usado en el presente trabajo.

Apéndice A

Renderizado directo de volumen

La visualización directa de volúmenes o *Volume rendering* implica el otorgamiento de propiedades de visuales, generalmente color y opacidad, a todas las posiciones del dominio del volumen de datos digitalizado (ver Sección 2.1) para posteriormente proyectarlas sobre la pantalla. Dichas propiedades se asignan mediante los siguientes mapeos:

$$\phi : v(\mathbf{k}) \rightarrow \Theta \quad , \quad (\text{A.1})$$

$$\alpha : v(\mathbf{k}) \rightarrow [0, 1] \quad , \quad (\text{A.2})$$

donde Θ representa vectores de la forma $\bar{c}_{rgb} = (c_r, c_g, c_b)^T \in \mathcal{C}_{rgb} \subset \mathbb{R}^3$ y las componentes $c_r, c_g, c_b \in [0, 1]$; mientras $\alpha \in [0, 1]$ representa la opacidad. Un valor de $\alpha = 1$ implica que el material es totalmente opaco y un valor de $\alpha = 0$ implica que éste es transparente.

La idea básica del método de renderizado es modelar el transporte de luz en materiales gaseosos. Se hace la suposición de que un haz de luz enviado a través de la región del espacio ocupada por el gas atraviesa ésta sin que se presenten fenómenos de absorción, emisión, reflexión o refracción. La imagen final, cuando se aplica este método, se forma conceptualmente de la siguiente manera según se presentó en [2] (ver Figura A.1): se asume que el conjunto $v(\mathbf{k})$ se encuentra entre una fuente de luz l y el plano que representa la pantalla P y para cada píxel de la pantalla es enviado un rayo de luz desde l hacia P . El rayo en cuestión entra al vóxel más cercano a la fuente de luz con una intensidad inicial de I_i y sale por el más cercano a la pantalla con una intensidad I'_i . Dicha intensidad es calculada como la integral de línea usando el color y

la opacidad asignada a cada vóxel del volumen atravesado por el rayo de luz (proceso conocido como *composición*). El proceso de *composición*, conocido también como *alpha*

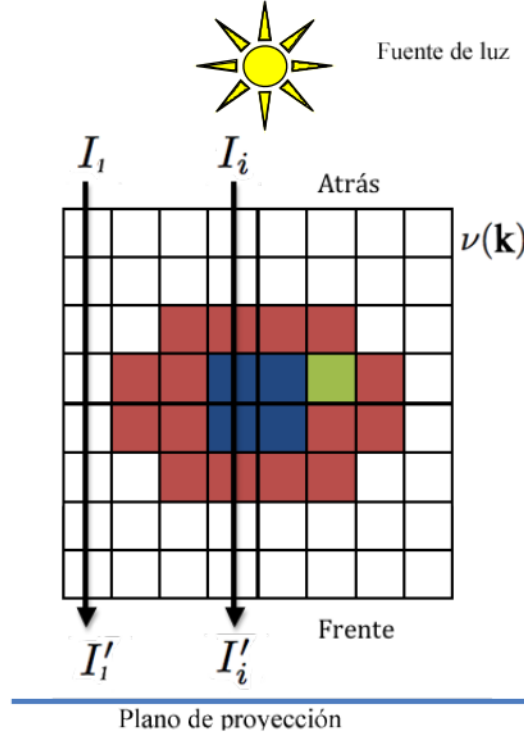


Figura A.1: Intuición del proceso de *Volume rendering*, cortesía de [2]

blending, permite calcular para cada píxel en la pantalla un color y un valor de opacidad por medio de la función $g : R_{\mathbf{u}} \rightarrow (\Theta, [0, 1])$, donde $R_{\mathbf{u}}$ es un subconjunto de los vóxeles de la imagen 3D. Para cada píxel \mathbf{u} en la pantalla (plano discretizado P_{Δ}) se hace uso de la línea recta definida como

$$\mathbf{x}(\mathbf{u}, \vec{\sigma}) = \mathbf{u} + \rho \vec{\sigma}, \quad (\text{A.3})$$

donde $\vec{\sigma}$ es un vector unitario de dirección perpendicular al plano formado por P_{Δ} y ρ es un número real positivo. El conjunto $R_{\mathbf{u}} = \{\mathbf{k}_1, \dots, \mathbf{k}_l, \dots, \mathbf{k}_L\}$ está formado por los vóxeles del volumen intersecados por la línea $\mathbf{x}(\mathbf{u}, \vec{\sigma})$ que parte de un píxel \mathbf{u} en la pantalla (L puede considerarse la longitud equivalente a la suma de los tramos de intersección entre los *vóxeles* y el rayo).

El color final para un píxel \mathbf{u} en la pantalla es calculado a partir de las contribuciones individuales de las propiedades visuales, color y opacidad, de cada vóxel en $R_{\mathbf{u}}$. Una

fórmula ampliamente utilizada para llevar a cabo la composición a partir del conjunto $R_{\mathbf{u}}$ fue presentada en [57]

$$g(R_{\mathbf{u}}) = \sum_{l=1}^L \left[\phi(\mathbf{k}_{L-l-1}) \alpha(\mathbf{k}_{L-l-1}) \prod_{m=l+1}^L (1 - \alpha(\mathbf{k}_{L-l-1-m})) \right], \quad (\text{A.4})$$

donde $\phi(\mathbf{k}_L)$ representa el color del fondo y $\alpha(\mathbf{k}_L) = 1$. Esta ecuación se obtiene mediante la consideración de que solo ocurren cambios locales en la iluminación. Una manera de entender su funcionamiento es analizar el cambio de iluminación dentro de un cilindro muy delgado cuyo eje central es la recta $\mathbf{x}(\mathbf{u}, \vec{\mathbf{o}})$ el cual atraviesa la región ocupada por el volumen [58]. La luz pasa de un extremo del cilindro al otro y a su salida se puede obtener el color y opacidad que debe ser asignada al píxel \mathbf{u} . Como se presentó en [59] la intensidad (I) de un rayo de luz que atraviesa un cilindro de estas características puede ser modelado como sigue

$$\frac{dI}{ds} = q(s)\kappa(s) - I(s)\kappa(s), \quad (\text{A.5})$$

donde κ es el coeficiente de atenuación, esto es, la cantidad de luz entrante que se espera sea extinguida por unidad de longitud; mientras q representa la luz emitida en la coordenada s (posición a lo largo del cilindro). La solución (A.5) es conocida como la ecuación de modelo óptico de absorción-emisión y tiene la forma

$$I(l) = I_0 e^{-\int_0^l \kappa(t) dt} + \int_0^l q(s)\kappa(s) e^{-\int_s^l \kappa(t) dt} ds. \quad (\text{A.6})$$

El primer término representa la cantidad de luz entrante I_0 que llega al punto a una distancia l de la entrada. El segundo término representa la cantidad de luz emitida en cada punto del rayo en su camino, tomando en cuenta la atenuación. A partir (A.6) puede obtenerse una solución en la forma discreta presentada en (A.4). De esta manera, el proceso de composición puede ser considerado como una integral de línea en donde es sumada la contribución de cada vóxel a la imagen final.

Un método utilizado con frecuencia para solucionar (A.6) es discretizar el rayo en subintervalos $s_0 < s_1 \dots < s_n = l$. Entonces, en el intervalo $[s_{i-1}, s_i]$ (A.6) es:

$$I(s_i) = I(s_{i-1})e^{-\int_{s_{i-1}}^{s_i} \kappa(t) dt} + \int_{s_{i-1}}^{s_i} q(s)e^{-\int_s^{s_i} \kappa(t) dt} ds. \quad (\text{A.7})$$

Ahora, si se introduce la notación:

$$\begin{aligned} t_i &= e^{-\int_{s_{i-1}}^{s_i} \kappa(t) dt} , \\ c_i &= \int_{s_{i-1}}^{s_i} q(s) e^{-\int_s^{s_i} \kappa(t) dt} ds , \\ I_i &= I(s_i), \end{aligned}$$

donde t_i es un término que permite medir la transparencia en el intervalo $[s_{i-1}, s_i]$ y c_i es un término que representa la cantidad de luz emitida en el mismo intervalo, entonces la relación (A.7) se representa de manera recursiva para evaluar la integral en sentido *back-front*:

$$I_i = I_{i-1}t_i + c_i . \quad (\text{A.8})$$

Ahora bien, para evaluar la integral en sentido *front-back*, se calcula I_i a partir de I_{i+1} usando dos pasos:

$$I_i = I_{i+1} + T_{i+1}c_i \quad (\text{A.9})$$

y

$$T_i = T_{i+1}t_i . \quad (\text{A.10})$$

Para $i = n - 1 \dots, 1$. Una posible inicialización es $I_n = 0$, $T_n = 1$. Este último conjunto de ecuaciones se puede relacionar con el método de la composición- α haciendo las siguientes relaciones: color de salida o color acumulado $\bar{c}_{out}^{(i)} \equiv I_i$, el color de entrada o color del intervalo actual $\bar{c}_{in}^{(i)} \equiv c_i$, la opacidad acumulada o de salida $\alpha_{out}^{(i)} \equiv 1 - T_i$, la opacidad de entrada $\alpha_{in}^{(i)} \equiv t_i$. Entonces el anterior conjunto de ecuaciones se puede escribir en una forma más común en términos de composición- α :

$$\bar{c}_{out}^{(i+1)} = \bar{c}_{out}^{(i)} + (1 - \alpha_{out}^{(i)})\bar{c}_{in}^{(i)} \quad (\text{A.11})$$

y

$$\alpha_{out}^{(i+1)} = \alpha_{out}^{(i)} + (1 - \alpha_{out}^{(i)})\alpha_{in}^{(i)}, \quad (\text{A.12})$$

con unos valores de inicialización $\bar{c}_{out}^{(i)} = (0, 0, 0)^T$ y $\alpha_{out} = 0$. En el caso del presente trabajo, en el cual una imagen 3D resultado de un proceso de segmentación es usada como *función de transferencia*, el color para cada vóxel a ser renderizado proviene del color asignado al vóxel correspondiente en el volumen segmentado (\bar{c}_{in}) mientras la opacidad (α_{in}) puede ser modificada por el usuario de la herramienta de *software* según

el órgano que quiera resaltar o hacer más visible en el proceso de renderizado.

La relación mostrada en (A.11) funciona para colores *asociados*, es decir pre-multiplicados por α . Para colores no asociados, se sustituye $\bar{c}_{in}^{(i)}$ por $\bar{c}_{in}^{(i)}\alpha_{in}^{(i)}$, entonces cada iteración se calcula:

$$\bar{c}_{out}^{(i+1)} = \bar{c}_{out}^{(i)} + (1 - \alpha_{out}^{(i)})\bar{c}_{in}^{(i)}\alpha_{in}^{(i)} \quad (\text{A.13})$$

y

$$\alpha_{out}^{(i+1)} = \alpha_{out}^{(i)} + (1 - \alpha_{out}^{(i)})\alpha_{in}^{(i)}. \quad (\text{A.14})$$

Es habitual aproximar las integrales considerando el integrando constante en cada sub-intervalo. Entonces, considerando que el coeficiente de atenuación κ es constante en el vóxel atravesado por la línea $\mathbf{x}(\mathbf{u}, \bar{\mathbf{o}})$, la transparencia en el intervalo $[s_{i-1}, s_i]$ se puede aproximar mediante:

$$t_i = e^{-\int_{s_{i-1}}^{s_i} \kappa(t) dt} \approx e^{-\kappa_i \Delta s}, \quad (\text{A.15})$$

donde $\Delta s = s_i - s_{i-1}$. Ahora bien, si se tiene en cuenta que no todos los rayos cruzan perpendicularmente los vóxeles, puede ser usada la distancia Δ que recorre el rayo al interior del vóxel para reescribir la anterior expresión como sigue:

$$t_i = (e^{-\kappa_i \Delta})^{\Delta s / \Delta}. \quad (\text{A.16})$$

Luego, si se relaciona el término entre paréntesis con la opacidad del vóxel actual, entonces la transparencia del intervalo actual en el rayo se corrige con la expresión:

$$t_i = \left(1 - \alpha_{in}^{(i)}\right)^{\Delta s / \Delta}. \quad (\text{A.17})$$

Posteriormente, en términos de corrección a la opacidad, se tiene:

$$\alpha_{in}^{i+1} = 1 - \left(1 - \alpha_{in}^{(i)}\right)^{\Delta s / \Delta}. \quad (\text{A.18})$$

Finalmente se tiene la expresión de acumulación que se usa en este trabajo,

considerando integración *front-back*

$$\alpha_{in}^{(i+1)} = 1 - \left(1 - \alpha_{in}^{(i)}\right)^{\Delta s / \Delta}, \quad (\text{A.19})$$

$$\bar{c}_{out}^{(i+1)} = \bar{c}_{out}^{(i)} + (1 - \alpha_{out}^{(i)})\bar{c}_{in}^{(i)}\alpha_{in}^{(i)}, \quad (\text{A.20})$$

y

$$\alpha_{out}^{(i+1)} = \alpha_{out}^{(i)} + (1 - \alpha_{out}^{(i)})\alpha_{in}^{(i)}, \quad (\text{A.21})$$

con la inicialización $\bar{c}_{out}^{(i)} = (0, 0, 0)^T$ y $\alpha_{out} = 0$. Este proceso iterativo se repite hasta que el rayo sale del volumen o $\alpha_{out} = 1$.

Apéndice B

Ejemplo archivo de salida - Experimento varios espacios de color

```
adjacency FACE. -M
#STATISTICS
Th          Dice          Sensibility          Specificity

rgb
0.7         0.157849744         0.369138032         0.933682323
0.7         0.0497727692        0.0255902354        0.99998951
0.7         0.281529278         0.168385729         0.999807179
0.7         0.14961718          0.807169557         0.89546001
0.7         0.467506438         0.793005288         0.806138396
elapsed time compute segment: 654.354719120

iii2i3
0.7         0.276280433         0.550530255         0.95118767
0.7         0.677004337         0.661339641         0.998862147
0.7         0.508097529         0.436034799         0.998058498
0.7         0.836848199         0.787056386         0.998906672
0.7         0.612342358         0.693570137         0.930704653
elapsed time compute segment: 494.936863562

hsi
0.7         0.0365268588         0.0186032578         0.99999994
0.7         0.00403037481        0.00201926962         1
0.7         0.00262653735        0.00131503365        0.999999821
0.7         0.827441931          0.717620373          0.999802947
0.7         0.54078263           0.891297698          0.829707026
elapsed time compute segment: 531.249562205

hsl
```

68 APÉNDICE B. EJEMPLO ARCHIVO DE SALIDA - EXPERIMENTO VARIOS ESPACIOS DE COLOR

0.7	0.0672803521	0.0348218381	0.99999392
0.7	0.00354503281	0.00177567522	1
0.7	0.0031669084	0.00158598844	0.999999881
0.7	0.832567453	0.726978183	0.999774516
0.7	0.532675922	0.887653589	0.824843407

elapsed time compute segment: 547.843788032

hsv

0.7	0.0417942293	0.0213432331	0.999999881
0.7	0.00296998024	0.00148720806	1
0.7	0.00315244636	0.00157876289	0.999999762
0.7	0.836985648	0.736480653	0.999728143
0.7	0.516195774	0.892639339	0.81020999

elapsed time compute segment: 557.893716025

YCbCr

0.7	0.289805233	0.70272845	0.936910272
0.7	0.602506876	0.461438358	0.999726474
0.7	0.688211143	0.691733003	0.997793972
0.7	0.447557181	0.828485966	0.978192985
0.7	0.735541463	0.762202322	0.962391973

elapsed time compute segment: 530.343619755

YUV

0.7	0.294501513	0.715657055	0.936959684
0.7	0.56993264	0.436123759	0.999632955
0.7	0.673823833	0.741249084	0.996821702
0.7	0.41894415	0.838135362	0.974826455
0.7	0.734992325	0.756823421	0.963326335

elapsed time compute segment: 535.582111282

CIELab

0.7	0.384887993	0.727465391	0.95884794
0.7	0.727408946	0.619274735	0.999675393
0.7	0.699704051	0.668864429	0.998317003
0.7	0.49096325	0.79306376	0.983269393
0.7	0.736952662	0.691436768	0.977572858

elapsed time compute segment: 555.180015703

CIELuv

0.7	0.237668902	0.741836667	0.909768522
0.7	0.651160419	0.526721656	0.999645591
0.7	0.667969406	0.625486374	0.998287022
0.7	0.515271604	0.824653864	0.983983815
0.7	0.716540456	0.759001136	0.956425905

elapsed time compute segment: 591.912132777

GrayScale

0.7	0.176749378	0.104044467	0.998531163
0.7	0.258714586	0.197920471	0.998707533
0.7	0.122777514	0.0662105009	0.999914587
0.7	0.0806755722	0.593796015	0.847230136

0.7 0.396025985 0.693161607 0.780935407
 elapsed time compute segment: 668.077026950

=====

adjacency FACE-EDGE. -M

rgb

0.7 0.297305763 0.345652908 0.980361342
 0.7 0.0445396379 0.0227953102 0.999996901
 0.7 0.540313482 0.490453362 0.997749031
 0.7 0.134871855 0.808189154 0.881567359
 0.7 0.462271869 0.889456332 0.762595713
 elapsed time compute segment: 1492.276203656

iii2i3

0.7 0.358706236 0.510543287 0.973215103
 0.7 0.61464721 0.497733921 0.999525845
 0.7 0.641605198 0.678355038 0.99697876
 0.7 0.849445224 0.829617858 0.998560369
 0.7 0.554420352 0.82021451 0.86199826
 elapsed time compute segment: 1399.261888107

hsi

0.7 0.433754861 0.527509153 0.981860936
 0.7 0.523212552 0.550978541 0.997839451
 0.7 0.00841750856 0.00422689365 0.999999404
 0.7 0.848449528 0.755150914 0.999709964
 0.7 0.556542814 0.89530319 0.839761555
 elapsed time compute segment: 1330.771870277

hsl

0.7 0.422011286 0.616839647 0.973807395
 0.7 0.498290449 0.626864612 0.996539474
 0.7 0.0166916698 0.00842127297 0.999995708
 0.7 0.851688921 0.762707114 0.999670208
 0.7 0.563704073 0.891506433 0.845888019
 elapsed time compute segment: 1357.022247547

hsv

0.7 0.443684071 0.574081123 0.979677081
 0.7 0.561580598 0.653031766 0.997382164
 0.7 0.0225166697 0.0113945501 0.999995112
 0.7 0.854768515 0.772326291 0.999595284
 0.7 0.53573209 0.896402776 0.824249387
 elapsed time compute segment: 1391.064463728

YCbCr

0.7 0.216496229 0.622467518 0.9172436
 0.7 0.415441543 0.270909041 0.99987042
 0.7 0.651480496 0.74335891 0.996268809
 0.7 0.43173936 0.871388733 0.974800706

70 APÉNDICE B. EJEMPLO ARCHIVO DE SALIDA - EXPERIMENTO VARIOS ESPACIOS DE COLOR

0.7 0.653752089 0.903499305 0.895699918
elapsed time compute segment: 1510.201542610

YUV

0.7 0.218498871 0.615685225 0.919408858
0.7 0.41514045 0.27113983 0.999863327
0.7 0.687866688 0.797567904 0.996388674
0.7 0.427793443 0.87192446 0.974344492
0.7 0.653033793 0.90444392 0.895095289
elapsed time compute segment: 1492.233639794

CIELab

0.7 0.352898806 0.674169242 0.956965268
0.7 0.486360013 0.33377564 0.999849141
0.7 0.682008803 0.747423232 0.996921897
0.7 0.306749403 0.850610256 0.956993461
0.7 0.69217056 0.853768706 0.925683081
elapsed time compute segment: 1470.999405352

CIELuv

0.7 0.247221231 0.684457779 0.92276001
0.7 0.408561766 0.263684571 0.9998945
0.7 0.662163556 0.722123265 0.996820986
0.7 0.494039118 0.860752165 0.981102228
0.7 0.587379456 0.898021162 0.8594414
elapsed time compute segment: 1568.900236535

GrayScale

0.7 0.26381427 0.183391228 0.995851755
0.7 0.28192243 0.169740438 0.999866009
0.7 0.379279226 0.472805172 0.992932618
0.7 0.085873805 0.626645505 0.849079072
0.7 0.449549645 0.830512643 0.774035633
elapsed time compute segment: 1466.088193938

Referencias

- [1] D. E. P. Hoy, “On the use of color imaging in experimental applications,” *Experimental Techniques*, vol. 21, no. 4, pp. 17–19, 1997.
- [2] C. L. Ceja, “Segmentación difusa como generador de funciones de transferencia para visualización directa de volúmenes,” Tesis de maestría, Universidad Autónoma Nacional de México, 2012.
- [3] C. G. A. Hoelen, F. F. M. de Mul, and R. Pongers, “Three-dimensional photoacoustic imaging of blood vessels in tissue,” *Optical Letters*, vol. 23, no. 8, pp. 648–650, 1998.
- [4] X. Wang, Y. Pang, and G. Ku, “Noninvasive laser-induced photoacoustic tomography for structural and functional in vivo imaging of the brain,” *Nature Biotechnology*, vol. 21, no. 7, pp. 803–806, 2003.
- [5] A. De La Zerda, C. Zavaleta, S. Keren, S. Vaithilingam, S. Bodapati, Z. Liu, J. Levi, B. R. Smith, T.-J. Ma, O. Oralkan, Z. Cheng, X. Chen, H. Dai, B. T. Khuri-Yakub, and S. S. Gambhir, “Carbon nanotubes as photoacoustic molecular imaging agents in living mice,” *Nature Nanotechnology*, vol. 1, no. 3, pp. 557–562, 2008.
- [6] C. S. Tainter and B. A. G., “Selenium and the photophone,” *Nature*, vol. 1, no. 22, pp. 500–503, 1880.
- [7] A. Rosenthal, V. Ntziachristos, and D. Razansky, “Acoustic inversion in optoacoustic tomography: a review,” *Current Medical Imaging Reviews*, vol. 1, no. 9, pp. 318–336, 2013.

- [8] C. Lutzweiler and D. Razansky, “Optoacoustic imaging and tomography: reconstruction approaches and outstanding challenges in image performance and quantification,” *Sensors*, vol. 1, no. 13, pp. 7345–7384, 2013.
- [9] N. Baddour, “Theory and analysis of frequency-domain photoacoustic tomography,” *The Journal of the Acoustical Society of America*, vol. 123, no. 5, pp. 2577–2590, 2008.
- [10] V. Ntziachristos, J. Ripoll, L. H. V. Wang, and R. Weissleder, “Looking and listening to light: the evolution of whole-body photonic imaging,” *Nature Biotechnology*, vol. 23, no. 1, pp. 313–320, 2005.
- [11] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Pearson / Prentice Hall, 2008.
- [12] J. F. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.
- [13] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: active contour models,” *International Journal in Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.
- [14] G. T. Herman and B. M. Carvalho, *Handbook of biomedical image analysis: Segmentation models, part B*, vol. II, ch. 12. Kluwer Academic / Plenum Publishers, New York, 2005.
- [15] A. Rosenfeld, “Fuzzy digital topology,” *Information Control*, vol. 40, p. 76–87, 1979.
- [16] J. K. Udupa and S. Samarasekera, “Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation,” *Graphical Models and Image Processing*, vol. 58, pp. 246–261, 1996.
- [17] G. T. Herman and B. M. Carvalho, “Multiseeded segmentation using fuzzy connectedness,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 460–474, 2001.
- [18] G. T. Herman, *Geometry of digital spaces*. Applied and Numerical Harmonic Analysis, Birkhäuser Basel, 1 ed., 1996.

- [19] B. M. Carvalho, G. T. Herman, and Y. K. T., “Simultaneous fuzzy segmentation of multiple objects,” *Discrete Applied Mathematics*, 2005.
- [20] B. M. Carvalho, E. Garduño, T. S. Santos, L. M. Oliveira, and J. F. Silva Neto, “Fuzzy segmentation of video shots using hybrid color spaces and motion information,” *Pattern Analysis and Applications*, vol. 17, pp. 249–264, 2014.
- [21] B. M. Carvalho, E. Garduño, and I. O. Santos, “Skew divergence-based fuzzy segmentation of rock samples,” in *2nd International Conference on Mathematical Modeling in Physical Sciences*, (Prague-Czech Republic), September 1-5 (2013).
- [22] E. Garduño, G. T. Herman, and R. Davidi, “Reconstruction from a few projections by l_1 -minimization of the Haar transform,” *Inverse problems*, vol. 27, no. 5, 2011.
- [23] T. Carron and P. Lambert, “Color edge detection using jointly hue, saturation and intensity,” in *IEEE International conference on image processing*, pp. 977–1081, 1994.
- [24] J. Gauch and C. W. Hsia, “A comparison of three color image segmentation algorithm in four color spaces,” in *SPIE. Visual communications and image processing*, vol. 1818, pp. 1168–1181, 1992.
- [25] M. e. Pietikainen, “Accurate color discrimination with classification based on feature distributions,” in *International conference on Pattern Recognition*, pp. 833–838, 1996.
- [26] E. Littmann and H. Ritter, “Adaptive color segmentation - a comparison of neural and statistical methods,” *IEEE Trans. Neural Network*, vol. 8, no. 1, 1997.
- [27] C. A. Poynton, *Digital video and HDTV - Algorithms and interfaces*. San Francisco: Elsevier Science, 2003.
- [28] H. D. Cheng, X. H. Jiang, and Y. Sun, “Color image segmentation: advances and prospects,” *Pattern recognition*, vol. 34, no. 1, pp. 2259–2280, 2001.
- [29] C. A. Poynton, *A technical introduction to digital video*. John Wiley & Sons, Inc., 1996.
- [30] Y. Ohta, T. Kanade, and T. Sakai, “Color information for region segmentation,” *Computer Graphics and Image Processing*, vol. 13, no. 1, pp. 222–241, 1980.

- [31] T. L. Huntsberger, C. L. Jacobs, and R. L. Cannon, "Iterative fuzzy image segmentation," *Pattern recognition*, vol. 18, no. 2, pp. 131–138, 1985.
- [32] W. S. Kim and R. H. Park, "Color image palette construction based on the HSI color system for minimizing the reconstruction error," in *IEEE International conference on image processing*, pp. 1041–1044, 1996.
- [33] P. W. M. Tsang and W. H. Tsang, "Edge detection on object color," in *IEEE International conference on image processing*, pp. 1049–1052, 1996.
- [34] E. Tepichin, J. G. Suarez-Romero, and G. Ramirez, "Hue, brightness, and saturation manipulation of diffractive colors," *Optical Engineering*, vol. 34, no. 10, pp. 2886–2890, 1995.
- [35] K. M. Kim, C. S. Lee, and Y. H. Ha, "Color image quantization using weighted distortion measure of HVS color activity," in *IEEE International Conference on Pattern Recognition*, pp. 1035–1039, 1996.
- [36] CIE, "Colorimetry," Tech. Rep. 15.2, Central Bureau of the CIE, Vienna, 1986.
- [37] N. Vandenbroucke, L. Macaire, and J. G. Posaire, "Color pixel classification in an hybrid color space," in *IEEE International conference on Image Processing*, pp. 176–180, 1998.
- [38] N. Otsu, "A threshold selection method from grey level histograms," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 1, pp. 62–66, Sept. 1979.
- [39] A. Kashanipour, N. Milani, A. Kashanipour, and H. Eghrary, "Robust color classification using fuzzy rule-based particle swarm optimization," in *IEEE Congress on Image and Signal Processing*, vol. 2, pp. 110–114, May 2008.
- [40] T. F. Cootes, C. J. Taylor, and D. H. Cooper, "Active shape models: Their training and application," *Computer Vision & Image Understanding*, vol. 65, pp. 38–59, 1995.
- [41] S. Beucher and C. D. M. Mathmatique, "The watershed transformation applied to image segmentation," *Scanning Microscopy*, p. 299–314, 1991.
- [42] J. Smolka, "Watershed based region growing," *Annales UMCS Informatica AI*, no. 3, pp. 169–178, 2005.

- [43] V. Vezhnevets and V. Konouchine, “GrowCut: Interactive multi-label ND image segmentation by cellular automata,” 2005.
- [44] E. Garduño, M. Wong-Barnum, N. Volkman, and M. H. Ellisman, “Segmentation of electron tomographic data sets using fuzzy set theory principles,” *Journal of Structural Biology*, vol. 162, no. 3, pp. 368–379, 2008.
- [45] B. M. Carvalho, C. J. Gau, G. T. Herman, and Y. K. T., “Algorithms for fuzzy segmentation,” *Pattern Analysis and Applications*, 1999.
- [46] J. K. Udupa and S. P. K., “Fuzzy connectedness and image segmentation,” *Proceedings of the IEEE*, vol. 91, p. 1649–1669, 2003.
- [47] K. C. Ciesielski and J. K. Udupa, “Affinity functions in fuzzy connectedness based image segmentation (i): Equivalence of affinities,” *Computer Vision and Image Understanding*, vol. 114, pp. 146–154, 2010.
- [48] K. C. Ciesielski and J. K. Udupa, “Affinity functions in fuzzy connectedness based image segmentation (ii): Defining and recognizing truly novel affinities,” *Computer Vision and Image Understanding*, vol. 114, pp. 155–166, 2010.
- [49] A. S. Pednekar and I. A. Kakadiaris, “Image segmentation based on fuzzy connectedness using dynamic weights,” *IEEE Transactions on Image Processing*, vol. 15, pp. 1555–1562, 2006.
- [50] B. M. Carvalho, L. M. Oliveira, and G. S. Andrade, “Fuzzy segmentation of color video shots,” in *Discrete geometry for computer imagery, proceedings*, vol. 4245, pp. 494–505, Springer, 2006.
- [51] B. Dogdas, D. Stout, and R. M. Chatziioannou, A. F. and Leahy, “Digimouse: a 3D whole body mouse atlas from CT and cryosection data,” *Physics in Medicine and Biology*, vol. 52, no. 1, pp. 577–587, 2006.
- [52] “Digimouse 3D - Biomedical Imaging Group.” http://neuroimage.usc.edu/neuro/Digimouse_Download. Accedido: 31-03-2017.
- [53] “Qt version 4.8 - Documentación.” <http://doc.qt.io/qt-4.8/>. Accedido: 24-04-2017.

- [54] A. Zijdenbos, B. Dawant, R. Margolin, and A. Palmer, “Morphometric analysis of white matter lesions in MR images: method and validation,” *IEEE Transactions on Medical Imaging*, vol. 13, no. 4, pp. 716–724, 1994.
- [55] H.-H. Chang, A. H. Zhuang, D. J. Valentino, and W.-C. Chu, “Performance measure characterization for evaluating neuroimage segmentation algorithms,” *Neuroimage*, vol. 47, no. 1, pp. 122–135, 2009.
- [56] T. Fawcet, “Roc graphs: notes and practical considerations for researchers,” tech. rep., HP laboratories, Palo Alto (USA), 2004.
- [57] M. Levoy, “Display of surfaces from volume data,” *IEEE Computer Graphics and Applications*, vol. 8, no. 1, pp. 29–37, 1988.
- [58] E. Cerezo, F. Pérez, X. Pueyo, F. J. Seron, and F. X. Sillion, “A survey on participating media rendering techniques,” *The Visual Computer*, vol. 21, no. 1, pp. 303–328, 2005.
- [59] K. D. Moreland, *Fast high accuracy volume rendering*. Tesis de doctorado, University of New Mexico, Albuquerque, New Mexico-USA, 2004.