



UNIVERSIDAD DEL BÍO-BÍO

Facultad de Ingeniería
Departamento de Ingeniería Eléctrica y Electrónica
Ingeniería Civil en Automatización

Profesor Guía: Dr. Cristhian Aguilera Carrasco
Profesor Corrector: Dr. Cristian Durán Faúndez

SEGMENTACIÓN DE OBJETOS EN VOLÚMENES DE DATOS POR MEDIO DE TÉCNICAS DE CONTORNOS DEFORMABLES

Trabajo de Titulación presentado en conformidad a los requisitos
para obtener el título de Ingeniero Civil en Automatización

Fecha de Presentación
4 de Mayo de 2010

Nombre del Alumno
Arnoldo S. Norambuena Aguayo

A mi familia, amigos, profesores y a Dios,
por su ayuda, apoyo incondicional y consejos
durante la carrera.

Arnoldo Norambuena A.

INDICE

Resumen	01
Introducción	02
Capítulo 1. Volúmenes de datos	04
Introducción.....	04
1.1. Tecnologías no invasivas.....	05
1.1.1. Rayos X.....	05
1.1.2. Resonancia nuclear magnética MRI.....	08
1.1.3. Ultrasonidos.....	09
1.1.4. Microondas.....	10
1.2. Formato de información.....	11
1.2.1. DICOM.....	11
Capítulo 2. Segmentación de imágenes	12
Introducción.....	12
2.1. Definición y tipos de imágenes.....	13
2.1.1. Imagen 2D.....	13
2.1.2. Imagen binaria.....	16
2.1.3. Imagen 3D.....	16
2.1.4. Imagen volumétrica.....	16
2.2. Segmentación en imágenes 2D.....	17
2.2.1. Método umbralización.....	17
2.2.2. Método Watershed.....	19
2.2.3. Método Correlación.....	19
2.3. Segmentación en volúmenes de datos.....	21
Capítulo 3. Contornos deformables	22
Introducción.....	22
3.1. Bi-dimensional.....	23
3.2. Tri-dimensional.....	27
Capítulo 4. Desarrollo e implementación	29
Introducción.....	29
4.1. Definición de superficie.....	30
4.1.1. Condiciones de contorno.....	31
4.2. Algoritmo superficie deformable.....	32
4.2.1. Gradiente volumen de datos.....	33
4.2.2. Método Exhaustivo.....	35
4.2.3. Método Simulated Annealing.....	37
4.2.4. Método Simulated Annealing Threshold Accepting.....	40
4.3. Librería de visualización de volúmenes de datos VTK.....	42
4.3.1. Orientación a objetos.....	42
4.3.2. Actores objetos de una escena 3D.....	43
4.3.3. Renderizado.....	44
4.3.3.1. Ray Casting.....	45
4.3.4. Función de Opacidad.....	46

4.3.5. Función de transferencia.....	47
Capítulo 5. Ensayos y Pruebas.....	48
Introducción.....	48
5.1. Ensayos con volumen de prueba.....	49
5.2. Prueba con volumen de datos área médica.....	54
5.2.1. Segmentación de rodilla.	55
5.3. Prueba con volumen de datos probeta madera.	63
5.3.1. Segmentación de nudo en probeta.....	66
5.4. Comparación Técnicas.....	70
Conclusiones y comentarios.....	73
Bibliografía.....	75
Anexos.....	78

RESUMEN

La segmentación de objetos en volúmenes de datos consiste en aislar y ubicar un objeto desde datos provenientes de tomografías computarizadas, por ultrasonidos, resonancia nuclear magnética o cualquier otro método de inspección no invasivo, donde se posee información interna de las estructuras por medio de datos en las tres dimensiones del espacio.

Como objetivo en esta tesis se plantea la utilización de contornos deformables, o en este caso, superficie deformable para la segmentación de objetos al interior de los volúmenes de datos. Esta técnica ha sido estudiada en diversas universidades y con aplicaciones directas en el área médica para la cubicación de tumores, análisis de huesos, por mencionar algunas. Como objetivos en estas investigaciones se buscan segmentaciones más exactas y tiempos de procesamientos menores a través de combinaciones de técnicas como lo son las redes neuronales, paralelización de programas, algoritmos genéticos u otras técnicas metaheurísticas al procesamiento de imágenes.

Para lograr los objetivos se utilizó la librería de código libre VTK, la cual provee la lectura y visualización de los datos. La implementación de la superficie deformable se obtuvo a través de la adición de una dimensión a los contornos deformables los cuales son moldeados gracias a fuerzas que se definen de la estructura en deformación (contorno o superficie) y de la imagen a segmentar, logrando ubicarlos en el borde del objeto de interés.

Los resultados fueron satisfactorios en tres tipos de volúmenes de datos, a los que se le aplicó tres técnicas de búsqueda (Greedy, Simulated Annealing, Simulated Annealing Threshold Accepting) para la minimización de un funcional de energía que rige la estructura deformable.

INTRODUCCIÓN.

En diversas áreas tales como medicina, mecánica e industria de la madera, se torna cada vez más esencial, obtener información interna de objetos y órganos para el análisis y toma de decisiones correctas. Métodos con técnicas no invasivas como rayos X, Rayos Gamma, microondas y ultrasonidos se utilizan para este fin, tanto en la localización y cálculo de radiación a aplicar a un tumor, como la detección de defectos y singularidades en la industria. Ejemplo de esto lo constituye la detección de fisuras (grietas) en distintos materiales o nudos en madera, lo cual implica visualizar y procesar grandes volúmenes de datos.

La visualización de estos datos tiene un amplio espectro de investigación y cada día lo vemos más cerca, a través de animaciones 3D en el área del entretenimiento. En cuanto al procesamiento encontramos problemas de tiempo de cómputo y aislación de regiones de interés. Algunas soluciones que se investigan en la actualidad son el procesamiento paralelo (Pan et al, 2008), con el uso de tarjetas gráficas para aminorar estos tiempos, e implementación de técnicas avanzadas de segmentación, que se derivan de la visión artificial, con líneas investigativas de gran actividad y avances, tanto en la realización de códigos de programación mas óptimos para disminuir los tiempos de computo, como también la mejora y adaptación de las técnicas de segmentación aplicadas a volúmenes de datos.

Este trabajo pretende abordar la problemática de segmentación de regiones de interés a través de una técnica de contornos deformables ampliamente utilizada en el procesamiento y análisis de imágenes en dos dimensiones (Rangayyan et al, 2008; Shen et al, 2006; Xiaoping et al, 2008; Yim y Foran, 2003), y en donde resulta posible ampliar a tres dimensiones sin mayores inconvenientes. En la visualización de los resultados se utilizará código libre. Cabe destacar que el trabajo se implementará como un procesamiento posterior a la captura de datos.

Objetivos generales.

- Estudio de técnicas de segmentación actuales en volúmenes de datos.
- Desarrollo e implementación de un algoritmo de segmentación de regiones de interés con la utilización de contornos deformables.

Objetivos específicos.

- Implementación de Simulated Annealing y Simulated Annealing con Threshold Accepting para la búsqueda de mínimos de energía.
- Segmentación de objetos en volúmenes de datos reales.

Para el desarrollo del tema se analizarán en el primer capítulo las fuentes de volúmenes de datos y las tecnologías presentes en la adquisición de estos. Principalmente los métodos de inspección no invasiva, para luego dar paso al capítulo número dos en el cual se revisarán a groso modo técnicas de segmentación de objetos en imágenes de dos dimensiones y volúmenes de datos, además de dar un vistazo a las definiciones de imágenes tanto bidimensionales como tridimensionales necesarias para la mejor comprensión de los capítulos posteriores.

En el tercer capítulo se abordará la técnica de contornos deformables para la segmentación de objetos, tanto en su definición 2D como en 3 dimensiones, viendo las ecuaciones que rigen esta técnica, como también sus ventajas y desventajas, para luego aplicarlo en volúmenes de datos, pasando luego al cuarto capítulo consistente en el desarrollo e implementación del algoritmo junto a definiciones necesarias tanto de la superficie que se deformará para la segmentación, como de la librería VTK utilizada para la visualización de los objetos y sus atributos.

Finalmente en el capítulo cinco se realizan experimentos con volúmenes de datos en una estructura simple virtual, como también reales, en el área médica y madera, para dar paso a conclusiones y comentarios hacia trabajos posteriores.

Capítulo 1

VOLUMENES DE DATOS.

INTRODUCCIÓN.

Los volúmenes de datos generalmente están asociados a técnicas no invasivas, para la adquisición de datos internos del cuerpo humano encontrando tomografías computarizadas, ultrasonidos, microondas, resonancias nucleares magnéticas o tomografías por emisión de positrones. A su vez se pueden obtener volúmenes de datos de forma invasiva como es el caso del “The Visible Human Project”, en el cual se realizaron cortes al cuerpo de una mujer y un hombre cada 1mm, tomando fotografías de alta resolución para fines de investigación y educación.

En este capítulo se mencionan a grandes rasgos técnicas no invasivas, como rayos X, tomografías computarizadas, resonancias nucleares magnéticas, microondas y ultrasonidos, para finalmente dar una pincelada al estándar de imágenes medicas DICOM.

1.1. TECNOLOGÍAS NO INVASIVAS.

Las tecnologías no invasivas cada día se van desarrollando y perfeccionando, desde aplicaciones visuales con cámaras y tarjetas adquisidoras de video, campos magnéticos, ultrasonidos y radiación, obteniendo información interna o externa sin la necesidad de intervenir la sustancia o cuerpo en estudio. Esta cualidad de cero intervención provee grandes beneficios tanto a la industria que prioriza la no detención de los procesos productivos para controles de calidad o en la medicina para la supervisión y visualización de estructuras internas dejando de lado riesgosas cirugías exploratorias en algunas patologías.

Sin duda en los últimos años la adquisición de imágenes con cámaras digitales para la inspección exterior de objetos a tenido una gran masificación, implicando un abaratamiento en los costos monetarios en sistemas de inspección dentro de la industria, a su vez la medicina también ha tenido un aumento de exámenes no invasivos pero en menor medida debido a que tecnologías de toma de datos de estructuras internas como resonadores nucleares magnéticos aun son muy elevados debido a sus complejos sistemas de adquisición y procesamiento.

Técnicas de ondas electromagnéticas y rayos X están siendo estudiadas para poder llevarlas a líneas de producción industrial (Neubauer, 1997; Baradit et al, 2006; Teramoto et al, 2007; Fuch et al, 2008), lo cual nos demuestra la importancia que la comunidad científica ve en estas técnicas y los beneficios que proveen.

1.1.1. RAYOS X

Los rayos X son ondas electromagnéticas como lo son las microondas, la luz visible, ondas de radio y radiación infrarroja. Estos Rayos ionizantes son producidos por los electrones del átomo, a diferencia de los rayos gamma que se producen desde el núcleo del átomo, los cuales provocan una ionización o dicho de otra forma producen carga a la materia en la cual se aplica.

Estos Rayos fueron descubiertos Wilhelm Conrad Röntgen en el año 1895, el cual obtuvo imágenes de piezas de metal menos densas que el plomo e imágenes de extremidades humanas sobre placas fotográficas veladas.

Los rayos X poseen una frecuencia de onda que se encuentra entre los 30 y 3.000 PetaHertz con una longitud de onda que va desde los 10 hasta los 0.01 nanómetros, posicionándolos en el espectro electromagnético por debajo de los rayos gamma, por sobre la luz visible y rayos ultravioleta que principalmente provienen del sol, los que podemos visualizar en la figura 1.1.

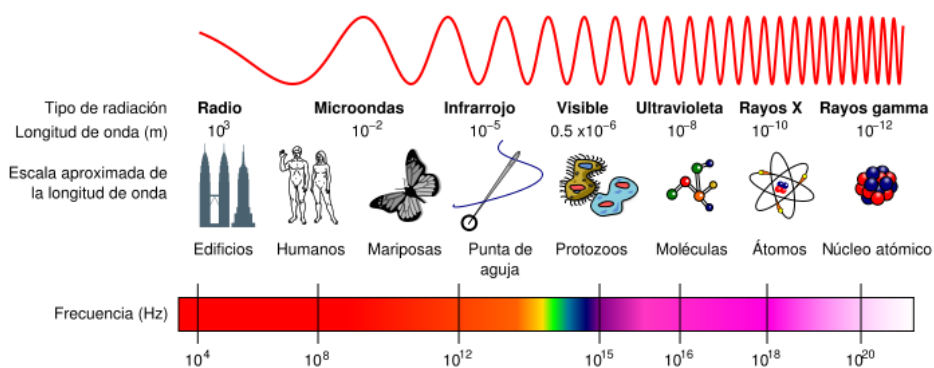


Figura N°1.1 Espectro electromagnético



Figura N° 1.2 Rayos X

La capacidad de los rayos X de atravesar estructuras del cuerpo humano impulsó una frenética carrera de investigación debido al inmenso aporte en el diagnóstico y estudio médico, naciendo la Radiología. Actualmente estos rayos son utilizados en medicina tanto para observar lesiones óseas (Fig. N° 1.2) o tejidos blandos como el pulmón a través de procedimientos con medios de contrastes como lo son angiogramas, en el cual se observan vasos circulatorios de sangre, pudiendo ser arterias o venas previa inyección de sustancias químicas en base a yodo, lo cual permite su visualización de estos en placas fotográficas o sistemas adquiredores con monitores para la visualización en tiempo real.

Pero sin lugar a dudas la mayor aplicación de estos rayos es en el desarrollo y creación de las tomografías computarizadas o CT (Computed Tomography), la cual ha logrado la comprensión y estudio del cuerpo humano a límites insospechados debido a las reconstrucciones computarizadas tridimensionales que se pueden lograr con estos cortes. Esta técnica consiste en la visualización de imágenes o cortes axiales adquiridos por medio de la aplicación de rayos X con el método de tomografía propuesto por el radiólogo Alessandro Vallebona en la década de 1930, dando a los médicos la capacidad de ver los reales volúmenes de crecimientos anormales de tejidos o la adecuada preparación para procedimientos tan delicados como lo son intervenciones en neurocirugía, para la cual el especialista tiene un modelo tridimensional del paciente a operar, planificando los pasos a desarrollar en la operación.

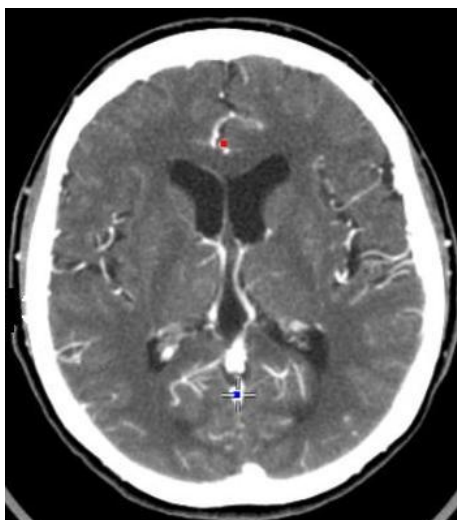


Figura N° 1.3 Tomografía computarizada (CT)

1.1.2. Resonancia nuclear magnética (MRI)

La técnica de resonancia nuclear magnética, MRI (Magnetic Resonance Imaging) consiste en la aplicación de un campo magnético, en modo de pulsos, a los átomos de un cuerpo, en donde el núcleo de estos últimos absorbe la energía para luego irradiarla a una frecuencia resonante específica, dependiente tanto del campo magnético o de que esté compuesta la materia.

El uso principal de esta técnica en la actualidad, son las imágenes del cuerpo humano al alinear los átomos de hidrogeno existentes en nuestro cuerpo debido al alto porcentaje agua que nos compone.

Por lo tanto se aplica un campo magnético de gran intensidad, lo cual hace resonar los protones del núcleo de los átomos de hidrógeno, obteniendo una magnitud pero no una posición espacial de la materia que se encuentra resonando por lo que se aplican 3 bobinas de gradiente las cuales afectan el campo magnético principal con una intensidad y frecuencia controlada, estas están dispuestas de forma ortogonal para luego decodificar esta información a través de la transformada de Fourier.



Figura N°1.4 Resonancia Nuclear magnética (MRI)

La técnica de MRI es ampliamente preferida en imagenología debido a que no es una radiación ionizante como lo es la tomografía computarizada, y provee mejores

contrastes en las imágenes de tejidos blandos como el cerebro mostrado en la figura 1.4, imagen que contrasta con la tomografía computarizada en la figura 1.2. Por otro lado, los beneficios de adquisición de datos de la resonancia provee reconstrucciones tridimensionales de mucha mejor definición ya que se trabaja con los datos entregados en 3 planos, a diferencia de la tomografía que trabaja con cortes axiales, lo cual implica que para una mejor reconstrucción se deben realizar cortes con un menor espaciado entre uno y otro.

1.1.3. Ultrasonidos

Se denomina ultrasonido a las ondas sonoras (ondas mecánicas) por sobre la frecuencia audible por el ser humano, aproximadamente unos 20 KiloHertz, mecánicas debido a que necesitan un medio físico para su transmisión el cual puede ser aire, líquido o sólido.

Las aplicaciones de los ultrasonidos van desde la medición de distancias midiendo el tiempo en recibir el eco de una onda emitida, medición de caudal en tuberías, hasta la visualización de fetos en el vientre materno (Fig. 1.5). Desarrollos en los últimos años han desencadenado máquinas de ultrasonido con reconstrucciones tridimensionales (Nelson, 2006), con la utilización de dos fuentes de ultrasonido y un gran procesamiento de las señales recibidas (Fig. 1.6).



Figura N° 1.5 Ultrasonido



Figura N° 1.6 Representación 3D ultrasonido

1.1.4. Microondas

Las microondas son ondas electromagnéticas que se encuentran por debajo de la luz visible y de la radiación infrarroja en el espectro electromagnético y por sobre las ondas de radio, con frecuencias que van desde los 100 MHz hasta los 100 GHz, y longitudes de onda entre 1 metro y un milímetro.

Utilizadas primeramente en tiempos de la 2° guerra mundial en el radar, pasando por las comunicaciones en teléfonos celulares, señales de televisión, ondas inalámbricas Wi-Fi, navegación a través del sistema de posicionamiento global GPS, electrodomésticos como el “Horno Microondas” que genera movimiento en las partículas de agua de los alimentos, calentándose estos debido a la vibración, y en medicina con la terapia de diatermia donde se aplican microondas a tejidos sensibles a temperaturas altas.

1.2. FORMATO DE INFORMACIÓN

1.2.1. Estándar DICOM

DICOM es la sigla para “Digital Imaging and Communication in Medicine”, es un estándar para las imágenes digitales provenientes de equipos de imágenes medica que deben ser guardadas, visualizadas e impresas dentro de muchas más operaciones dentro de un examen médico, tanto para diagnostico como para supervisión .

Este estándar fue emitido por el “American College of Radiology” y estandarizado por la National Electrical Manufacturers Association (NEMA) en la década de 1980, nace de la necesidad de observar las imágenes provenientes de diversos equipos y de diferentes fabricantes, además de controlar la pérdida de información que se suscita debido a la cantidad de imágenes que se deben tomar. Como ejemplo una tomografía computarizada dependiendo la necesidad de resolución entre corte y corte puede albergar alrededor de 250 imágenes, las cuales deben contener información relevante y primordial ante posibles pérdidas y así buscar la fuente de generación. Cabe destacar que este ejemplo corresponde a un examen teniendo una gran población de pacientes y médicos que solicitan este servicio, por lo que una estandarización era necesaria.

Los archivos Dicom (.dcm) contienen información concerniente al paciente como su nombre, especialista que adquiere las imágenes, tamaño de estas imágenes, resolución entre cortes, las imágenes propiamente tal, modelo y marca de la maquina que las genera por decir algunas.

Los datos de cada pixel de las imágenes son guardados en 16 bits, de los cuales 12 son de profundidad de una escala de grises.

Capítulo 2

SEGMENTACIÓN DE IMÁGENES.

INTRODUCCIÓN.

La segmentación de imágenes es un proceso ampliamente investigado, para la localización o seguimiento de objetos y patrones, con variadas técnicas que proporcionan ventajas o desventajas sobre algún aspecto, como lo pueden ser imágenes de baja calidad, ruido, formas complejas, tiempo de procesamiento.

Para poder hablar de imágenes se definirán los tipos de imágenes ya sean en dos dimensiones o volúmenes de datos, para luego mostrar tres técnicas de segmentación como lo son por umbralización, watershed y correlación.

2.1. DEFINICIÓN Y TIPOS DE IMAGENES

Definir y nombrar algunos tipos de imágenes se hace necesario ya que depende en gran medida de donde las obtengamos pudiendo ser dibujos virtuales como también de imágenes provenientes de cámaras o de un proceso previo como lo es la umbralización.

2.1.1. Imagen 2D

Una imagen se puede definir como una función $f(x,y)$, en donde x e y son coordenadas espaciales dentro de un plano y el valor resultante de la función es la intensidad en esa posición (Fig. 2.1). En el caso de ser una imagen monocromática, esta solo posee un plano de intensidades de grises, en cambio las imágenes a color dependen del modelado del color, existiendo distintas formas de modelos, ejemplo de esto son HSV y RGB, este último se basa en rojo, verde y azul, (RGB, Red, Green, Blue), donde un color es la combinación aditiva de estas 3 componente como lo muestra la figura 2.2, por lo tanto una imagen tiene 3 planos, uno para el rojo, verde y azul.

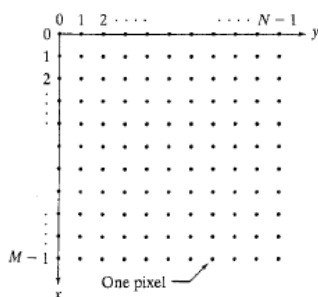


Figura N° 2.1 Representación imágenes 2D por pixeles

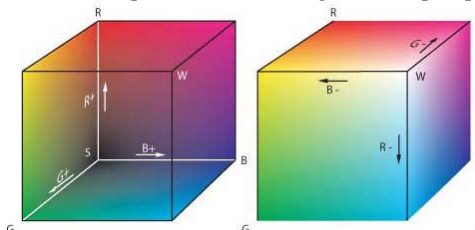


Figura N° 2.2 Representación color RGB

Estas imágenes pueden ser obtenidas a través de cámaras digitales en donde se captura los rayos de luz, como lo muestra la figura 2.3, y donde un arreglo (matriz) de sensores procesa estas intensidades y cuantifica la cantidad de luz en tonalidades rojo verde y azul por cada sensor, en el caso de adquirir una imagen a través de una cámara a color provista de sensores CCD (Charge-Coupled Device), que captura imágenes en el rango visible del espectro electromagnético que se encuentra entre los 700 y 900 nanómetros, habiendo también cámaras que capturan el rango infrarrojo o ultravioleta.

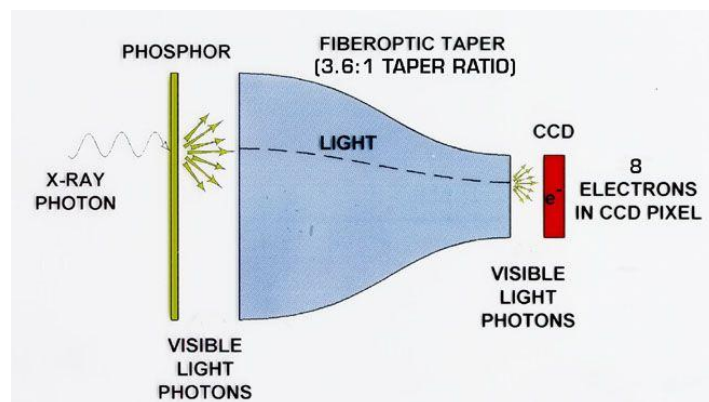


Figura N° 2.3 Funcionamiento cámara CCD

Las imágenes digitales tienen como mínimo elemento el píxel, que posee una profundidad de color, el cual digitalmente está limitado por una cantidad de bits pudiendo ser 8, 16, 32 bits.

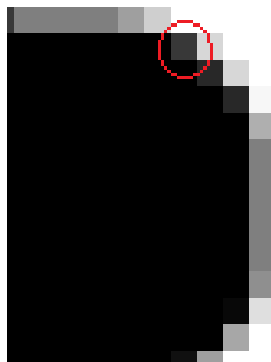


Figura N° 2.4 Pixel, encerrado por círculo.

En imágenes bidimensionales una vecindad de un pixel viene dada por 2 posibles criterios los cuales son de los 4 conectados u 8 conectados (Fig. 2.5).

Para el caso de los 4 conectados la definición matemática para un pixel $p = (x, y)$

$$V_4(p) = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\} \quad (2.1)$$

En el caso de los 8 conectados se presentan como la unión de los pixeles diagonales definidos en (2.2) (Fig. 2.5 (b)) y los que se definen en (2.3), obteniendo finalmente la vecindad de 8 conectados en (2.4).

$$V_D(p) = \{(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)\} \quad (2.2)$$

$$V_8(p) = V_4(p) \cup V_D(p) \quad (2.3)$$

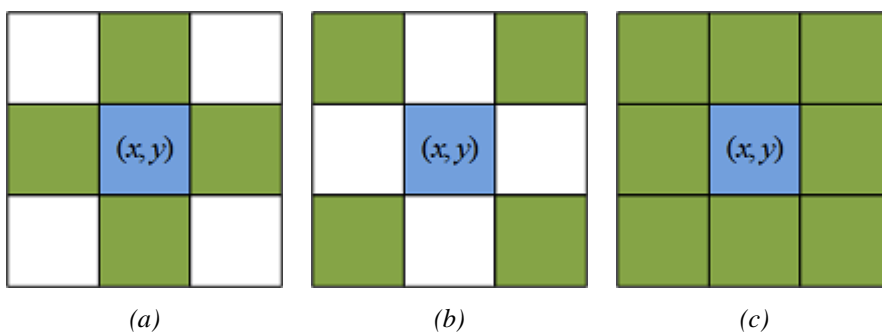


Figura N° 2.5 Vecindad, (a) 4 conectados, (b) Diagonales, (c) 8 conectados.

2.1.2. Imagen binaria

Una imagen binaria consiste en una matriz binaria en donde los valores de pixel tienen como profundidad de color sólo 2 valores, los cuales pueden ser cero o uno.

2.1.3. Imagen 3D

Una imagen en 3D es el resultado de un proceso de renderizado desde un volumen de datos o modelo en tres dimensiones, por lo que una imagen 3D, tiene 2 dimensiones simulando la tercera dimensión a través de sombras y perspectivas, concepto que se aclara en el capítulo 4.3 donde se definirán una escena 3D y el renderizado.

2.1.4. Imagen volumétrica

Una imagen volumétrica, puede definirse como una serie de imágenes 2D alineadas en un espacio tridimensional virtual, con una distancia entre imagen como lo muestra la figura 2.6, pudiendo definirla como una función $f(x,y,z)$, en donde x,y,z son coordenadas espaciales dentro del espacio, o $f(s,r,c)$ con **row** (fila), **column** (columna) y numero de imagen o corte (slice), con un valor resultante que puede ser el valor del pixel de la imagen “s”, en la columna “c” y fila “r”, o el valor dentro del voxel en la posición x,y,z , entendiéndose como voxel el mínimo elemento de volumen, similar al pixel en imágenes en 2D. Este valor de intensidad del voxel se puede definir como el promedio de las 6 caras en una resonancia nuclear magnética o el procesamiento entre 2 slices promediando el valor del pixel superior e inferior de una tomografía computarizada.

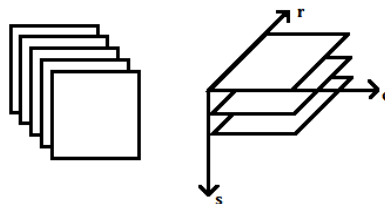


Figura N° 2.6 Imagen volumétrica desde imágenes 2D

Una vecindad de un voxel en imágenes volumétricas puede definirse como una vecindad 6, 18 o 26 definidas a continuación:

- Vecindad 6:

$$V_6(x, y, z) = \{(x \pm 1, y, z), (x, y \pm 1, z), (x, y, z \pm 1)\} \quad (2.4)$$

- Vecindad 18:

$$V_{18}(x, y, z) = \{(x, y \pm 1, z \pm 1), (x \pm 1, y, z \pm 1), (x \pm 1, y \pm 1, z)\} \cup V_6(x, y, z) \quad (2.5)$$

- Vecindad 26:

$$V_{26}(x, y, z) = \{(x \pm 1, y \pm 1, z \pm 1)\} \cup V_{18}(x, y, z) \quad (2.6)$$

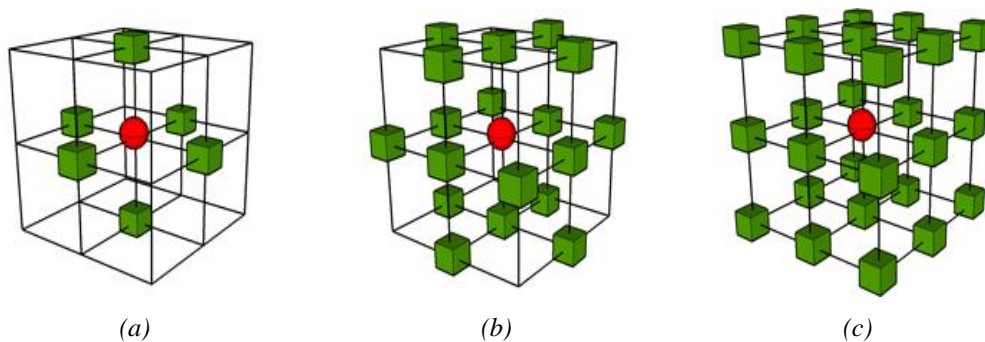


Figura N° 2.7 Vecindad volumen de datos, (a) 6 conectados, (b) 18 conectados, (c) 26 conectados.

2.2. SEGMENTACIÓN EN IMÁGENES 2D

La segmentación es un concepto fundamental en el procesamiento digital de imágenes, este proceso consiste en aislar un objeto en particular desde una escena para su posterior análisis. Técnicas de segmentación hay variadas, algunas muy ventajosas en algunas circunstancias como por ejemplo imágenes con ruido, que puede provenir de una escasa iluminación.

2.2.1. Método umbralización

Esta técnica consiste en definir un umbral para la profundidad de color de cada pixel. Si éste supera ese umbral se le da un valor 1 y si está por debajo o igual un 0, debiendo recorrer todos los pixeles de la imagen o de la región de interés que estemos segmentando, obteniendo una imagen binaria.

Este método se puede definir matemáticamente como:

$$g(x, y) = \begin{cases} 1, & f(x, y) < \text{umbral} \\ 0, & f(x, y) \geq \text{umbral} \end{cases} \quad (2.5)$$

El umbral puede ser definido automáticamente como el método Otsu (Otsu N.,1979), el cual se basa en estadísticos como la varianza para medir la dispersión de los valores, o una persona especializada, basándose en el histograma, el cual es una grafica estadística de la frecuencia de aparición de una profundidad de color en la imagen, como ejemplo podemos apreciar la figura 2.8 (b) los 3 histogramas para una imagen a color (Fig. 2.8(a)), o solo uno para el caso de imágenes en escala de grises como lo muestra la figura 2.9.



Figura N° 2.8 Histograma colores, (a) Imagen, (b) Histograma para cada canal de imagen a color

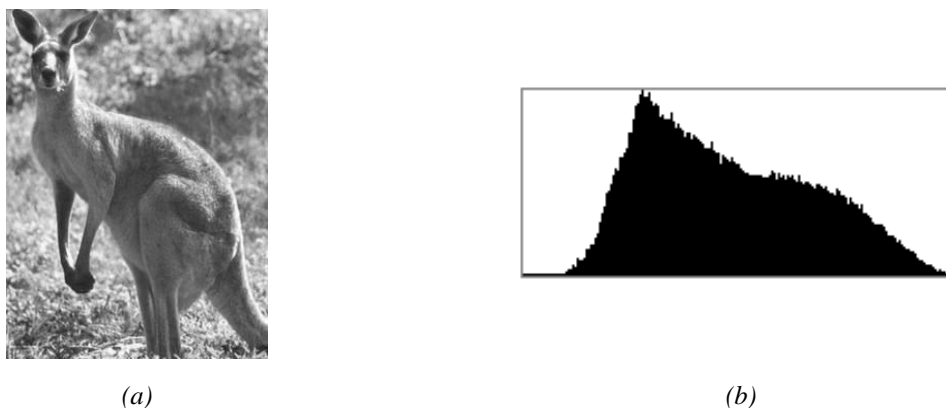


Figura N° 2.9 Histograma grises (a) Imagen, (b) Histograma imagen en escala de grises

2.2.2. Método Watershed

El presente método se basa en principios morfológicos, transformando una imagen en escala de grises a un relieve o llamada de otra forma superficie topográfica a través del modulo del gradiente de la imagen, donde el valor de la profundidad de color máximo del modulo del gradiente corresponderá a una elevación y valores menores a valles. Luego se le añade, metafóricamente, “agua” desde un mínimo global o varios mínimos locales (valles), formando lagunas que cada vez irán subiendo, debido a que cada vez se agrega más agua, hasta alcanzar las mayores elevaciones o fluyendo a otro mínimos, finalmente las lagunas definirán contornos correspondientes a la segmentación.

2.2.3. Método Correlación

Si bien el método de correlación es un método de búsqueda de patrones, se puede definir como un método de segmentación ya que encuentra un patrón dentro de una imagen, localizando y pudiendo aislar el objeto encontrado, el método se basa en el conocimiento a priori de lo que se está buscando a través de una imagen patrón (Fig. 2.11) de menor tamaño que la imagen a procesar (Fig. 2.12), para luego correlacionarlas (Fig.2.10).

La correlación se puede definir de forma que haya una invariancia al tamaño y al ángulo del patrón con respecto a la imagen a procesar, definida a continuación.

$$R(x, y) = \frac{\sum_{i=0}^{L-1} \sum_{j=0}^{K-1} (w(i, j) - \bar{w})(f(i + x, j + y) - f(x, y))}{\left[\sum_{i=0}^{L-1} \sum_{j=0}^{K-1} (w(i, j) - \bar{w}) \right]^{\frac{1}{2}} \left[\sum_{i=0}^{L-1} \sum_{j=0}^{K-1} (f(i + x, j + y) - f(x, y))^2 \right]^{\frac{1}{2}}} \quad (2.6)$$

Siendo \bar{w} : Valor promedio imagen patrón

L : Alto imagen patrón

K : Ancho imagen patrón

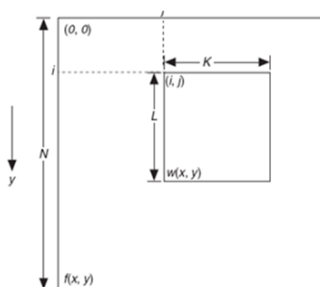


Figura N° 2.10 Esquema de correlación imágenes

Finalmente se obtiene la imagen correlacionada como muestra la figura 2.13 en donde las zonas de alta intensidad corresponden a lugares donde se encuentra un alto puntaje donde debiese estar el patrón.



Figura N° 2.11 Patrón a buscar.

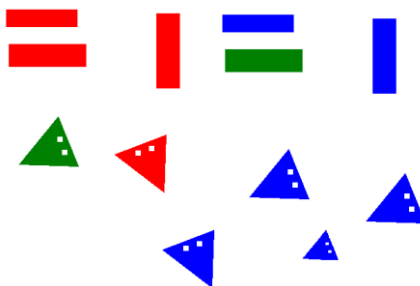


Figura N° 2.12 Imagen a correlacionar con el patrón.

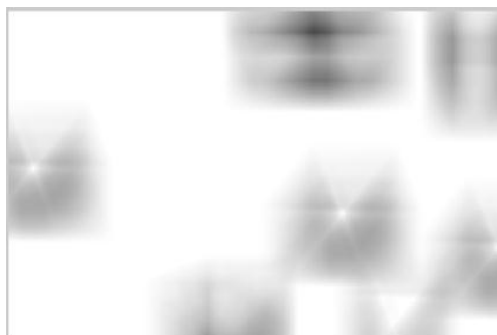


Figura N° 2.13 Resultado final correlación.

2.3. Segmentación en volúmenes de datos

Existen dos grandes caminos a seguir en la segmentación de volúmenes de datos, preferentemente de datos médicos, debido a su gran investigación durante las últimas décadas. Uno de estos caminos es tomar cortes o Slice del volumen de datos y utilizar las técnicas tradicionales de imágenes en dos dimensiones para la segmentación y luego reconstruir un modelo tridimensional de acuerdo a los datos segmentados de cada slice (imagen de un corte) (Allevatoa y Pedarréa, 2006). El otro camino es trabajar directamente en las tres dimensiones y en donde la técnica mas utilizada es la de superficies deformables o contornos deformables (Kass M. 1988), la cual presenta menos incongruencias que reconstrucciones de segmentaciones en 2 dimensiones, donde puede haber una sobre-segmentación entre cortes.

Capítulo 3

CONTORNOS DEFORMABLES.

INTRODUCCION.

Los contornos deformables en el procesamiento de imágenes fueron introducidos por Kass et al (1988), basado en las características físicas que caracterizan a los contornos (curvas paramétricas) como son la elasticidad y la curvatura.

Estas curvas son moldeadas y deformadas, a través de fuerzas internas y externas.

En el presente capítulo se darán las bases de la teoría de los contornos activos o Snakes, para luego definir una superficie deformable y así poder implementarlos en volúmenes de datos.

3.1. BI-DIMENSIONAL

Los Snakes son curvas paramétricas, que dependiendo de una condición de contorno estas pueden ser cerradas o abiertas, deformándose a través del tiempo (cada iteración). Definidas matemáticamente por puntos de control o puntos Snake $v(s) = (x(s), y(s))$, con $s \in [0,1]$ el cual es un parámetro de espacio para el caso continuo, teniendo la variable de tiempo t en las iteraciones de deformación como lo muestra la figura 3.1 con una curva discretizada.

El parámetro s se discretiza como explica (Cohen L. 1993) a través de

$$s = [0,1] = \bigcup_{i=0}^N [ih, (i+1)h] \tag{3.1}$$

Donde N es la cantidad de puntos a discretizar y $h = \frac{1}{N+1}$

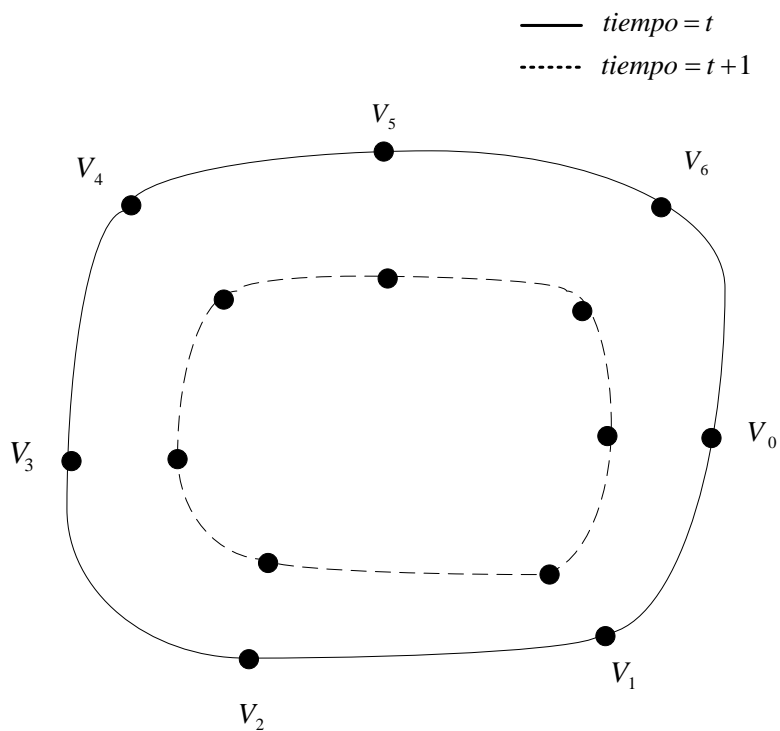


Figura N° 3.1 Evolución Snake.

Definido el Snake, se procede a definir el modelo de contornos deformables, el cual es un funcional de energía correspondiente a la suma de la energía en cada punto del Snake. Curva (Snake) que se superpone sobre una imagen, la cual influencia la deformación, debido a una adición de energía de esta (la imagen) a la curva.

$$E_{Snake} = \int_0^1 E_{elemento}(v(s))ds \quad (3.2)$$

Cada punto Snake está influenciado por energías internas (propias de la curva) y externas (de la imagen), que se describen como:

- Energía interna: Corresponde a características internas dadas por el modelo de la curva (Snake), como lo son la rigidez y suavidad, donde la rigidez es una característica de fuerza o tensión entre los puntos y la suavidad una característica de curvatura (redondez) de la curva descrita por los puntos Snake.
- Energía externa: Proviene de un procesamiento de la imagen. Típicamente se utiliza los bordes de los objetos de la imagen obtenidos a través del gradiente de la imagen ($\nabla I(v(s,r))$). Pudiendo también utilizarse la intensidad de la imagen directamente. Un ejemplo de la energía externa se aprecia en la figura 3.2 donde se encuentra la imagen y su imagen con un procesamiento de gradiente, obteniendo los bordes de los objetos.

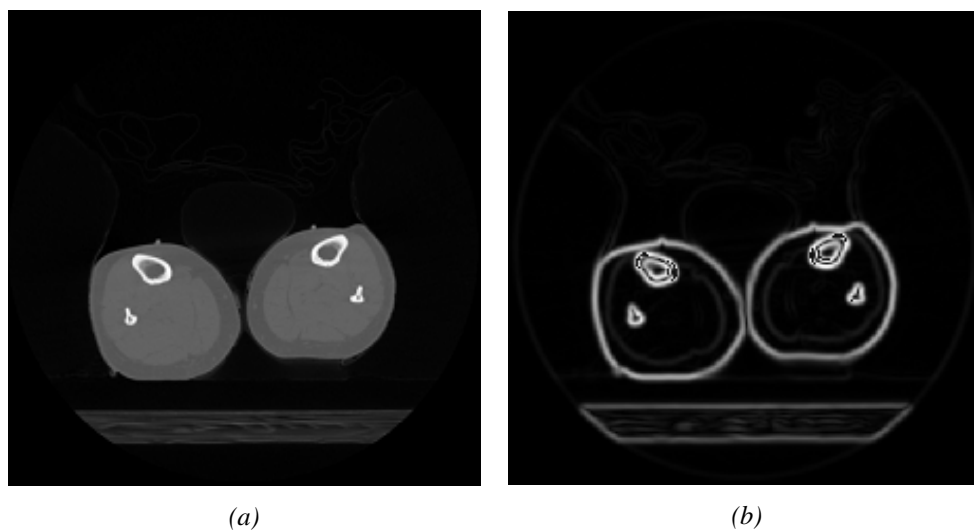


Figura N° 3.2 Energía externa del contorno, (a)Imagen, (b)Gradiente de la imagen

Reescribiendo la ecuación 3.2 en términos de las energías mencionadas (Kass et al, 1988) obtenemos:

$$E_{Snake} = \int_0^1 E_{Interna}(v(s))ds + \int_0^1 E_{Externa}(v(s))ds \quad (3.3)$$

Definidas por:

$$E_{Interna}(v(s)) = \frac{1}{2} \alpha(s) |v_s(s)|^2 + \frac{1}{2} \beta(s) |v_{ss}(s)|^2 \quad (3.4)$$

$$E_{Externa}(v(s)) = \gamma(s) |\nabla I(v(s))|^2 \quad (3.5)$$

$$\nabla I(v(s)) = \left\{ \frac{\partial(v(s))}{\partial x}, \frac{\partial(v(s))}{\partial y} \right\} \quad (3.6)$$

En donde α, β, γ corresponden a pesos que controlan los aportes dentro de la ecuación, siendo α quien controla la elasticidad de la curva dada por $v_s(s)$, que corresponde a la derivada de primer orden respecto a "s", β la responsable de la suavidad representada por $v_{ss}(s)$ siendo la derivada de segundo orden respecto a "s", finalmente γ controla el aporte de la energía de la imagen $\nabla I(v(s))$ modelada por el gradiente de la intensidad de la imagen, estos parámetros (α, β, γ) se escogen arbitrariamente dependiendo de la implementación.

Esta energía se utiliza para encontrar el o los objetos a segmentar por medio de la minimización de este funcional, el cual genera un movimiento de los puntos Snake obteniendo energías más bajas y deteniéndose la deformación en algún momento de equilibrio del funcional de energía.

Un método utilizado para la minimización es Greedy o método exhaustivo, el cual encuentra un mínimo local, esperando que la solución final sea un óptimo global, donde el mínimo local está dado por la búsqueda del mínimo de energía en una ventana cuadrada de búsqueda como la que se observa en la figura 3.3 de tamaño 3x3 pixeles, esta ventana puede ser mayor pero siempre de tamaño impar para dejar el punto Snake actual en el pixel

central. Si bien un aumento en el tamaño de la ventana de búsqueda implica un mayor rango de búsqueda también implica una mayor cantidad de cálculos al proceso. Otros métodos de búsqueda que podemos encontrar en la literatura se basan en colonias de hormigas (Li et al, 2008), bandadas de pájaros (Particle Swarm Optimization) (Tseng et al, 2009) o en la metalurgia (Simulated Annealing) (Aguilera et al, 2008-B).

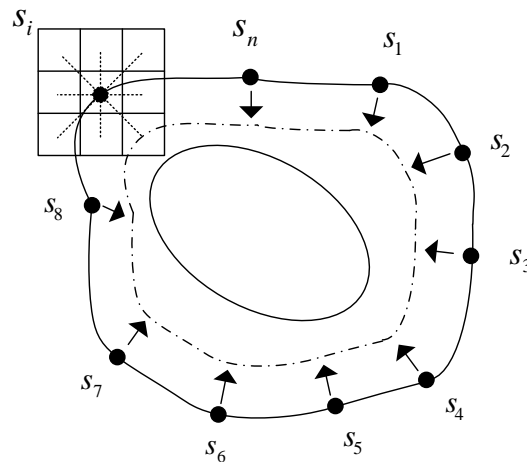


Figura N° 3.3 Evolución Snake a través de Greedy.

La ubicación de los puntos iniciales del Snake debe ser dada por una persona que supervisa o un pre-procesamiento.

- **Ventajas**

- Segmentación de contornos suaves.
- Se evita la sobre-segmentación que ocurre en otras técnicas.
- Se obtiene un modelo del objeto a segmentar.
- Buena segmentación en imágenes de baja calidad.

- **Desventaja**

- El contorno no evoluciona hacia formas muy cóncavas con el método exhaustivo, como lo muestra la figura 3.4.

- Inicialización por procesamiento previo y difícil de automatizar (Lakare, 2000).
- Puntos iniciales deben estar cerca de la solución, de otro modo el contorno convergerá a un mal resultado.

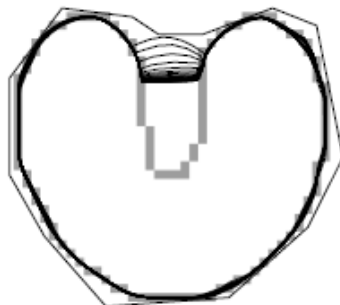


Figura N° 3.4 Evolución Snake forma cóncava.

Este funcional de energía a minimizar se ha ido modificando a través de las investigaciones con mejores resultados, debido a nuevas formas de guiar este contorno; Gradient Vector Flow (GVF), el cual añade una fuerza externa (Xu y Prince 1998; Li et al, 2008; Cheng et al, 2008; Ying et al, 2008), basándose en técnicas de evolución de curvas (Chan y Vese, 2001) o añadiendo información a priori de la forma final del contorno en el funcional (Aguilera et al, 2008-A).

3.2. TRI-DIMENSIONAL

Los contornos deformables aplicados a volúmenes de datos implican una reformulación del contorno, que ahora será una superficie.

Para el cálculo de la energía se utiliza el mismo funcional descrito por la suma de la energía interna y externa, esto quiere decir la derivada de primer y segundo orden del contorno (en este caso superficie) a deformar y el gradiente de la imagen (Cohen y Cohen 1991).

$$w_{10} \left\| \frac{\partial v}{\partial s} \right\|^2 + w_{01} \left\| \frac{\partial v}{\partial r} \right\|^2 + 2 \cdot w_{11} \left\| \frac{\partial^2 v}{\partial s \partial r} \right\|^2 + w_{20} \left\| \frac{\partial^2 v}{\partial s^2} \right\|^2 + w_{02} \left\| \frac{\partial^2 v}{\partial r^2} \right\|^2 + P(v(s, r)) \quad (3.7)$$

La superficie se define como una mapeo en 2 dimensiones a 3 dimensiones

$$v: \Omega = [0,1] \times [0,1] \rightarrow \mathbb{R}^3$$

$$(s, r) \rightarrow v(s, r) = (v_1(s, r), v_2(s, r), v_3(s, r)) \quad (3.8)$$

Con una energía

$$E(v) = \int_0^1 \int_0^1 \left[w_{10} \left\| \frac{\partial v}{\partial s} \right\|^2 + w_{01} \left\| \frac{\partial v}{\partial r} \right\|^2 + 2 \cdot w_{11} \left\| \frac{\partial^2 v}{\partial s \partial r} \right\|^2 + w_{20} \left\| \frac{\partial^2 v}{\partial s^2} \right\|^2 + w_{02} \left\| \frac{\partial^2 v}{\partial r^2} \right\|^2 + P(v(s, r)) \right] ds dr \quad (3.9)$$

Donde $P(v(s, r)) = -\|\nabla I(v(s, r))\|^2$

Los pesos de las energía interna están dados por los coeficientes w_{ij} con (w_{10}, w_{01}) determinando la elasticidad, (w_{20}, w_{02}) la rigidez y por último w_{11} la resistencia a la torsión (twist).

Capítulo 4

DESARROLLO E IMPLEMENTACIÓN.

INTRODUCCIÓN.

La implementación de la superficie deformable posee algunos detalles necesarios de explicar como lo son su definición, y las condiciones que debe cumplir para realizar los cálculos del funcional de energía, los cuales serán descritos en este capítulo. Además de ver como se ayuda a expandir la fuerza externa aplicada a la superficie, lo cual hará que puntos de la superficie sean atraídos desde posiciones lejanas al objeto a segmentar.

Posteriormente se desarrollarán tres métodos de búsqueda de nuevas posiciones para los puntos de control de la superficie deformable, con la técnica de búsqueda exhaustiva, Simulated Annealing y Simulated Annealing con Threshold Accepting.

Finalmente se explicará a grandes rasgos la librería VTK, utilizada para la visualización de los datos pasando además por los elementos que componen una escena tridimensional, como también las teorías detrás de la visualización de volúmenes de datos.

4.1. DEFINICIÓN DE SUPERFICIE.

La manera de definir la superficie adquiere una gran importancia en el resultado final de la segmentación ya que esta puede deformarse mas rápido, generar superficies más suaves o una mejor aproximación al objeto a segmentar. Entre las formas que se tiene para definir una superficie una es la utilización de superficies cuadradas como Terzopoulos y Metaxas (1991). En el cual a través de parámetros deforman las superficie tanto de forma local como global.

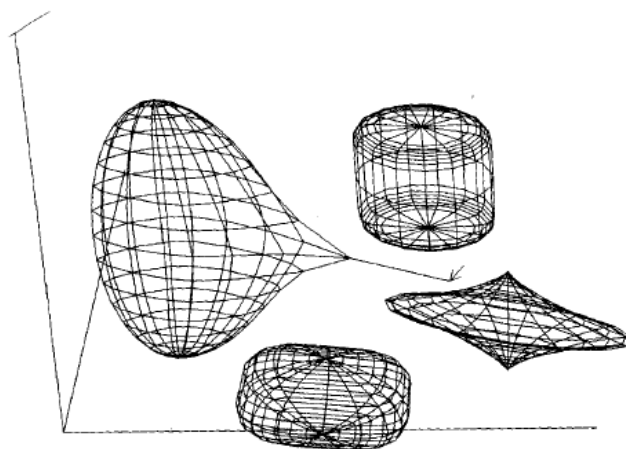


Figura N° 4.1 Superficies cuadradas.

En la implementación del algoritmo de superficies deformables se definió como una malla de puntos, que se encuentran mapeados en \mathcal{R}^2 definidos por (s,r) los cuales tienen una posición en el espacio x,y,z . Esto se puede apreciar en la figura 4.2 en donde están ubicados los puntos del Snake (superficie deformable) y su posición en el plano x,y,z , mostrado en la figura 4.3, lo cual matemáticamente se expresa como sigue:

$$v(s, r) = (x(s, r), y(s, r), z(s, r)) \quad (4.1)$$

Donde s corresponde a los puntos del Snake del contorno a una altura r , siguiendo la definición adoptada por (Cohen L. 1993)

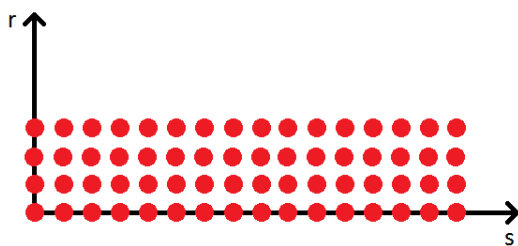


Figura N° 4.2 Puntos Snake en plano rs.

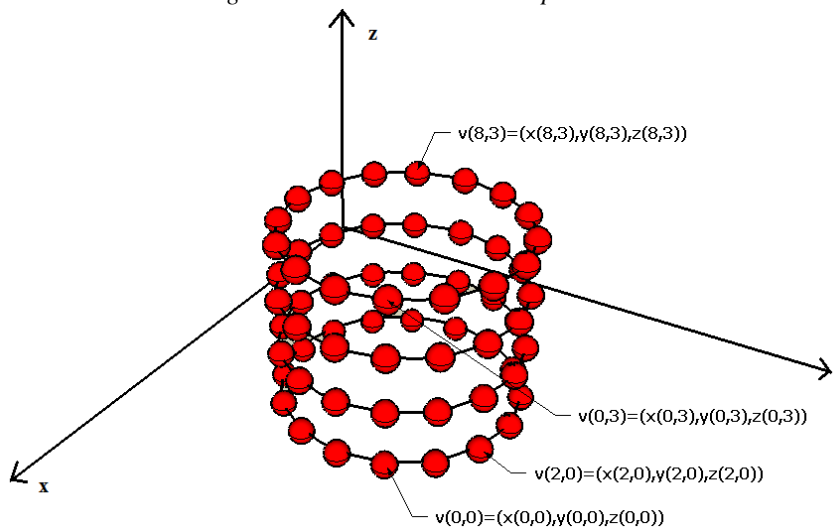


Figura N° 4.3 Ejemplo de puntos ubicados en plano xyz.

4.1.1. Condiciones de contorno.

Como condición de contorno se puede pedir que los $\text{Max}\{s\}$ de cada r tenga como vecino siguiente $s = 0$, formando un cilindro como se muestra en la figura 4.3. Otra forma de abordar una condición de contorno es pedir que cada $\text{Max}\{r\}$ de cada s tenga como vecino siguiente $r = 0$ y que a su vez los $\text{Max}\{s\}$ de cada r tenga como vecino siguiente $s = 0$, lo cual formaría una superficie toroidal, como lo muestra la figura 4.4.

Para la implementación se optó por una condición de contorno que formará un cilindro.

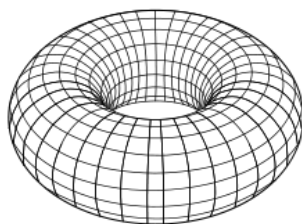


Figura N° 4.4 Superficie toroidal

4.2. Algoritmo superficie deformable

El algoritmo se implementa de forma discreta a través de iteraciones, en donde cada una significa recorrer todo los puntos “Snake” del mapeo, el cual debe estar inicializado cerca del objeto a segmentar para que la energía externa capture al punto Snake, la cual está representada por la magnitud del gradiente de las intensidades de los pixeles de la imagen, en el punto donde se desea calcular la energía. Una técnica aplicada para aumentar este rango de la energía es aplicar una máscara gaussiana difuminando la fuerza externa (energía en el funcional) a través de toda la imagen. Como consecuencia negativa se tiene la pérdida de precisión en la segmentación.

Cada punto Snake $v(s,r)$ es movido en el espacio (x,y,z) a una energía menor calculada a través de la ecuación (3.9), la cual tiene como energía externa $P(v(s,r)) = -\|\nabla I(v(s,r))\|^2$, que guiara la superficie en su deformación. Al añadirle el filtro gaussiano, esta energía queda definida como sigue:

$$P(v(s,r)) = -\|\nabla [G_\sigma(v(s,r)) * I(v(s,r))]\|^2 \quad (4.2)$$

Este filtro gaussiano tiene un máximo valor de intensidad en el voxel central y disminuye hacia los extremos, lo cual distribuye la fuerza dentro del volumen de información. Este filtro corresponde a una máscara cúbica (Fig. 4.5) que recorre el volumen de datos con intensidades descritas por la ecuación 4.4, siendo σ la desviación estándar, en cuanto menor sea este parámetro más rápido será el difuminado.

$$g(x,y,z) = e^{-\frac{x^2+y^2+z^2}{2\sigma^2}} \quad (4.3)$$

$$G(x,y,z) = \frac{g(x,y,z)}{\min_{x,y,z} \{g(x,y,z)\}} \quad (4.4)$$

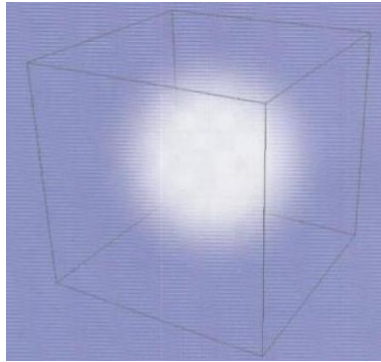


Figura N° 4.5 Fuente gaussiana volumétrica.

Luego es necesario obtener las derivadas discretas de los puntos de control de la superficie deformables para poder calcular el funcional (3.9). Esto se realiza como se muestra a continuación, de acuerdo a la definición del mapeo discreto (plano rs en la figura 4.2).

$$\frac{\partial}{\partial s}(v(s, r)) = \frac{v(s+1, r) - v(s-1, r)}{2} \quad (4.5)$$

$$\frac{\partial}{\partial r}(v(s, r)) = \frac{v(s, r+1) - v(s, r-1)}{2} \quad (4.6)$$

$$\frac{\partial^2}{\partial s \partial r}(v(s, r)) = \frac{v(s+1, r+1) + v(s-1, r-1) - v(s+1, r-1) - v(s-1, r+1)}{4} \quad (4.7)$$

$$\frac{\partial^2}{\partial s^2}(v(s, r)) = v(s+1, r) + v(s-1, r) - 2v(s, r) \quad (4.8)$$

$$\frac{\partial^2}{\partial r^2}(v(s, r)) = v(s, r+1) + v(s, r-1) - 2v(s, r) \quad (4.9)$$

4.2.1. Gradiente volumen de datos

Como se mencionó anteriormente, la fuerza externa es provista por la magnitud del gradiente de la imagen en x,y,z, la cual se calcula en cada pixel a través de las intensidades de los pixeles, la cual puede definirse por:

$$\frac{\partial}{\partial x}(I(x, y, z)) = \frac{I(x+1, y, z) - I(x-1, y, z)}{2} \quad (4.10)$$

$$\frac{\partial}{\partial y}(I(x, y, z)) = \frac{I(x, y+1, z) - I(x, y-1, z)}{2} \quad (4.11)$$

$$\frac{\partial}{\partial z}(I(x, y, z)) = \frac{I(x, y, z+1) - I(x, y, z-1)}{2} \quad (4.12)$$

4.2.2. Método exhaustivo

El método exhaustivo o “Greedy” consiste en el cálculo de la energía en el punto snake $v(s,r)$ a mover dentro un volumen de búsqueda, como se presenta en la figura 4.6 donde los puntos rojos representan los puntos snake y los cubos de color verde las posibles nueva posición del punto Snake. Este volumen de búsqueda cubico y tamaño impar, es posicionado de tal forma que su voxel central coincida con el punto Snake a mover.

Se calcula la energía en cada voxel del volumen de búsqueda sustituyendo la posición anterior (x,y,z) por la posición dentro de la búsqueda que obtuvo menor energía, para esto se debe calcular las fuerzas internas como externas en cada punto del volumen de búsqueda para calcular el valor del funcional (energía) debiendo recorrer todos los voxeles que lo conforman. Esto se realiza a lo largo de todos los puntos del mapeo del snake, con la premisa de que la suma de los mínimos locales implicará un mínimo global, correspondiente a la segmentación.

Cabe destacar que el volumen de búsqueda está dado por una matriz de $3 \times 3 \times 3$ en la figura 4.6, la cual no necesariamente debe ser de ese tamaño pudiendo ser de $5 \times 5 \times 5$ o más, lo cual aumenta el rango de búsqueda, pero a su vez la cantidad de cálculos a realizar.

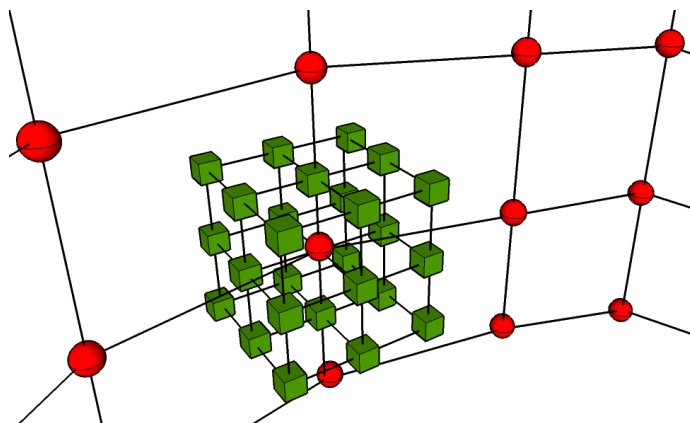


Figura N° 4.6 Vecindad de un punto Snake junto a puntos vecinos.

El algoritmo que se ha descrito lo podemos encontrar como diagrama de flujo en la figura 4.7, donde se recorren todos los puntos de control de la superficie $(v(s,r))$.

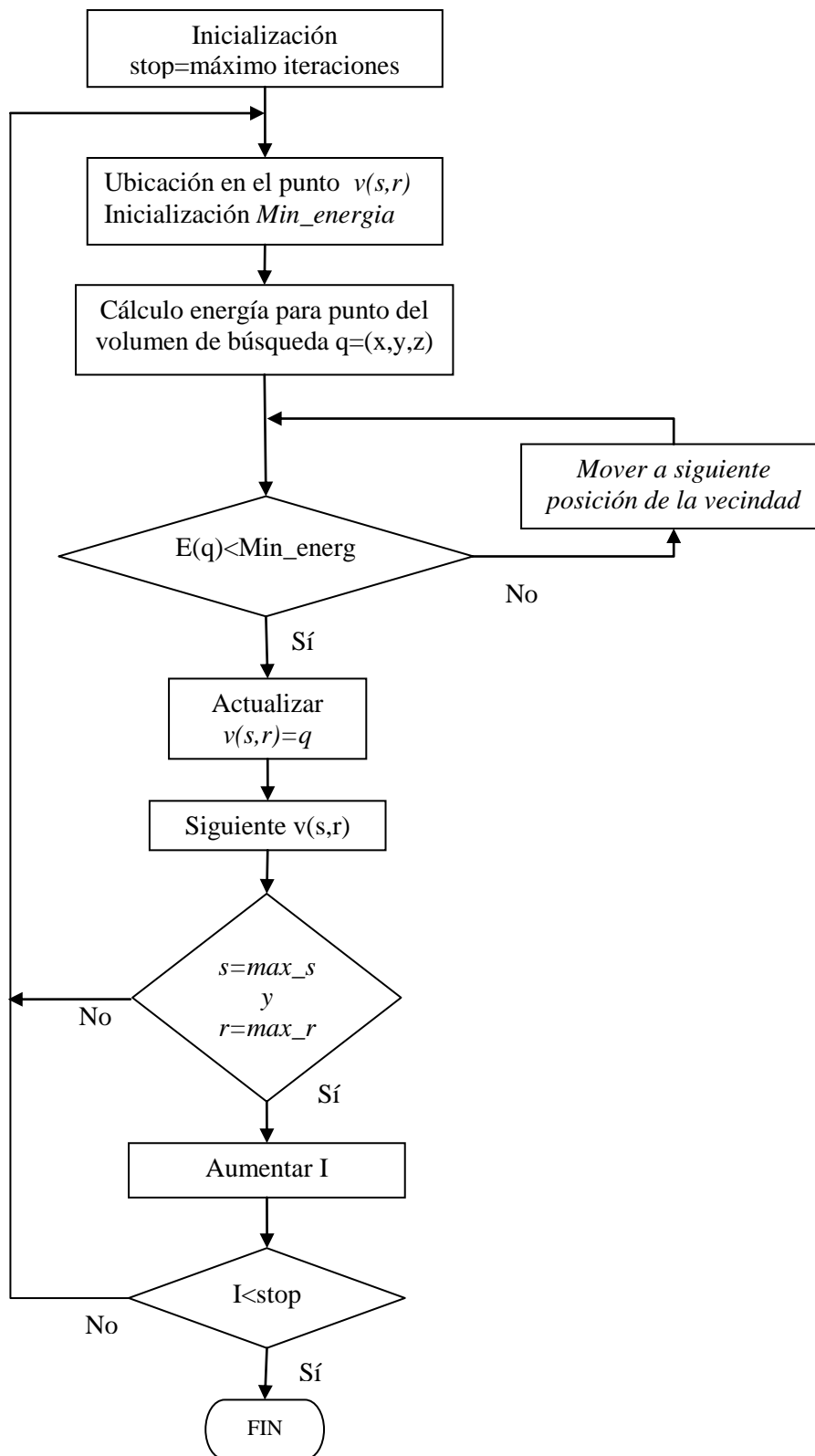


Figura N° 4.7 Diagrama de flujo, búsqueda minimización a través de Greedy.

4.2.3. Método Simulated Annealing

Como se explico en el capitulo anterior, el método Greedy busca una solución dentro de un volumen local (vecindad del punto), encontrando la posición a mover el punto que tiene la menor energía, pero que no necesariamente puede llevar a una energía global mínima al finalizar las iteraciones.

Una técnica bastante utilizada es la de permitir movimientos a soluciones parciales malas, no deseables, o malos movimientos, que al final nos lleven a buen término.

Simulated Annealing corresponde a una técnica de optimización de meta-heurística basada en un proceso térmico metalúrgico, proceso que calienta los metales logrando que las partículas ganen energía, para luego descender la temperatura controladamente lo cual genera que las partículas se agrupen de forma ordenada, lo que finalmente se traduce en un metal de mejor calidad.

El algoritmo de recocido simulado o SA (Simulated Annealing) se basa en una simulación hecha a través del algoritmo de metrópolis el cual genera un vecino de la posición actual y le calcula la energía, esta de ser menor se acepta inmediatamente como solución, pero de ser mayor no se desecha inmediatamente ya que se le calcula una probabilidad dependiente de una temperatura y de la diferencia de energía entre el estado anterior y el actual dada por el vecino generado, calculándose mediante la ecuación 4.13.

$$P = e^{\left(\frac{E_i - E_j}{k \cdot T}\right)} \quad (4.13)$$

Donde T es la “temperatura” y k es la constante de Boltzmann, aunque generalmente no se consideran, ya que no cumplen trabajo alguno en los problemas de optimización, en su reemplazo es utilizado un parámetro t que asume la temperatura como se ve en (4.13).

Luego SA toma esa simulación y genera un numero aleatorio r el cual tiene distribución uniforme entre 0 y 1, de ser $r < P$ se deja como solución, de no ser así se desecha definitivamente el vecino generado como solución, repitiendo la simulación varias veces para luego disminuir el parámetro t de temperatura para la próxima solución en un factor alfa como:

$$t_{k+1} = \alpha \cdot t_k \quad (4.13)$$

Esto quiere decir que a altas temperaturas se acepta una gran cantidad de movimientos, pero que al ir descendiendo la temperatura, se aceptaran pocos, debido a que la solución debería estar cerca de un óptimo aproximado, realizando pequeños movimientos.

Esto se realiza varias veces hasta llegar a una condición de detención provista por el usuario.

La técnica posee parámetros que de alguna forma influyen sobre el resultado final como lo son la temperatura inicial, la función de decrecimiento de la temperatura (que no necesariamente puede ser lineal como (4.13)), la solución inicial, el número de repeticiones de la simulación antes del decremento de la temperatura, y la condición de detención para dar por finalizada la búsqueda.

Resulta beneficioso en cuanto a tiempo en búsquedas de un óptimo dentro de una gran vecindad de soluciones, en el caso particular aplicado a superficies deformables no se busca una reducción de los tiempos de cálculo, más bien una mejor segmentación llevando a los puntos de control de la superficie a energías mayores que después se muevan a mejores posiciones.

El algoritmo se aprecia en la figura 4.8, el cual es aplicado a cada punto de control de la superficie deformable, donde “nRep” se refiere a la cantidad de repeticiones o cantidad de búsqueda aleatoria de nuevos movimiento o lugares a mover el punto de control. δ consiste en la diferencia entre la nueva energía $E(s_{k+1})$ y la anterior $E(s_k)$.

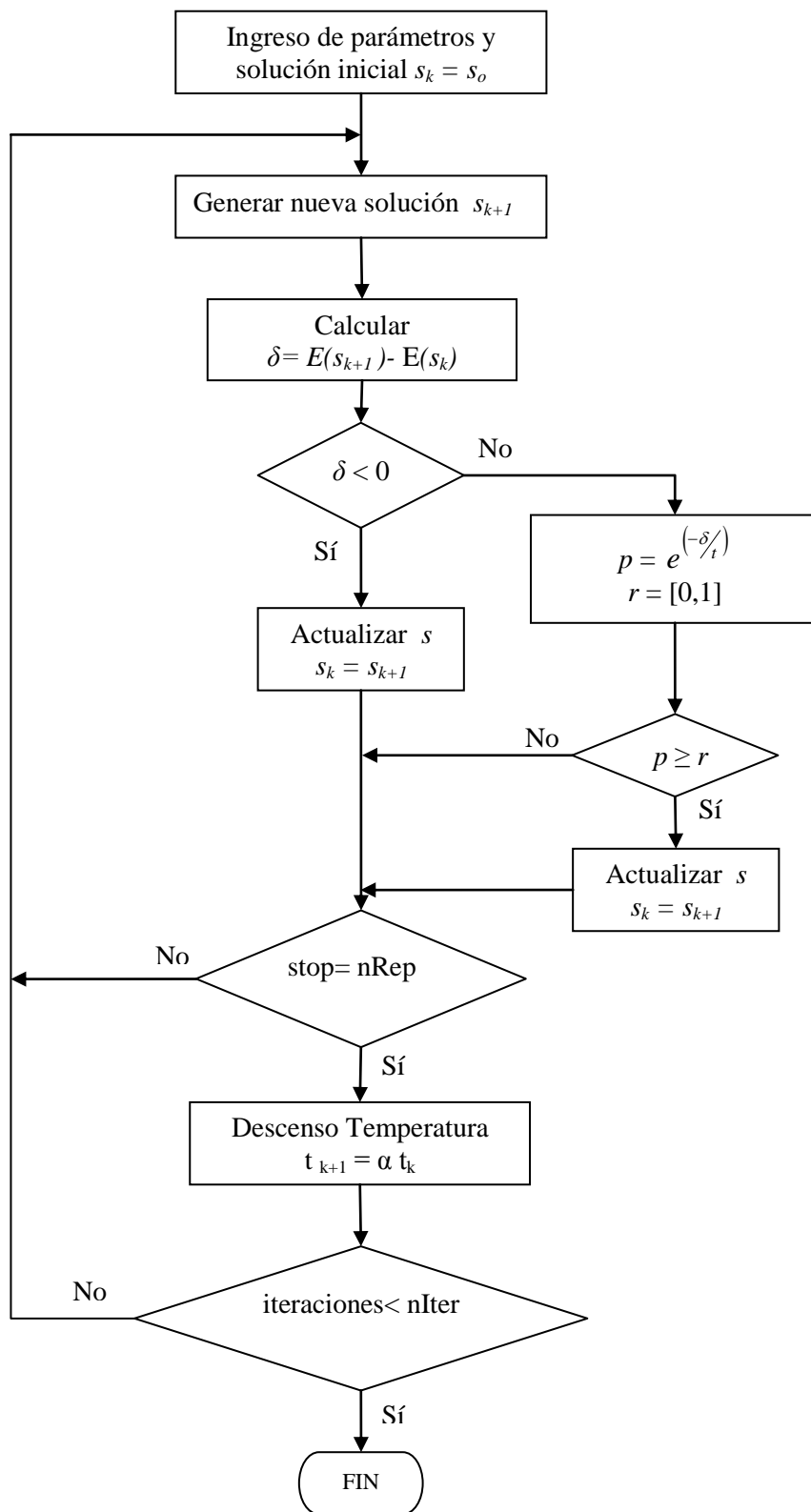


Figura N° 4.8 Diagrama de flujo búsqueda minimización a través de Simulated Annealing.

4.2.4. Método Simulated Annealing con Threshold Accepting

Este método es una variante de Simulated Annealing (SA) el cual se basa al igual que SA en generar una solución, pero en cambio este método rechaza de inmediato una posición que no cumpla un umbral de energía (T), el cual es reducido en cada iteración. Esto implica un menor tiempo de cálculo, debido que al comparar los resultados se sabe de inmediato si se acepta o no la nueva solución, por otro lado, este nuevo método implica una complicación en la elección del umbral inicial, el cual es de difícil estimación, debiendo realizar estudios anteriores a la aplicación del algoritmo o sencillamente, realizar pruebas con posibles umbrales analizando los resultados.

El algoritmo se puede observar como diagrama de flujo en la figura 4.9, donde “nIter” representa el número de iteraciones a cada punto de control, movimiento o búsqueda de lugar con menor energía, “nRep” corresponde a la cantidad de búsquedas o vecinos generados aleatoriamente en cada iteración y donde μ representa un factor constante de descenso del umbral.

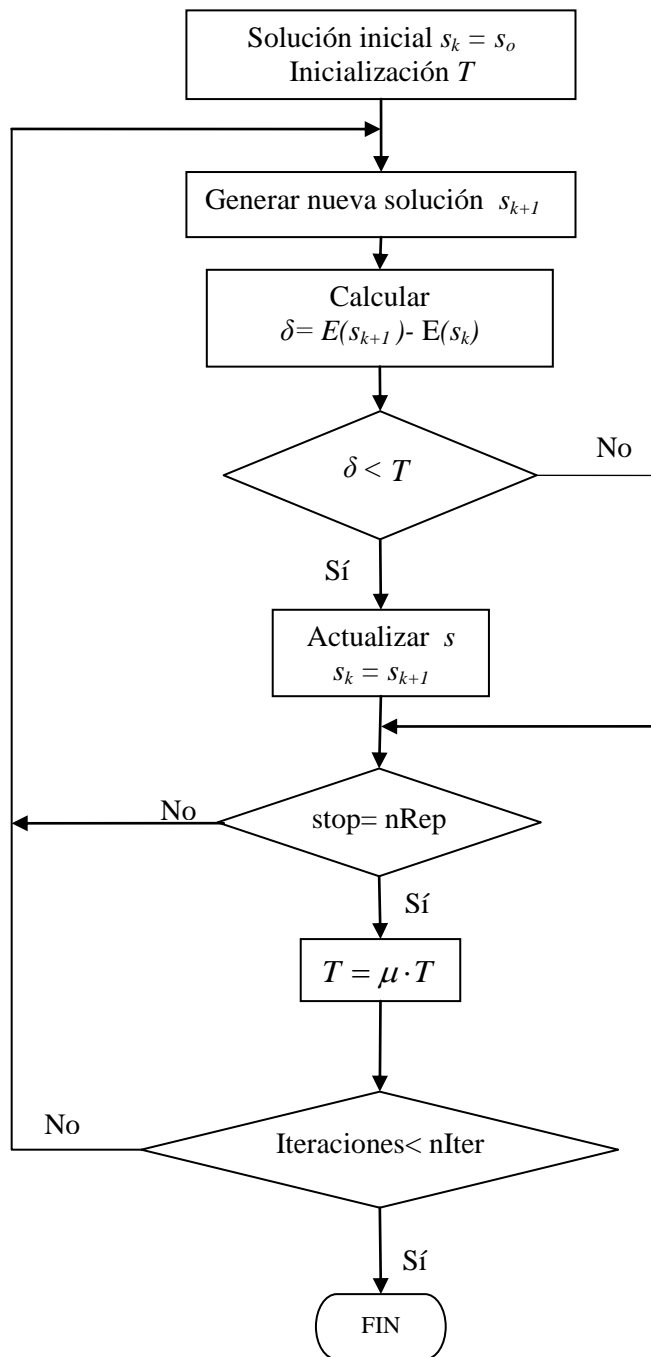


Figura N° 4.9 Diagrama de flujo para búsqueda de minimización funcional a través de Simulated Annealing con Threshold Accepting.

4.3. LIBRERÍA DE VISUALIZACIÓN DE VOLÚMENES DE DATOS VTK.

VTK (Visualization Toolkit), es una librería de código abierto orientada a objeto, par la visualización, computación grafica y visualización de datos de ingeniería, generada en C++. Posee intérpretes para lenguajes script como lo son Tcl, Java y Python, además de ser una librería que corre bajo distintas plataformas como Windows, Linux y Macintosh.

Desde este proyecto (VTK) se han generado programas tan útiles como lo son Cmake, el cual se utiliza para agrupar las librerías necesarias para la compilación multiplataforma, Volview, creado para la visualización de volúmenes de datos médicos o científicos y Paraview para la visualización de grandes cantidades de datos científicos pudiendo utilizar recursos distribuidos, correr sobre una supercomputadora o un computador de escritorio, utilizado para el análisis de movimientos de fluidos, túneles de viento y resistencia de materiales entre otras aplicaciones.

VTK posee la capacidad de leer una gran cantidad de formatos de imágenes como los son los archivos Dicom, Vtk, Obj, Bmp, Png, ingresar a la escena tridimensional texto plano, como también texto tridimensional, varios widget para interfaces de usuarios y controles de cámara como son acercamiento, alejamiento y recorrido de la escena. Además de poseer algunos algoritmos de procesamiento de imágenes.

4.3.1. Orientación a objetos

Como se mencionó anteriormente, VTK es una librería orientada a objetos, debido a la necesidad de ser modular y extensible. Con cada nueva versión de esta librería obtenemos nuevas funciones y mejoras, con un alto grado de especialización, llegando a ser una librería robusta, grande y compleja, con unidades más pequeñas y más fáciles de solucionar y crear.

Básicamente, un objeto en programación es una instancia de una clase, la cual es una abstracción de un modelo para un propósito específico. Este objeto tendrá un nombre, atributos y operaciones que se pueden obtener o realizar con él.

Como ejemplo se podría tomar un roble, el cual es un árbol que tiene un nombre, atributos como diámetro del tronco, altura, edad, ubicación, y operaciones como crecer. Con esta etapa de abstracción pasamos desde un objeto en la vida real a un modelo virtual (abstracto).

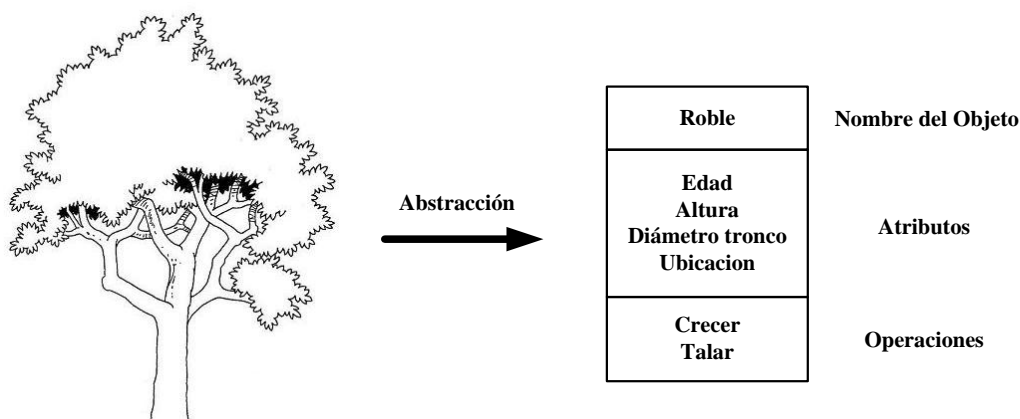


Figura N° 4.10 Esquema ejemplo de orientación a objeto.

4.3.2. Actores y objetos de una escena 3D

Al igual que en una escena de película, se tiene actores, luces y cámaras, donde éstas últimas nos permiten visualizar objetos tridimensionales en la pantalla del computador o televisor.

Los actores son los datos que introducimos en la escena y son los que deseamos observar, poseen una posición, y propiedades como el color o textura.

Las luces proporcionan la iluminación creando sombras o efectos visuales, por otra lado la cámara provee el control de la visualización, ya sea los acercamiento que se desee realizar, como propiedades de posición o ángulo de visión a los objetos. Por otra parte se tiene un render y una ventana de renderizado. Esta última controla las ventanas del sistema operativo donde se mostrará lo capturado por la cámara, en tanto el render se hace cargo de la transformación de los objetos virtuales a una imagen 2D, proceso que se explicará en detalle a continuación.

4.3.3. Renderizado

El renderizado es el proceso por el cual se proyecta un modelo tridimensional dispuesto en una escena, a una imagen bidimensional. En esta etapa de la visualización entra los efectos físicos como la difracción de la luz, iluminación, brillo, textura, sombreado etc.

Librerías como VTK nos proporciona un pre-renderizado al momento de mover la cámara en situaciones cuando se rodea el objeto, lo cual proporciona una visualización fluida pero de menor calidad, luego, fijada la posición de visualización, se realiza el renderizado final de mayor detalle, como se muestra en las figuras 4.11 y 4.12.

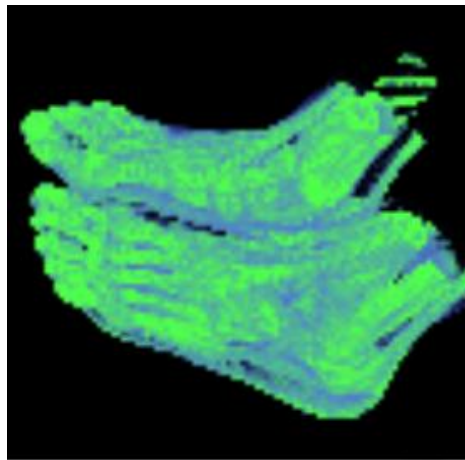


Figura N° 4.11 Volumen de datos pre-renderizado

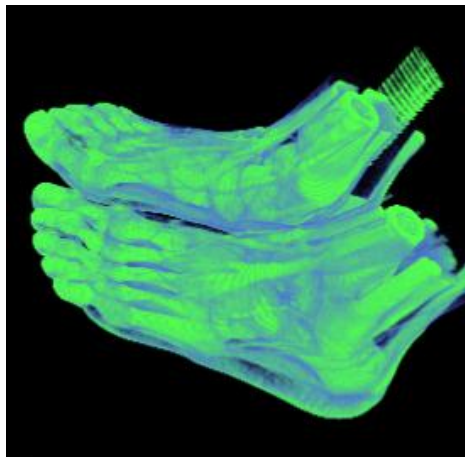


Figura N° 4.12 Volumen de datos renderizado.

Entre las técnicas conocidas de renderizado de volúmenes encontramos Ray Casting, Ray tracing y Scanline Rendering entre otros, los que proveen distintos niveles de calidad y tiempo de procesamiento.

4.3.2.1. Ray Casting

La técnica de Ray Casting, utilizada en la implementación, consiste en trazar un rayo desde un pixel de la imagen 2D hacia el volumen de datos haciendo esto para todos los pixeles de la imagen, con rayos paralelos entre sí y perpendiculares a la imagen 2d, como lo muestra la figura 4.13.

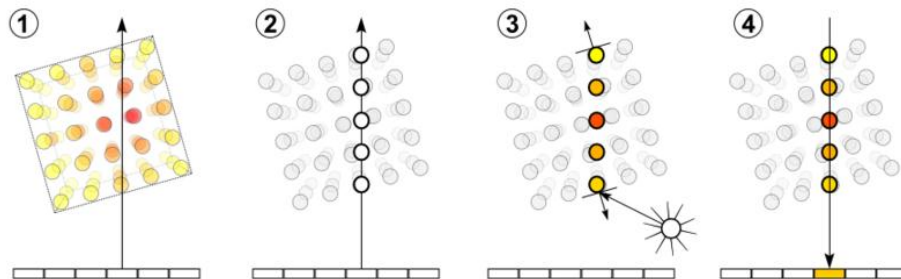


Figura N° 4.13 Etapas renderizado.

El renderizado por ray casting en volúmenes de datos podemos definirlo en 4 etapas

- 1.- Repartir rayos: lanzar el rayo desde un pixel de la imagen
- 2.- Muestrear: se toman muestras de valores de intensidad del voxel equidistantes, a lo largo del rayo
- 3.- Sombreado: a los puntos muestreados en el rayo, se le calcula el gradiente, para luego darle una intensidad que depende de la dirección en la superficie y la fuente de luz.
- 4.- Composición: obtenidas las intensidades de cada muestra, se crea una intensidad a través de la ecuación de rendering que asemeja a hojas de papel celofán de distintos colores en un proyector, obteniendo la intensidad del pixel.

4.3.4. Función de Opacidad

Al tener un volumen de datos obtenido de tomografías, resonancias magnéticas etc., obtenemos información representada por un volumen de datos (ejemplo un cubo), el cual se debe transparentar para poder visualizar estructuras internas. De no ser así, se vería solo la superficie del volumen de datos. Esto se soluciona con una función de opacidad la cual es representada a través de puntos formando una curva spline, con la que se define cuan opaco son los valores de intensidad del volumen de datos al momento de realizar el renderizado.

Una intensidad con opacidad cero, querrá decir que es totalmente transparente, y una opacidad de 1, que es una intensidad totalmente opaca. Ejemplo de esto se visualiza en la figuras 4.14 y 4.15 donde se aplico una función de opacidad para dejar transparentes los datos con una baja intensidad y que corresponden al “fondo de la imagen (Slice)” o exterior del volumen.



Figura N° 4.14 Volumen sin función de opacidad



Figura N° 4.15 Volumen con función de opacidad parametrizada para ver los huesos

4.3.5. Función de transferencia

Esta función se utiliza para dar un color a las intensidades obtenidas en el renderizado. Esta función asocia las intensidades a un mapeo de color, el cual puede estar definido en RGB o HSV, realizando un gradiente al color entre los puntos.

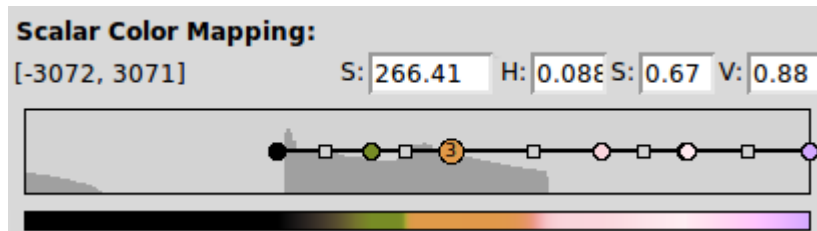


Figura N°. 4.16 Función de transferencia de coloreado, en modelo HSV de color.



Figura N° 4.17 Resultado final del renderizado con la función de transferencia.

Capítulo 5

ENSAYOS Y PRUEBAS.

INTRODUCCION.

Este capítulo muestra los resultados cualitativos obtenidos de la aplicación de una superficie deformable sobre volúmenes de datos, los cuales fueron realizados sobre una imagen virtual de ejemplo, una imagen médica y finalmente un tronco de madera.

En la imagen virtual se segmentó una esfera, en la imagen médica una rodilla y finalmente en el tronco se segmentó un nudo, ésto aplicado a las tres técnicas mencionadas de búsqueda de solución como lo son Greedy, Simulated Annealing y Simulated Annealing con Threshold Accepting.

5.1. ENSAYOS CON VOLUMEN DE PRUEBA

La imagen de prueba consiste en 32 archivos Dicom, el cual forma un volumen de información de $32 \times 32 \times 32$, que en su interior contiene una esfera sólida, como se muestra en la figura 5.1 donde se visualiza las imágenes que contienen los archivos.

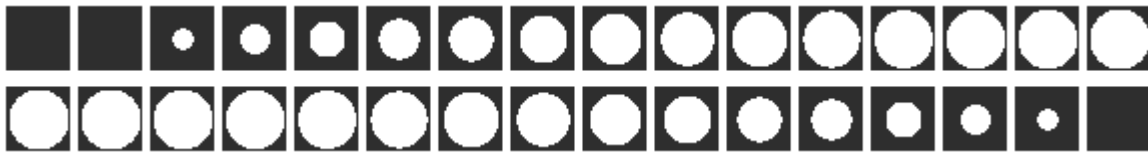
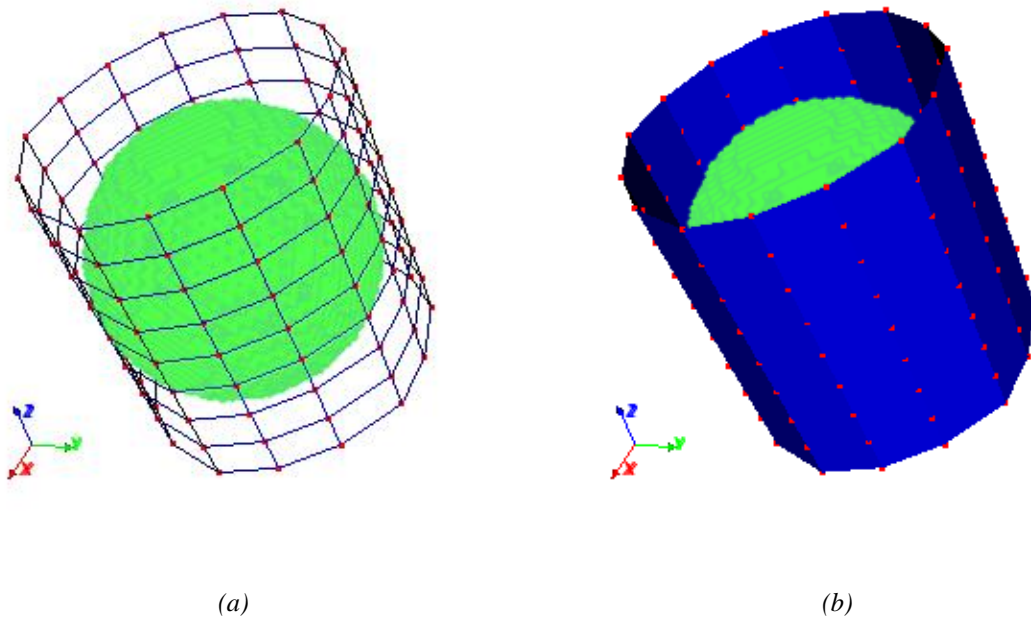


Figura N° 5.1 Imágenes de archivos Dicom volumen prueba.

La cantidad de puntos utilizada fue 135, distribuidos de forma cilíndrica en su condición inicial como se visualiza en la figura 5.2. Esta condición inicial o superficie inicial se utilizara para los 3 algoritmos de búsqueda utilizados en el volumen de prueba.



(a)

(b)

Figura N° 5.2 Imágenes de archivos Dicom volumen prueba.

Greedy

El algoritmo Greedy se aplico 15 veces a cada punto. Luego de las iteraciones se obtiene la segmentación visualizada en la figura 5.3, para un volumen de búsqueda de 3x3x3.

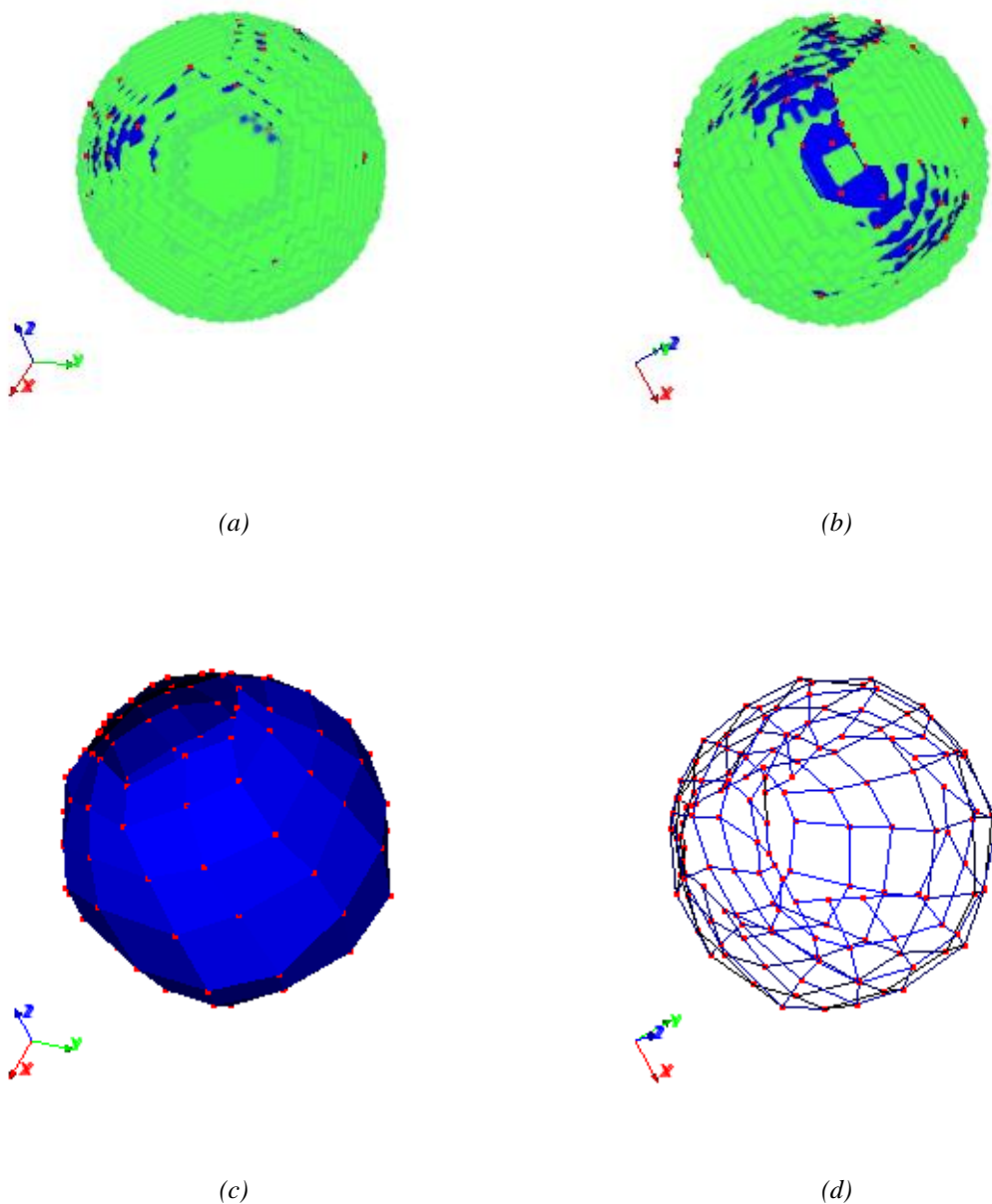


Figura N° 5.3 Segmentación a través de Greedy, (a) y (b) Volumen de datos a segmentar con superficie deformada, (c) Superficie deformada, (d) Malla de la superficie

Simulated Annealing

En el algoritmo de simulated annealing se utilizó como parámetro de reducción de temperatura $\alpha = 0.85$, con 15 iteraciones del algoritmo y 100 repeticiones para la búsqueda de posibles soluciones antes de descender la temperatura. Los resultados visuales son mostrados en la figura 5.4.

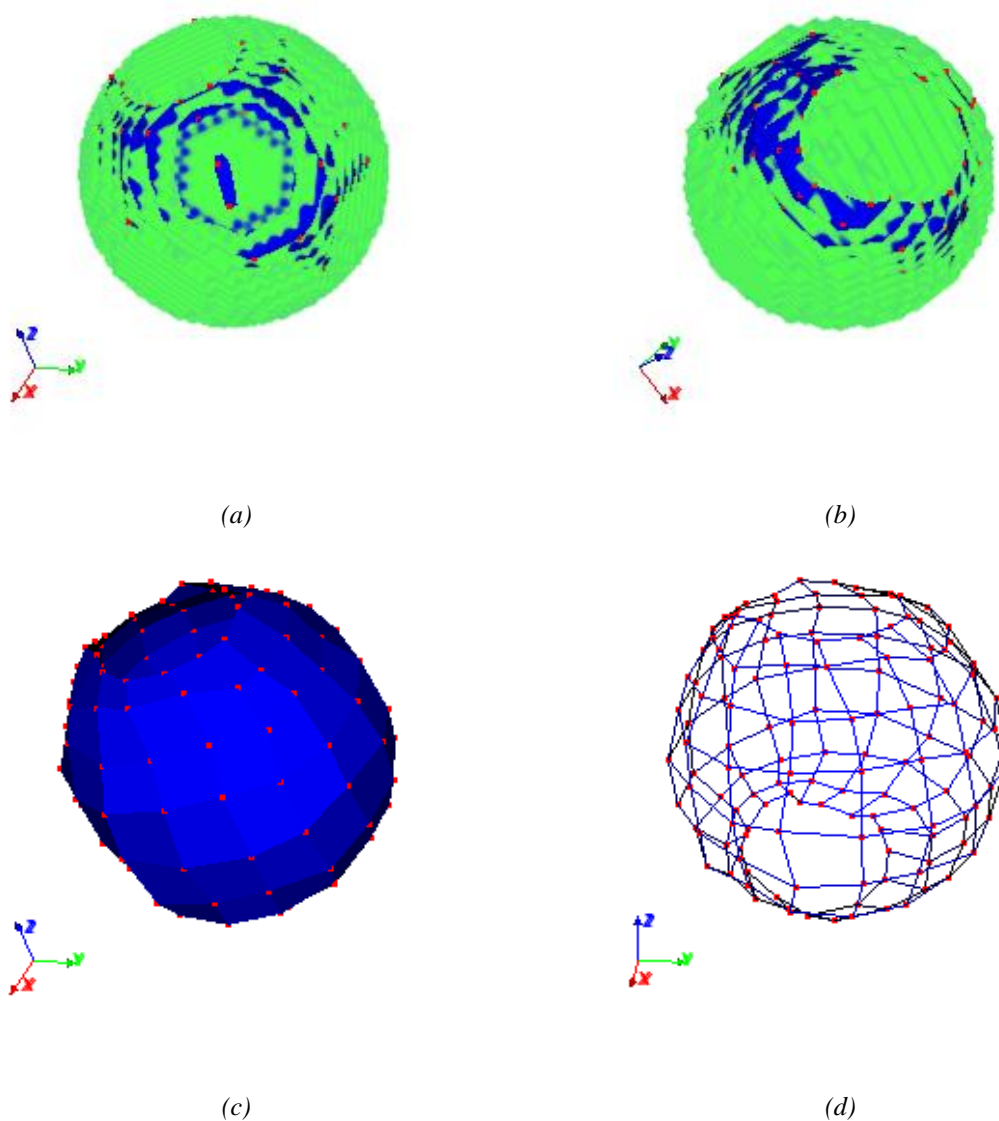


Figura N° 5.4 Segmentación a través de Simulated Annealing, (a) y (b) Volumen de datos a segmentar con superficie deformada, (c) Superficie deformada, (d) Malla de la superficie

Simulated Annealing Threshold Accepting

En el algoritmo de simulated annealing con threshold accepting se utilizó como umbral (Threshold, T) inicial el valor 0.03, el cual fue descendido con $\mu=0.7$, realizando 15 iteraciones y 100 repeticiones para la búsqueda de posibles soluciones. Los resultados visuales son expuestos en la figura 5.5.

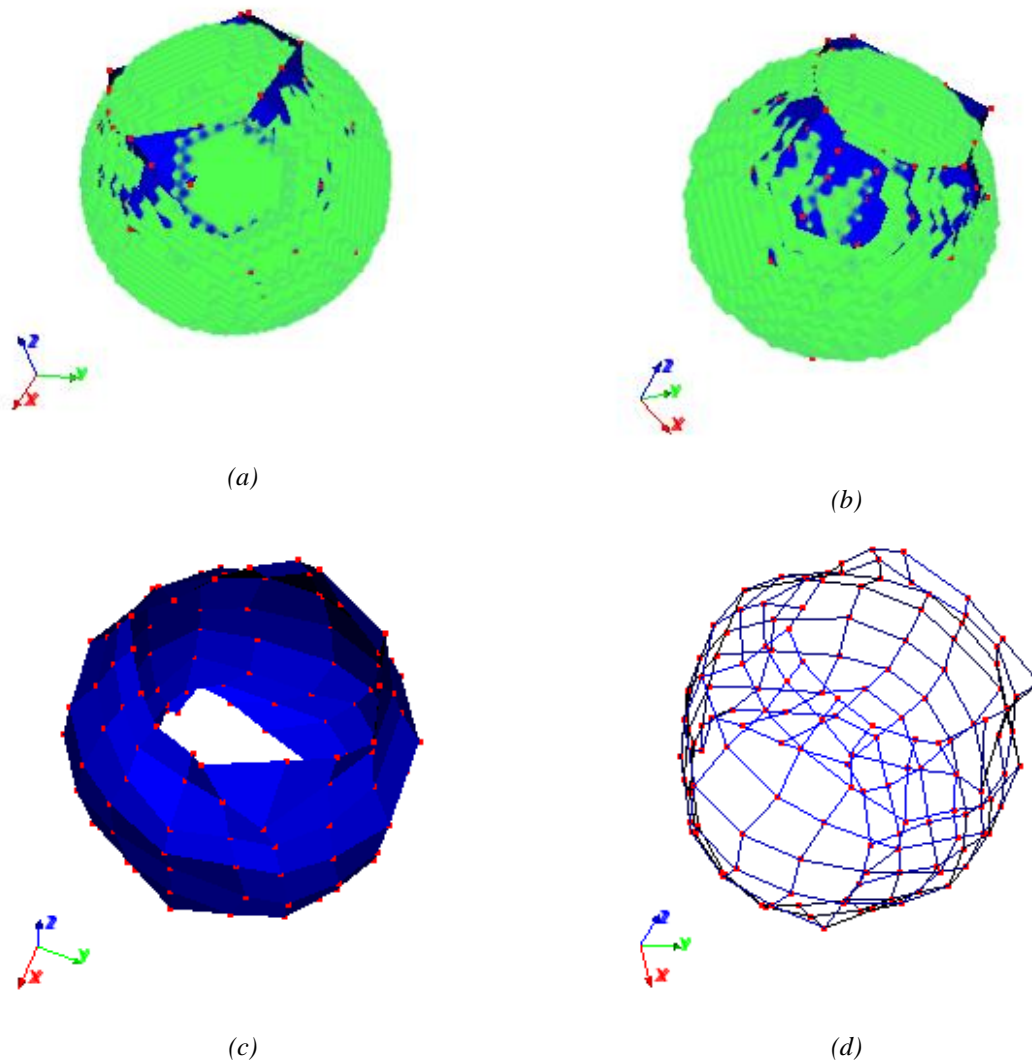


Figura N° 5.5 Segmentación a través de Simulated Annealing con Threshold Accepting, (a) y (b) Volumen de datos a segmentar con superficie deformada, (c) Superficie deformada, (d) Malla de la superficie

Resultados cuantitativos de la aplicación de las 3 técnicas de búsqueda son resumidas en la tabla 5.1

	Método exhaustivo (Greedy)	Simulated Annealing	Simulated Annealing Threshold
Lectura archivos	0,07	0,073	0,062
Filtro gaussiano	0,013	0,013	0,031
Cálculo magnitud gradiente	0,003	0,007	0,000
Módulo gradiente a matriz	0,073	0,077	0,078
Deformación	0,057	0,125	0,047
Render	0,329	0,281	0,249

Tabla N° 5.1 Resumen de tiempos en segundos, Volumen Sintético con las diferentes técnicas.

Luego de aplicarlas 3 técnicas con resultados que podemos observar en las figuras 5.3, 5.4 y 5.5, se aprecia que Simulated Annealing (SA) provee una superficie ligeramente más suave, con contornos más redondeados que SA Threshold Accepting (SA-TA) o Greedy. Por otro lado SA-TA mueve en menor medida los puntos de control superiores e inferiores dejando una superficie más cilíndrica que esférica si lo comparamos con Greedy o SA. En cuanto a los tiempos de procesamiento SA-TA resultó de mejor performance que el resto de las búsquedas, lo cual era esperable debido a su salida al encontrar una energía menor al threshold. Lo que podría llamar la atención es el elevado tiempo de SA, debido a la parametrización del algoritmo que busca 100 posibles vecinos antes de descender la temperatura, caso muy distinto de Greedy que busca sólo en los 27 posibles movimientos, esta situación se analizará en el apartado 5.4 con un volumen de búsqueda de 9x9x9.

5.2. PRUEBA CON VOLUMEN DE DATOS ÁREA MEDICA

Para el experimento del área médica, es utilizada una base de datos abierta al público que consiste en las tomografías computarizadas del proyecto Visible Human, la cual está destinada principalmente al estudio médico, con imágenes de los cortes “literales” de dos cuerpos que fueron congelados, un hombre y una mujer, además de tomografías computarizadas de éstos cuerpos.

Se utilizó 236 cortes o “Slice” en formato Dicom correspondientes al cuerpo femenino de la base de datos, los cuales están espaciados cada 1mm entre cortes. Estas imágenes corresponden al tercio distal de la pierna, (entre la pantorrilla y el muslo). Algunas imágenes de estos cortes se aprecian en la figura 5.6, con su correspondiente reconstrucción tridimensional en la figura 5.7.



Figura N° 5.6 Imágenes de archivos Dicom volumen área médica.

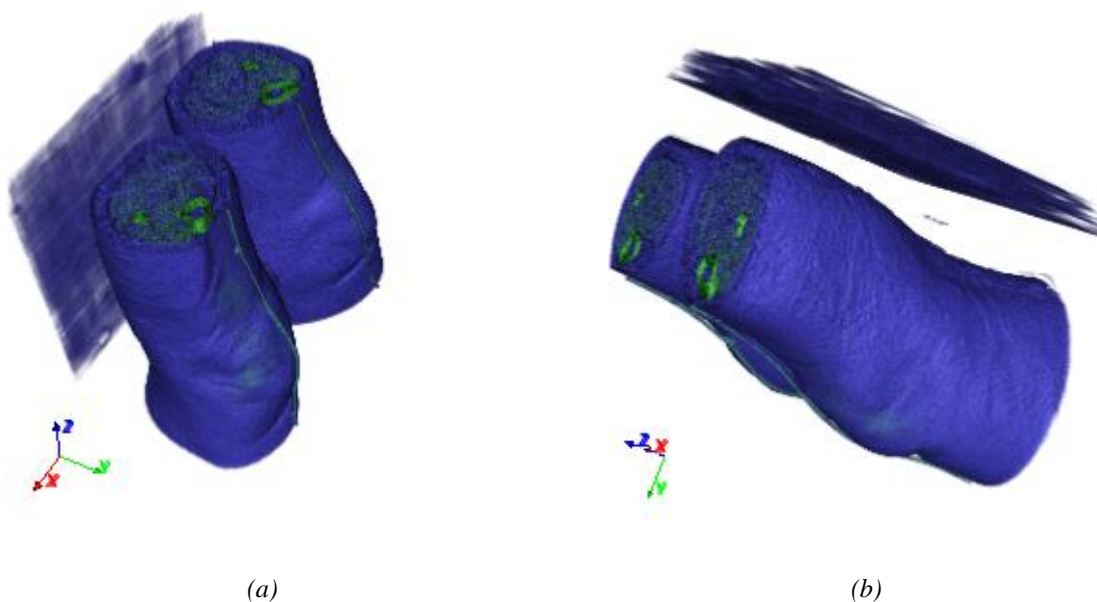


Figura N° 5.7 Imágenes de archivos Dicom volumen área médica, reconstrucción 3D.

5.2.1. Segmentación de rodilla.

La inicialización de la superficie se realizó de forma manual realizando sucesivos ensayos llegando a la superficie inicial que se ve en la figura 5.8 con un total de 135 puntos de control de la superficie.

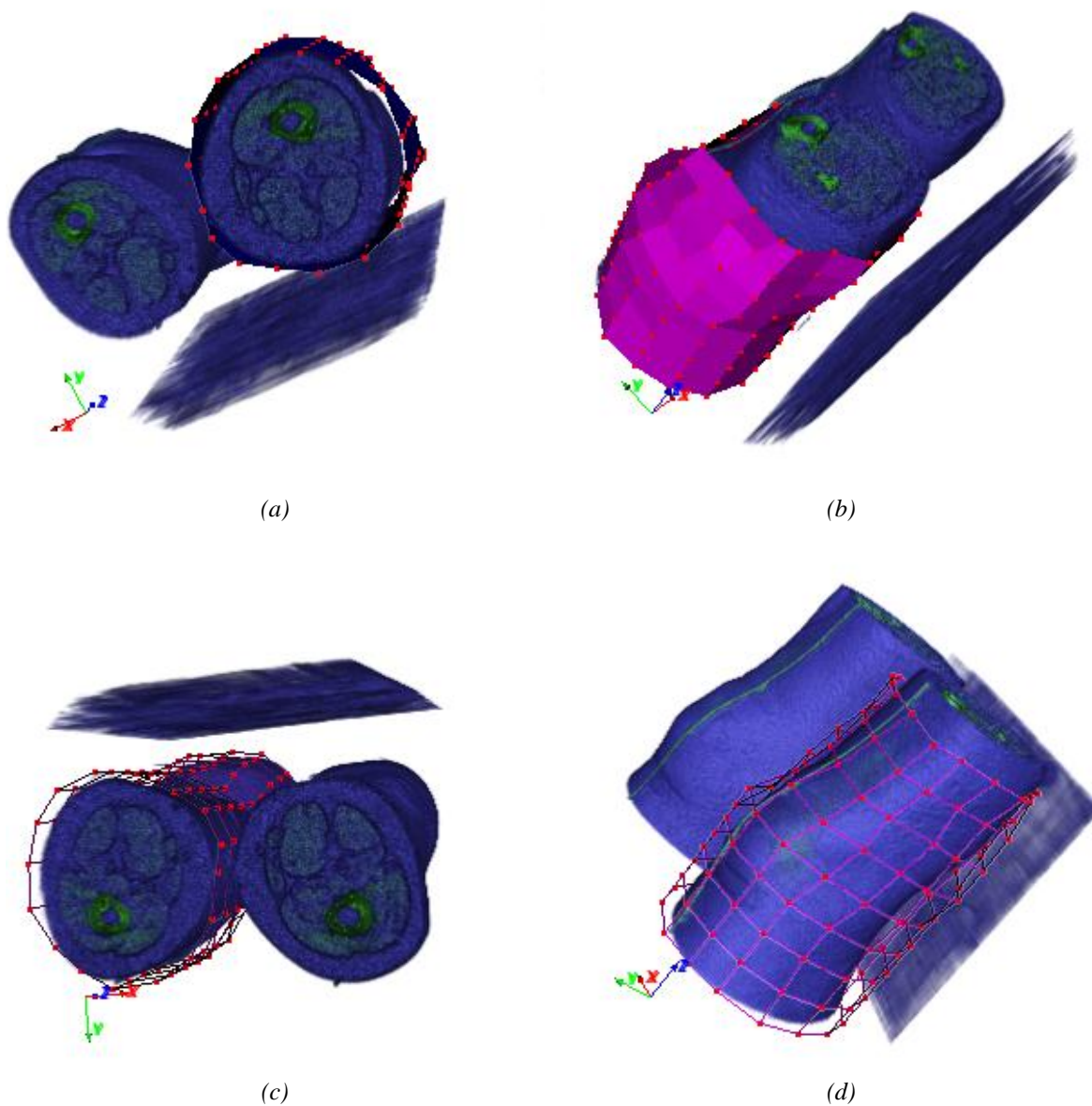


Figura N° 5.8 Superficie inicial, (a) y (b) Volumen de datos con superficie inicial, (c) y (d) volumen de datos con malla de la superficie inicial.

Greedy

El algoritmo Greedy se aplicó 30 veces a cada punto de control. Obteniendo los resultados de la figura 5.9, para un volumen de búsqueda de $3 \times 3 \times 3$.

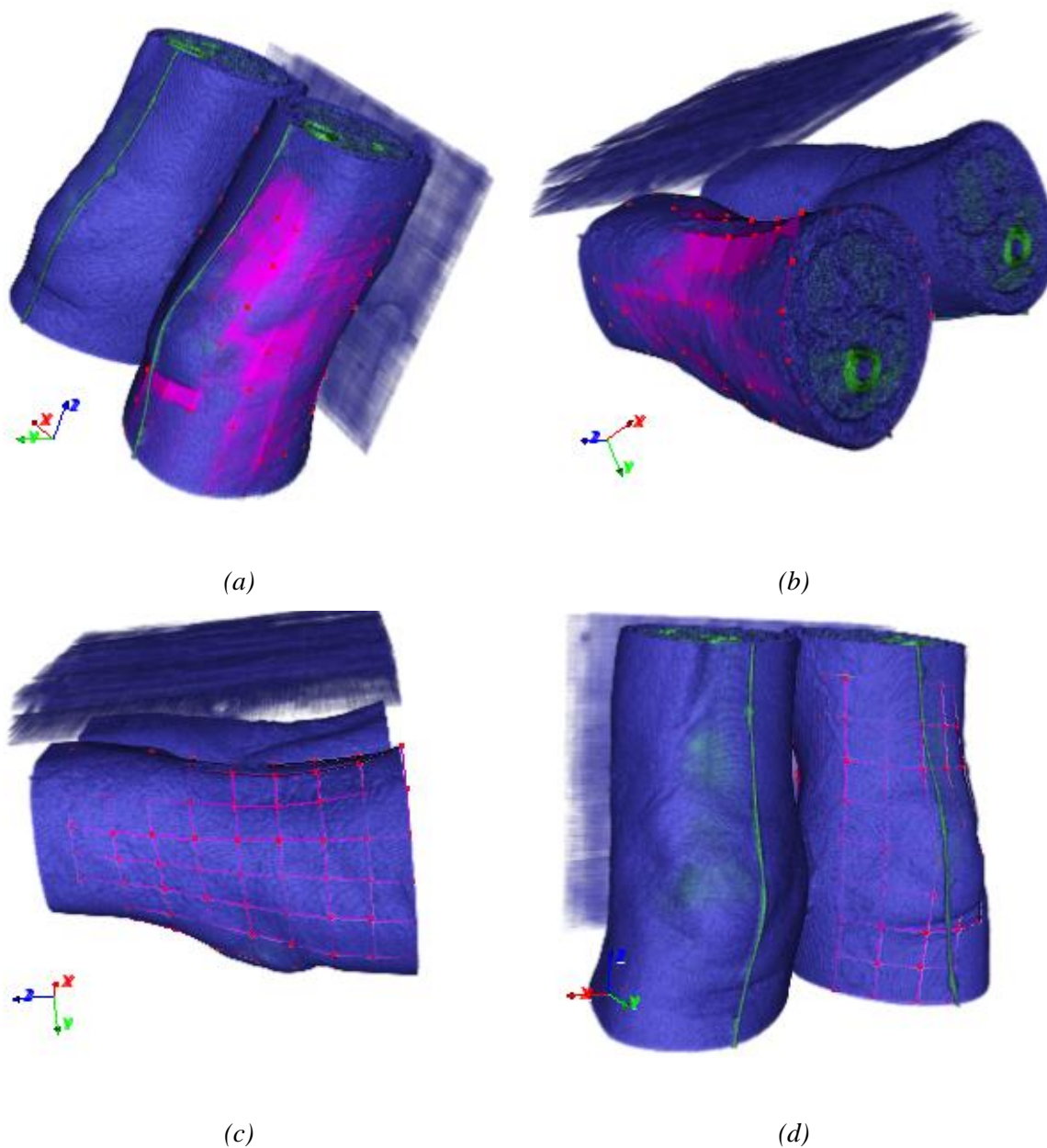


Figura N° 5.9 Superficie deformada y volumen, (a) y (b) Volumen de datos con superficie deformada a través de Greedy, (c) y (d) volumen de datos con malla de la superficie deformada a través de Greedy.

La superficie final segmentada sin el volumen de datos, podemos apreciarla con más detalle en figura 5.10.

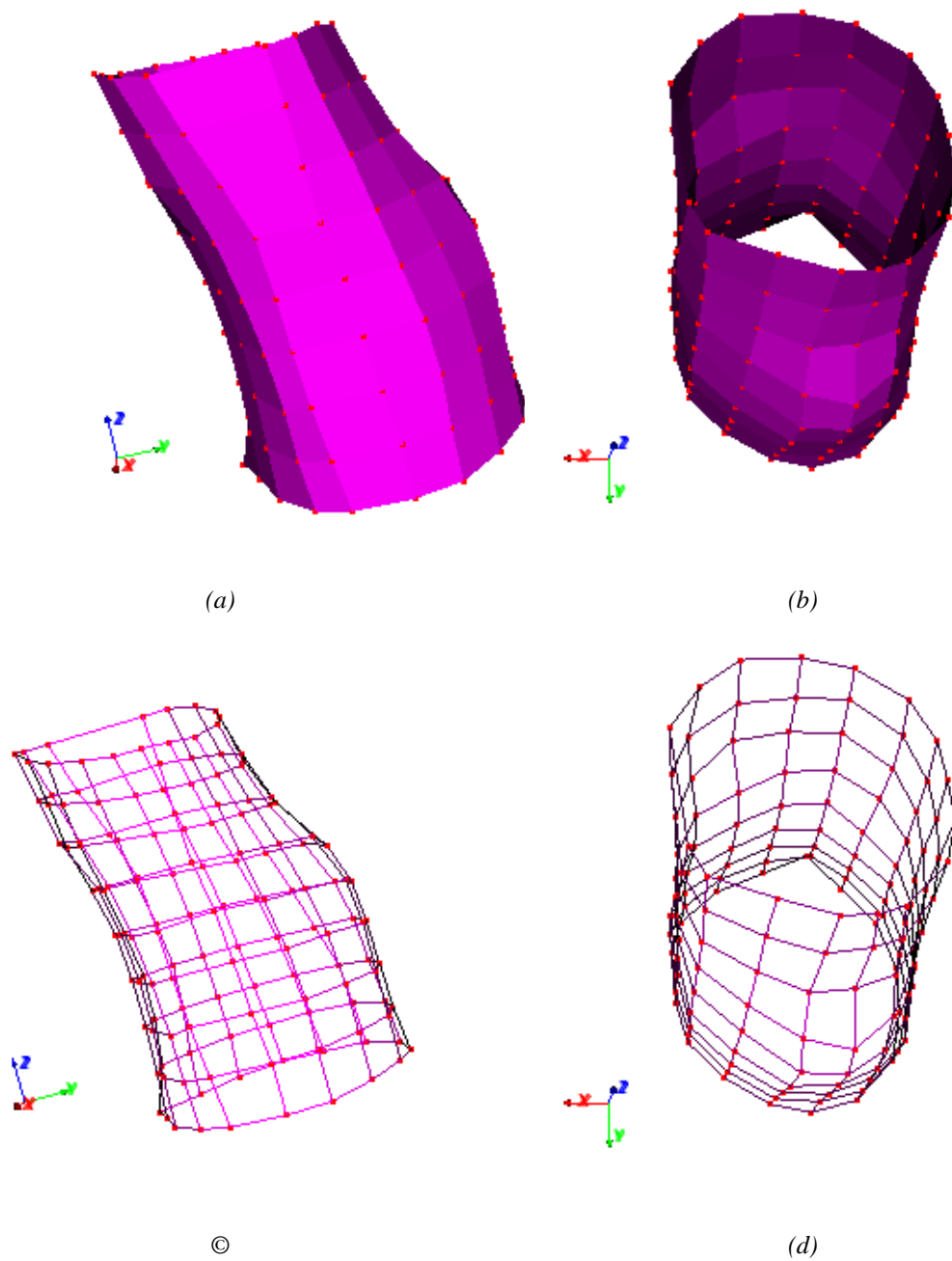


Figura N° 5.10 Superficie deformada, (a) y (b) Superficie deformada a través de Greedy, (c) y (d) Malla de la superficie deformada a través de Greedy.

Simulated Annealing

Para simulated annealing se utilizó como parámetro de reducción de temperatura $\alpha = 0.85$, con 30 iteraciones del algoritmo y 100 repeticiones para la búsqueda de posibles soluciones antes de descender la temperatura, con resultado observable en la figura 5.11.

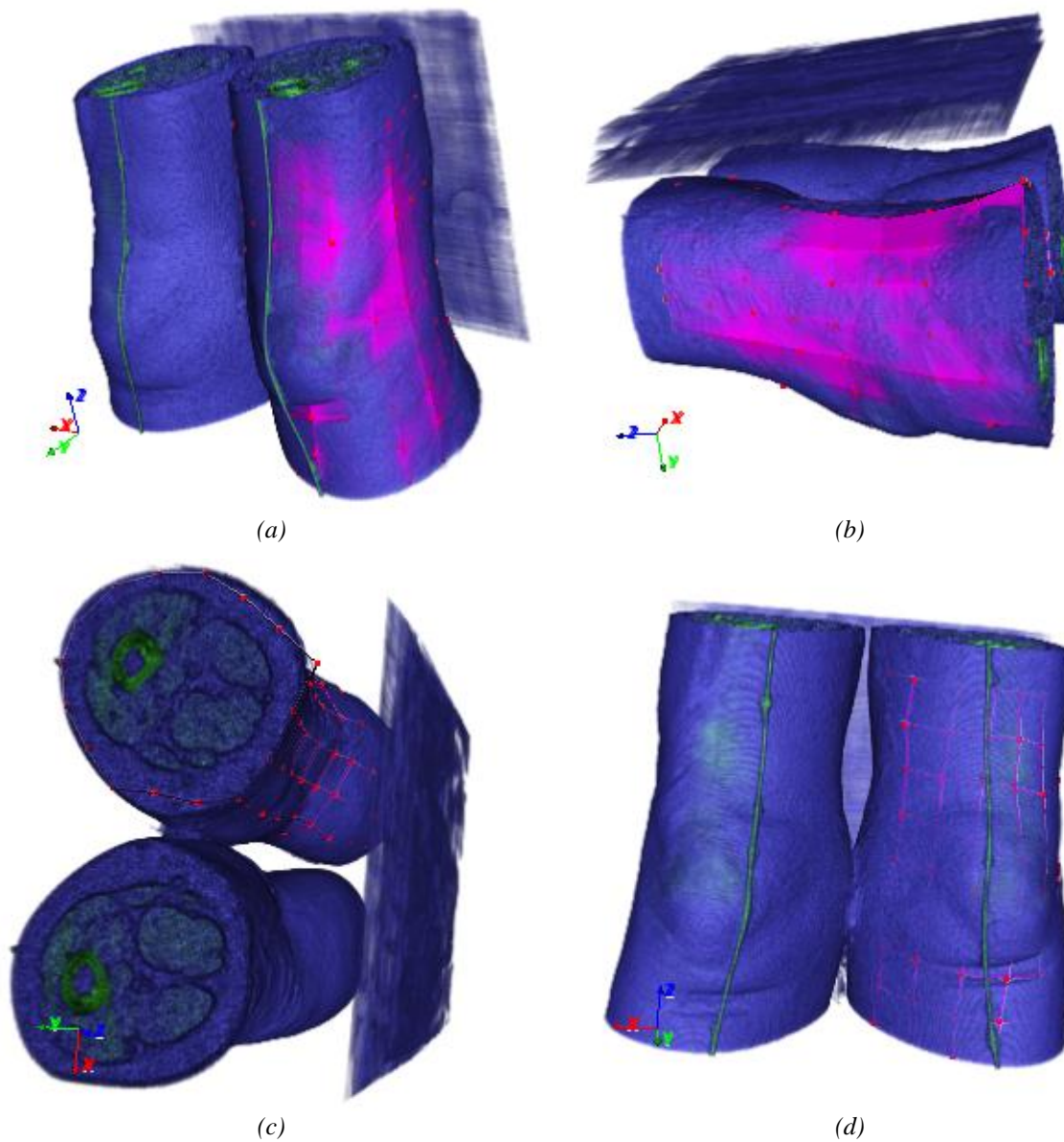


Figura N° 5.11 Superficie deformada y volumen, (a) y (b) Volumen de datos y superficie deformada a través de SA, (c) y (d) volumen de datos y malla de la superficie deformada con SA.

Observamos en más detalle la segmentación, en la figura 5.12 donde se ocultó el volumen de datos.

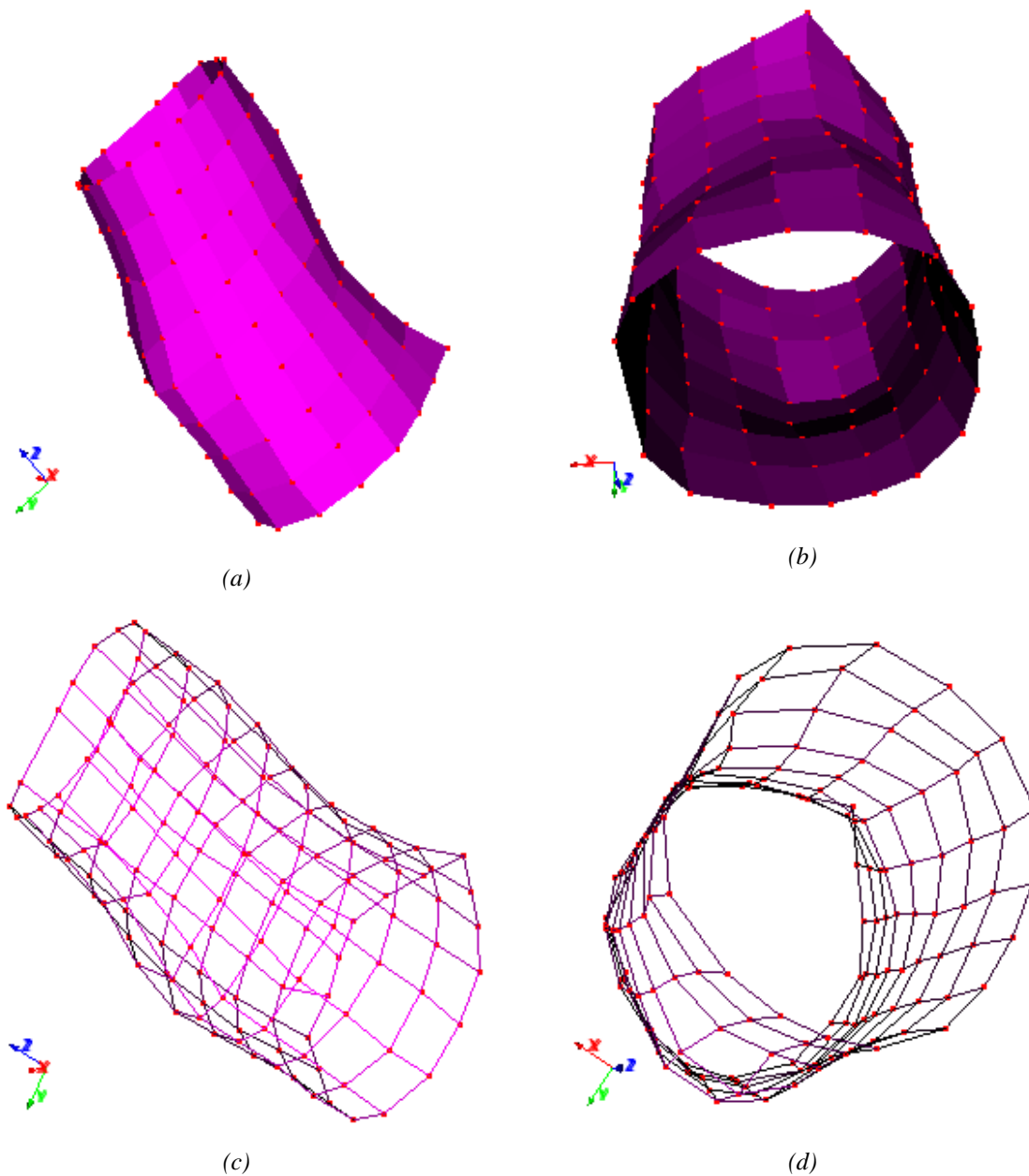


Figura N° 5.12 Superficie deformada, (a) y (b) Superficie deformada a través de SA, (c) y (d) Malla de la superficie deformada a través de SA.

Simulated Annealing Threshold Accepting

En este algoritmo de simulated annealing con threshold accepting se utilizó como umbral (Threshold, T) inicial el valor 0.03, el cual fue descendido con $\mu=0.85$, realizando 30 iteraciones y 100 repeticiones para la búsqueda de posibles soluciones, observando el resultado en la figura 5.13.

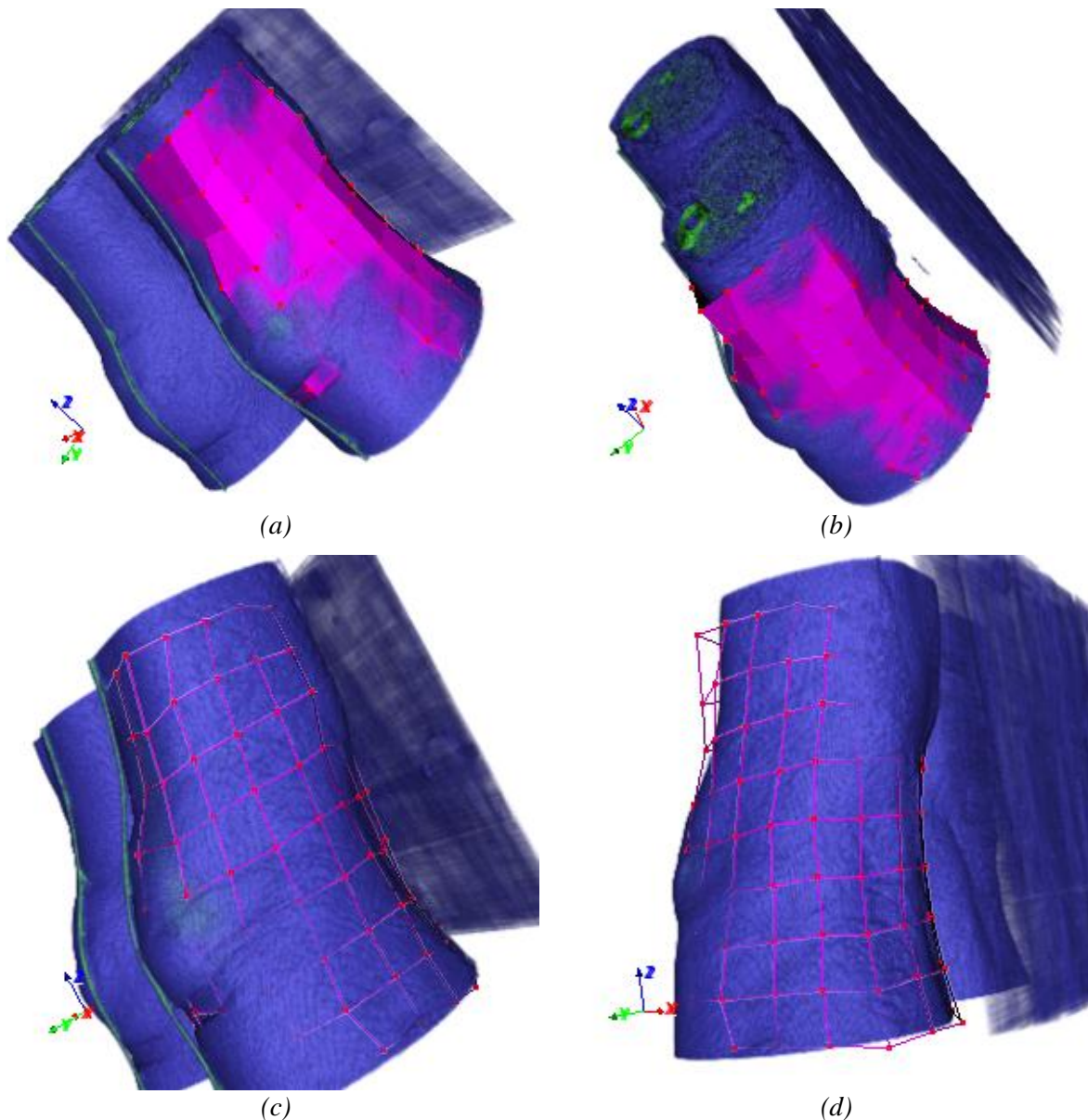


Figura N° 5.13 Superficie deformada y volumen, (a) y (b) Volumen de datos y superficie deformada a través de SA con Threshold Accepting, (c) y (d) volumen de datos y malla de la superficie deformada con SA Threshold Accepting.

La figura 5.14 muestra la superficie final luego de las deformaciones, con el volumen de datos a segmentar oculto.

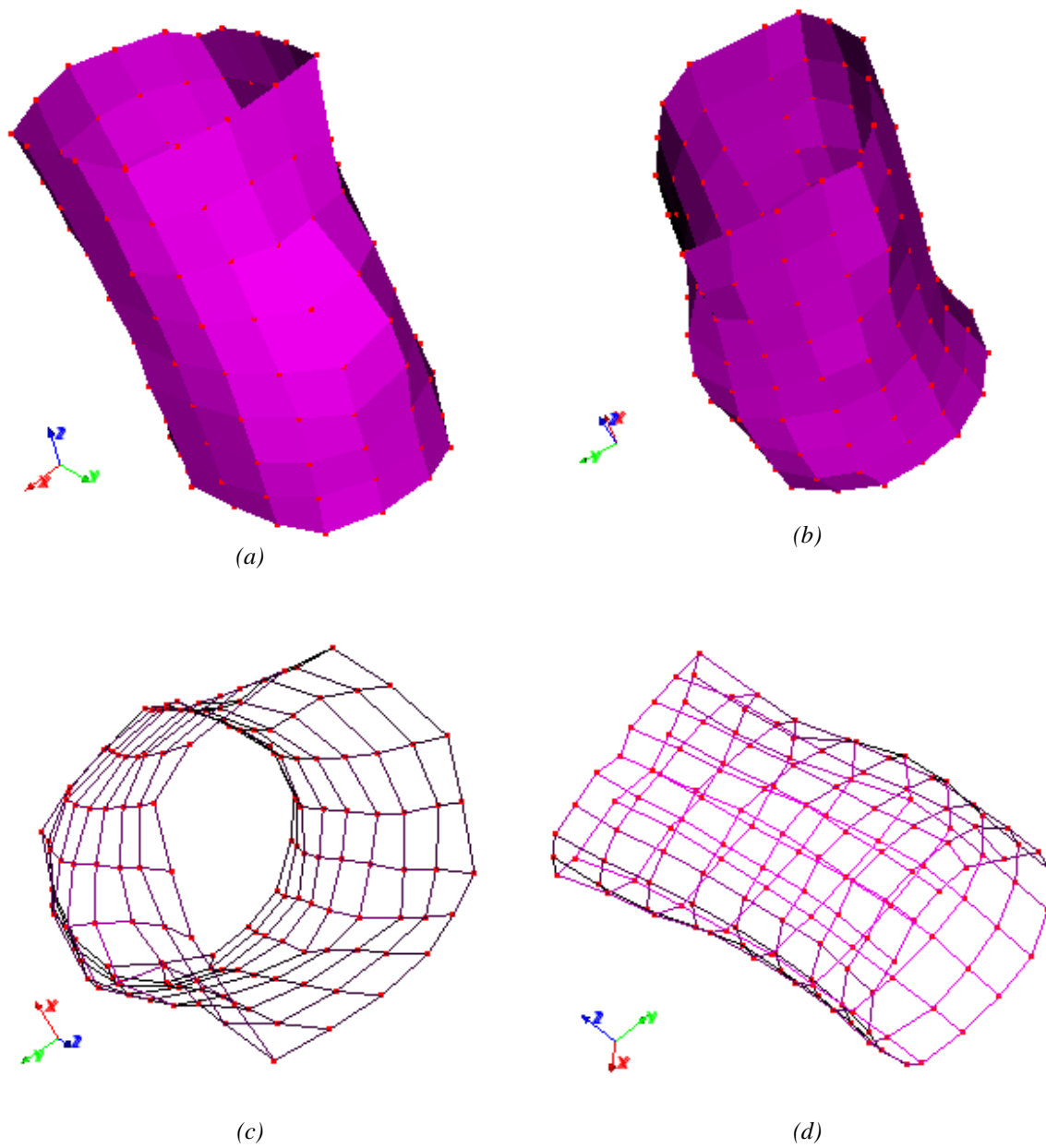


Figura N° 5.14 Superficie deformada, (a) y (b) Superficie deformada con SA Threshold Accepting, (c) y (d) Malla de la superficie deformada con SA. Threshold Accepting.

	Método exhaustivo (Greedy)	Simulated Annealing	Simulated Annealing Threshold
Lectura archivos	2,378	8,611	11,450
Filtro gaussiano	649,828	471,838	579,662
Cálculo magnitud gradiente	3,742	3,603	1,996
Módulo gradiente a matriz	157,228	147,873	139,044
Deformación	0,13	0,312	0,171
Render	13,988	14,103	13,603

Tabla N° 5.2 Resumen tiempos en segundos, Volumen área médica con las diferentes técnicas.

Cambios o grandes diferencias entre las superficies finales no se visualizan como sucedió con el volumen de prueba, salvo SA con Threshold Accepting el cual en la imagen 5.13 (b) y 5.13 (d) muestra que en la parte superior de la rodilla los puntos de control no se movieron de una forma adecuada como sí lo hizo SA en la figura 5.11 (a) (c) o Greedy en la figura 5.9 (a) y 5.9 (c).

En cuanto a tiempos de los algoritmos se ve que SA tiene el mayor tiempo de procesamiento, lo que se comentó en el análisis final de las técnicas en el volumen de prueba y situación a analizar más adelante. Otro aspecto interesante de comentar es el tiempo que le toma SA Threshold Accepting realizar sus cálculos, el cual resulto ser superior que Greedy, lo que contrasta con los resultados en el volumen de prueba. Ésto sucede debido a la característica aleatoria de este proceso que de obtener una energía menor en un punto aleatorio, éste es aceptado inmediatamente. Por lo tanto, se concluye que este tiempo (SA-TA), ya sea mejor o no que Greedy, es impredecible por la aleatoriedad del proceso.

5.3. PRUEBA CON VOLUMEN DE DATOS PROBETA MADERA.

En este último experimento se utilizó una serie de 75 imágenes médicas, adquiridas en el Sanatorio Alemán ubicado en la ciudad de Concepción, las cuales corresponden a un tronco de largo aproximado de 1 metro, el cual fue ingresado a un tomógrafo, gracias a un proyecto interno de la Universidad del Bio Bio.

Las imágenes, si bien se encuentran en formato Dicom, no pudieron ser leídas directamente por la librería VTK, ya que el formato con que se guardó la imagen dentro de los archivos Dicom (JPEG200), no es soportado por la librería. Por ésto, se procedió hacer un proceso previo con un programa de lectura de archivos Dicom, que soportaba la compresión de las imágenes, exportándolas en formato BMP, archivos que se observan en la figura 5.15.

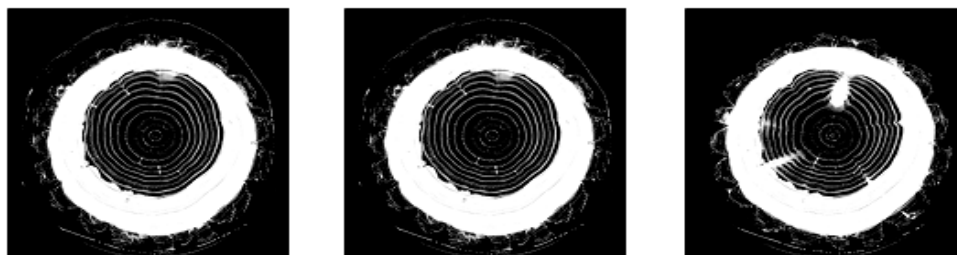


Figura N° 5.15 Imágenes de archivos Dicom madero.

Estas imágenes debieron ser procesadas nuevamente para omitir los anillos de crecimiento que se observan en la figura 5.15, debido a que afectan en gran medida en la evolución de los puntos de control de la superficie deformable, por la fuerza externa que estas intensidades aplican en el funcional de energía. Esto se debe a la formula de energía utilizada, la cual corresponde al método más simple de implementación y cuya solución (mejor movimiento puntos debido a la fuerza externa) se analizará en las conclusiones.

Algunas imágenes de la serie de imágenes pre-procesadas y que finalmente se utilizaran para la reconstrucción tridimensional son mostradas en la figura 5.16, visualizando la reconstrucción en la figura 5.17.



Figura N° 5.16 Imágenes procesadas de archivos Dicom madera.

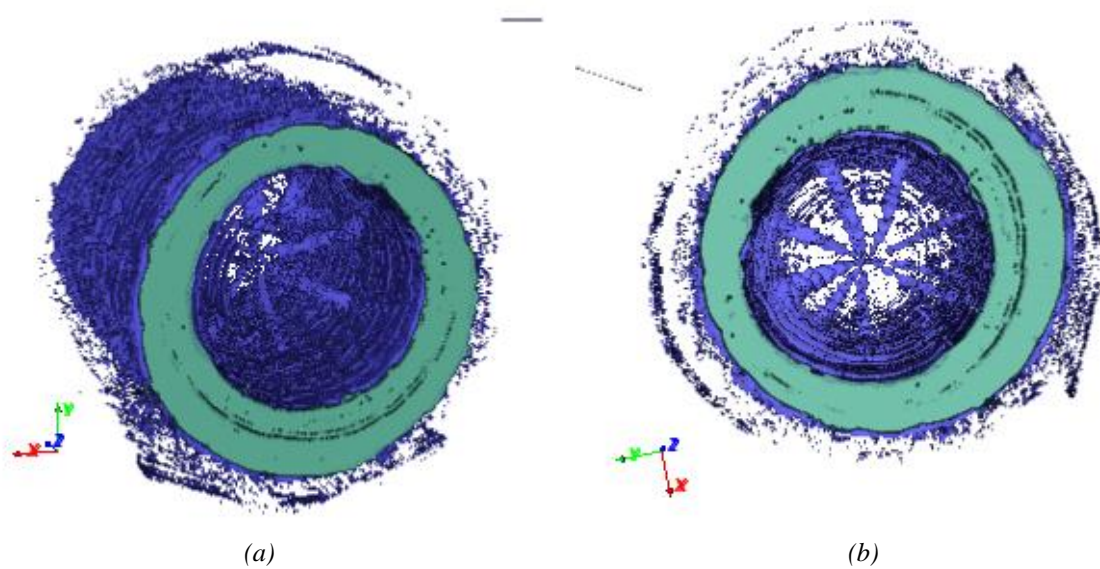


Figura N° 5.17 Reconstrucción tridimensional tronco desde archivos de CT.

La superficie inicial utilizada para la segmentación de un nudo, se obtuvo de sucesivos experimentos de forma manual dando como resultado la superficie mostrada en la figura 5.18.

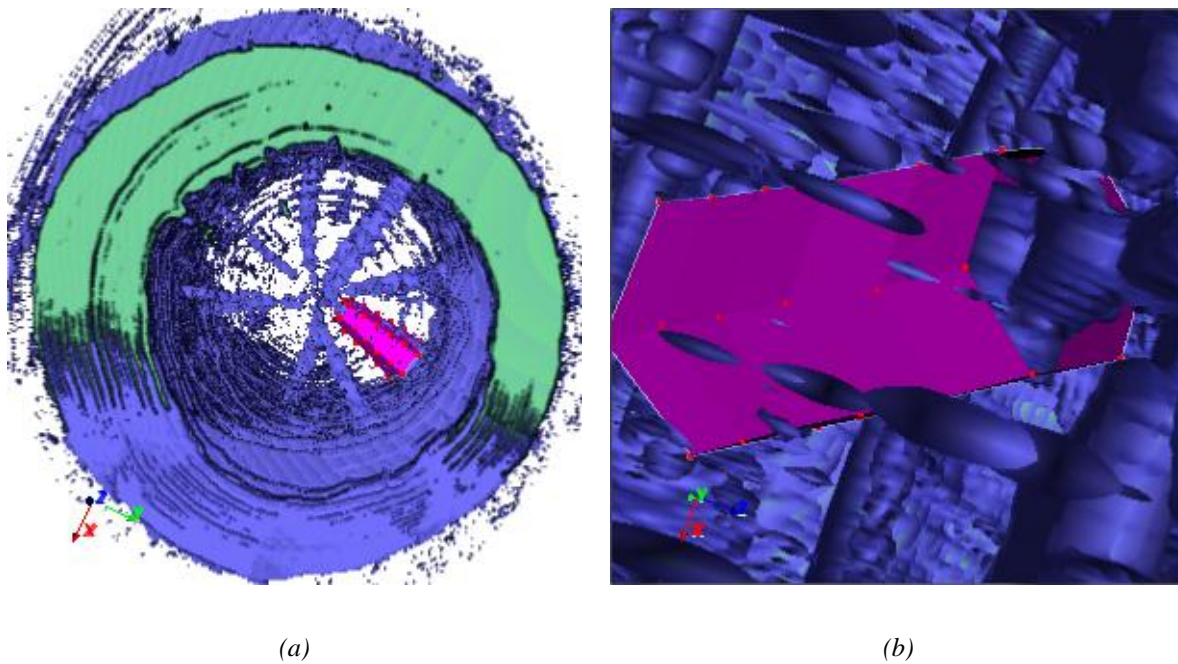


Figura N° 5.18 Superficie inicial, (a) y (b) Superficie inicial a deformar, en nudo a segmentar.

5.3.1. Segmentación de nudo en probeta

Greedy

La búsqueda exhaustiva se realizó 12 veces por cada punto, teniendo un universo de 25 puntos y un volumen de búsqueda de 3x3x3.

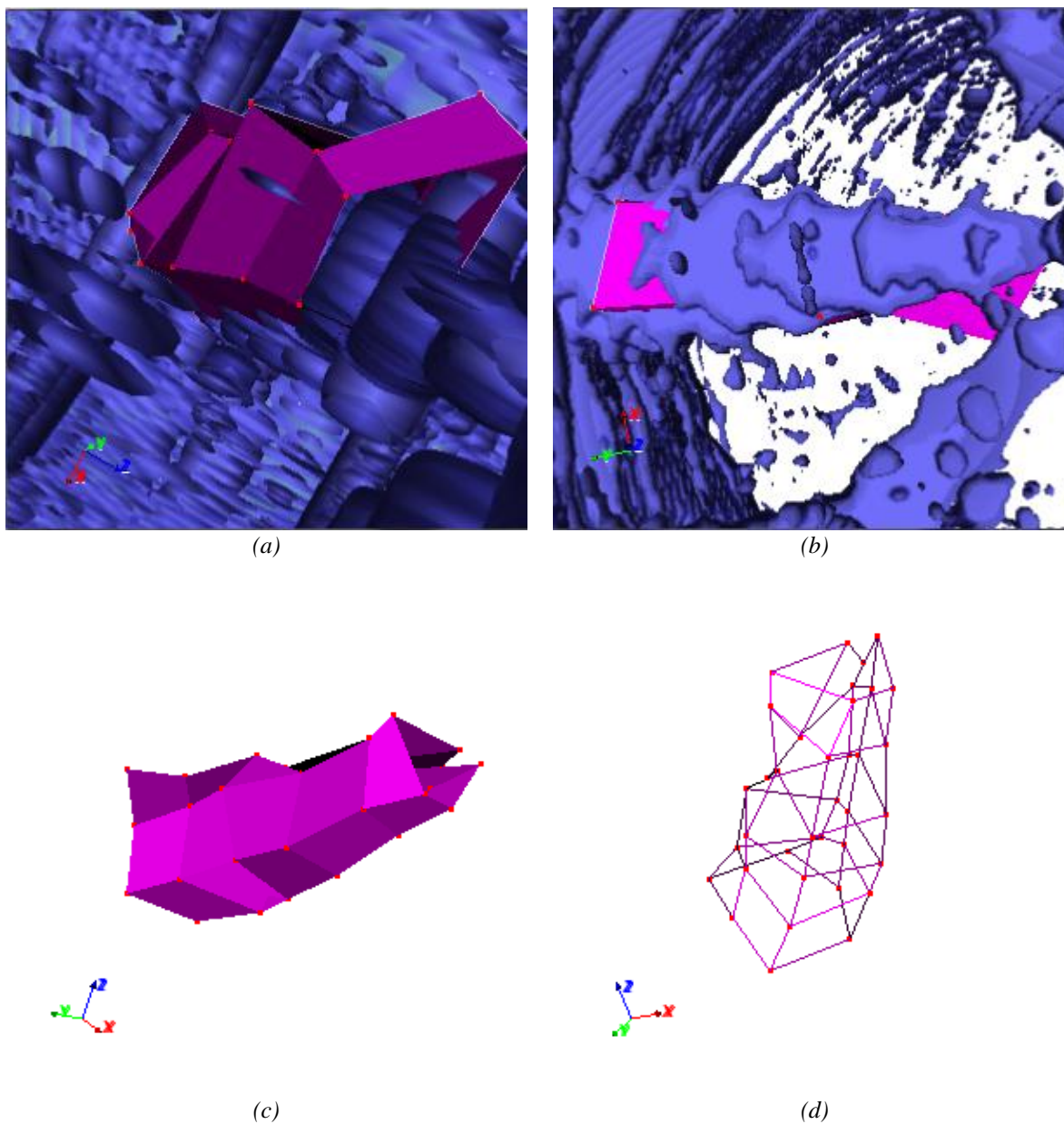


Figura N° 5.19 Segmentación nudo con Greedy, (a) y (b) Volumen de datos y superficie deformada con Greedy, (c)Superficie deformada con Greedy, (d)Malla de superficie deformada.

Simulated Annealing

Este algoritmo tuvo como parámetros 12 iteraciones y 100 repeticiones en la búsqueda de soluciones posibles, con un parámetro α para el descenso de la temperatura igual a 0.85 y una temperatura inicial de 200, lo cual dio buenos resultados como se aprecia en la figura 5.20, después de realizar algunas pruebas.

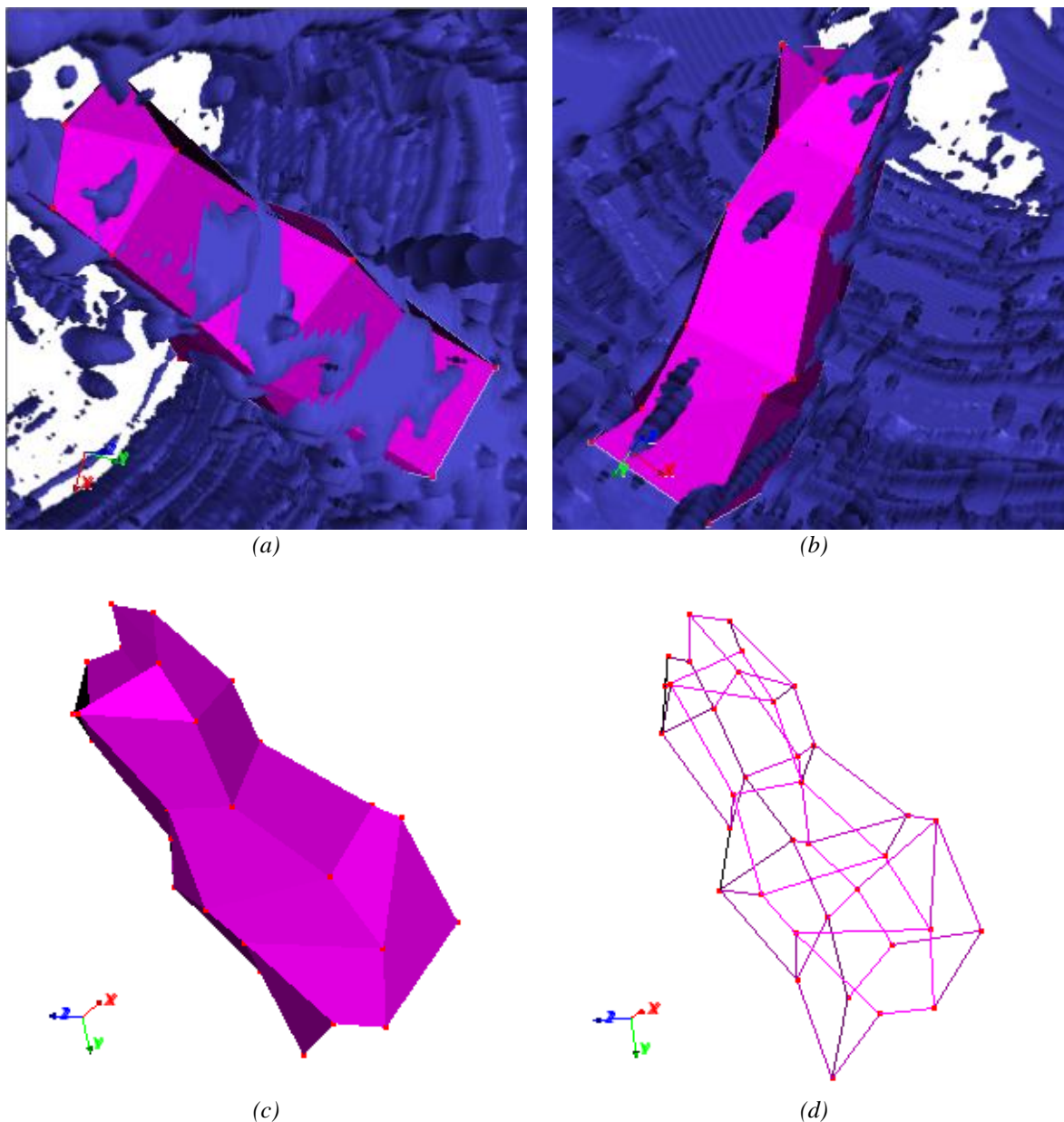


Figura N° 5.20 Segmentación nudo con SA, (a) y (b) Volumen de datos y superficie deformada con SA, (c)Superficie deformada con SA, (d)Malla de superficie deformada.

Simulated Annealing Threshold Accepting

Para la implementación de simulated annealing con threshold accepting se utilizó como umbral (Threshold, T) inicial el valor constante 0.03, el cual fue descendido con $\mu=0.7$, realizando 12 iteraciones y 100 repeticiones para la búsqueda de posibles soluciones. Observando el resultado final en la figura 5.21.

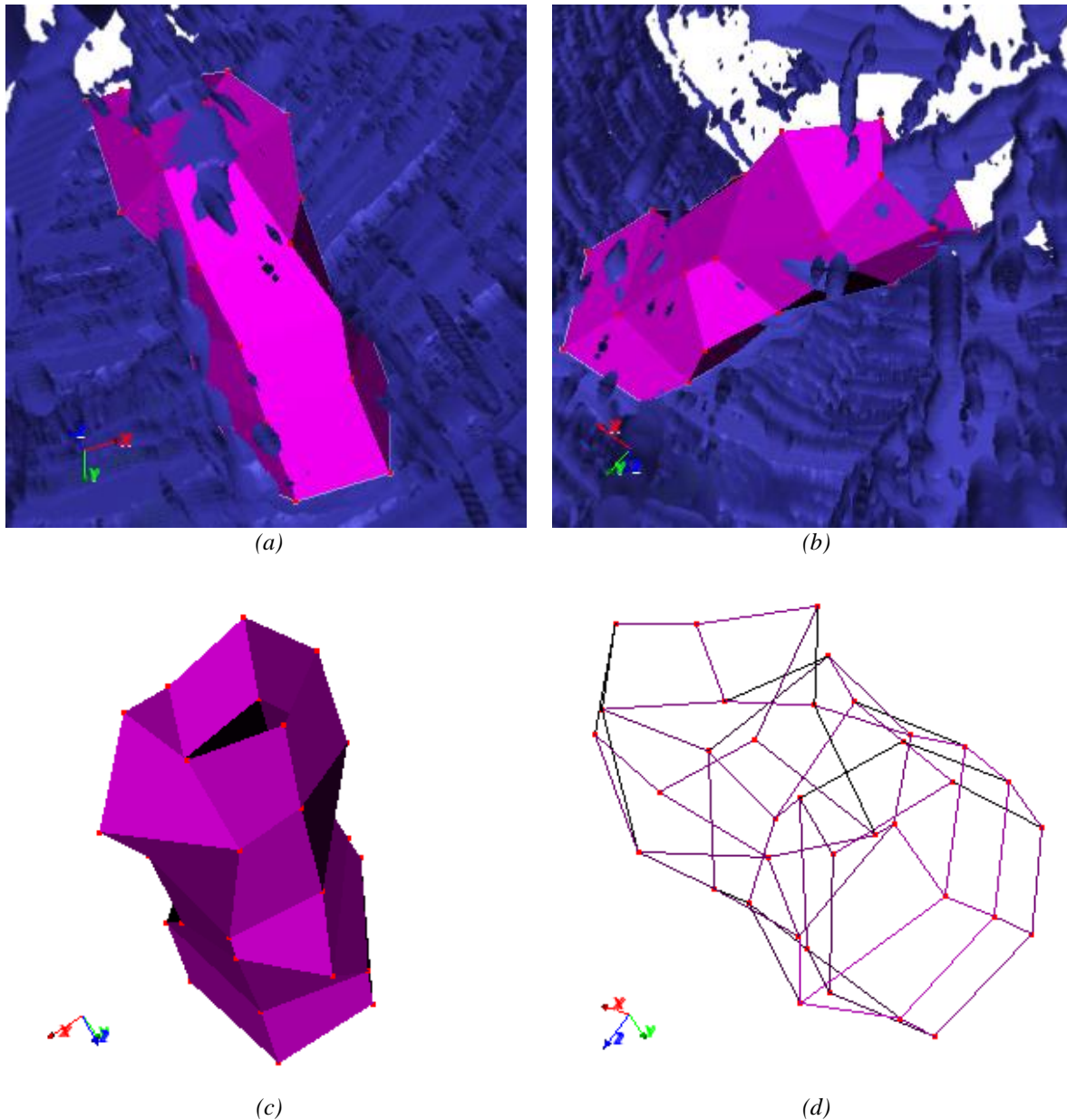


Figura N° 5.21 Segmentación nudo con SA Threshold Accepting, (a) y (b) Volumen de datos y superficie deformada con SA Threshold Accepting, (c)Superficie deformada, (d)Malla de superficie deformada

	Método exhaustivo	Simulated Annealing	Simulated Annealing Threshold
Lectura archivos	22,745	33,883	23,509
Filtro gaussiano	26,957	27,081	26,006
Cálculo magnitud gradiente	2,793	2,792	2,886
Módulo gradiente a matriz	172,674	177,442	174,081
Deformación	0,016	0,046	0,015
Render	12,870	14,134	12,964

Tabla N° 5.3 Resumen de tiempos en segundos, Volumen área maderera con las diferentes técnicas.

Dentro de la calidad de la segmentación se puede mencionar que Simulated Annealing genera una superficie algo más suave que Greedy o SA Threshold Accepting (SA-TA).

Greedy encierra de mala forma el nudo como se aprecia en la figura 5.19 (b) en comparación a las otras dos técnicas en 5.20 (b) y 5.21 (b), SA y SA-TA respectivamente.

Nuevamente, se observa que SA consume una mayor cantidad de tiempo (tabla 5.3) en la segmentación o deformación de la superficie siendo casi el triple que Greedy, caso que se discutirá en el siguiente apartado con la finalidad de explicar a cabalidad la situación.

5.4. COMPARACIÓN DE TÉCNICAS

Como se mencionó en el capítulo 4.2.4, Simulated Annealing es una técnica de búsqueda meta-heurística para problemas con un universo grande de soluciones, encontrando soluciones aproximadas al óptimo global. Esta técnica se aplicó a la superficie deformable para obtener mejores segmentaciones y no un menor tiempo de cálculo. Debido a que el volumen de búsqueda en la implementación realizada es muy pequeño (27, 3x3x3) para la utilización de SA. Volumen que no necesariamente es fijo, el cual puede ser aumentado a una búsqueda en un cubo de 9x9x9, lo cual genera un rango de búsqueda de 729 posiciones que se deben recorrer calculando su energía. Una búsqueda con SA revertirá una buena segmentación y un tiempo de procesamiento menor a una búsqueda exhaustiva con la técnica Greedy.

Para demostrar lo antes expuesto, se tomaron los tiempos en el volumen de prueba (segmentación esfera), buscando la nueva posición de los puntos de control sobre un volumen de 3x3x3 y 9x9x9. Obteniendo los resultados de tiempos (segundos) en las tablas 5.4 y 5.5 respectivamente. Además de las 2 técnicas antes mencionadas, se aplicó la técnica de SA con Threshold Accepting que, si bien busca una aproximación al óptimo global, genera segmentaciones de menor calidad que SA, pero con una performance de tiempo mejor.

	Método exhaustivo (Greedy)	Simulated Annealing	Simulated Annealing Threshold
Lectura archivos	0,07	0,073	0,062
Filtro gaussiano	0,013	0,013	0,031
Cálculo magnitud gradiente	0,003	0,007	0,000
Módulo gradiente a matriz	0,073	0,077	0,078
Deformación	0,057	0,125	0,047
Render	0,329	0,281	0,249

Tabla N° 5.4 Resumen de tiempos en segundos, Volumen de prueba con las diferentes técnicas, búsqueda volumen 3x3x3.

	Método exhaustivo (Greedy)	Simulated Annealing	Simulated Annealing Threshold
Lectura archivos	0,094	0,125	0,109
Filtro gaussiano	0,015	0,015	0,047
Cálculo magnitud gradiente	0,063	0,078	0,094
Módulo gradiente a matriz	0,078	0,078	0,093
Deformación	5,070	1,576	1,498
Render	0,530	0,499	0,499

Tabla N° 5.5 Resumen de tiempos en segundos, Volumen de prueba con las diferentes técnicas, búsqueda volumen 9x9x9.

La Figura 5.22 (a) muestra la gráfica de los tiempos en la búsqueda con un volumen de 3x3x3, en el que se puede apreciar que la técnica SA implica un mayor tiempo de cálculo. Esto se debe a la parametrización de la técnica, la cual itera 15 veces cada punto al igual que Greedy, pero repite el proceso de búsqueda aleatoria 100 veces en comparación a las 27 que realiza Greedy. SA Threshold Accepting al igual que SA itera sobre un punto 15 veces con una búsqueda de posibles nuevas posiciones 100 veces, pero por tener un umbral este proceso se realiza más rápido que SA. La corroboración de lo antes expuesto se ve en la figura 5.22 (b) para la búsqueda en un volumen de 9x9x9 en donde Greedy debe buscar sobre un universo de 727 posibles nuevas posiciones, en comparación a las 100 veces que realiza SA y SA con Threshold Accepting, obteniendo para todos los casos una buena calidad de segmentación en cuanto a la forma final de la superficie deformable.

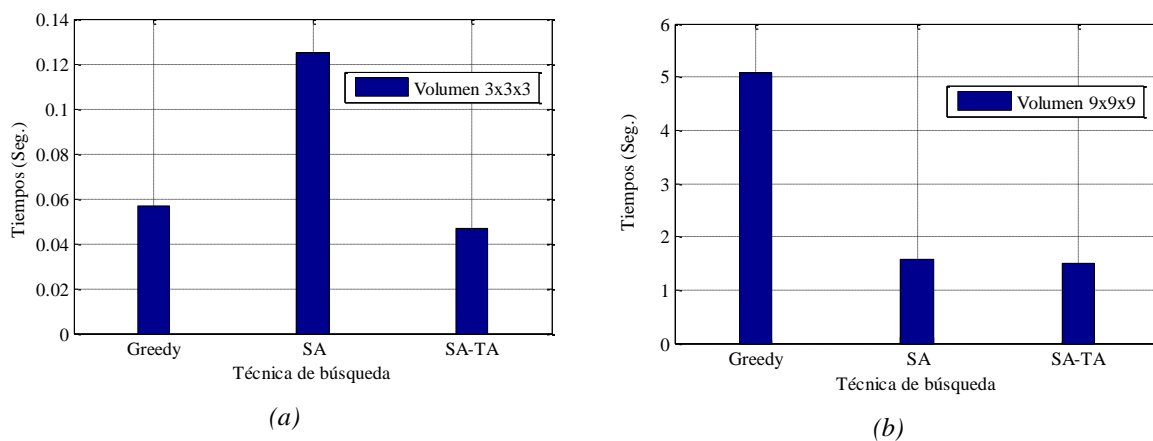


Figura N° 5.22 Gráfico Comparación, (a) Gráfico búsqueda volumen 3x3x3, (b) Gráfico búsqueda volumen 9x9x9.

COMENTARIOS Y CONCLUSIONES

El presente trabajo abordó el procesamiento de volúmenes de datos, logrando extender la técnica de contornos deformables a tres dimensiones creando una superficie paramétrica deformable. Se obtuvieron segmentaciones más suaves comparadas a técnicas con procesamiento de cortes (imágenes 2D) del volumen de datos y su posterior visualización.

Desde un punto de vista tecnológico, un aspecto importante a destacar es la utilización de la librería VTK para la visualización de datos lo cual deja y da a conocer una experiencia relevante para proyectos venideros.

Cabe destacar que si bien se logró capturar (segmentar) un nudo, de varios existentes dentro del tronco por medio de las tomografías, no tuvo la exactitud y el detalle que si tuvo el volumen de prueba y el volumen de área médica. Las imágenes muestran íntegramente el tronco, incluyendo información poco relevante para la aplicación realizada en esta tesis, como lo son los anillos de crecimiento del árbol. Esto se debe en gran medida a la calibración del equipo, el cual esta construido para la captura de imágenes en humanos, debiendo pre procesar las imágenes para omitir esta información poco relevante, que movía los puntos de control de la superficie deformable a zonas erradas. Hecho que demuestra una de las falencias de la técnica implementada, la cual, al tener un objeto cercano al área a segmentar, detiene y no deja avanzar a la superficie deformable hacia el objeto que realmente se pretende aislar, debido a que todos los puntos están unidos entre sí, lo cual mantiene una coherencia de la superficie. Este problema podría ser solucionado a través de información a priori ingresando nuevas energías al funcional.

De las tres técnicas aplicadas; Greedy, Simulated Annealing (SA) y Simulated annealing Threshold Accepting (SA-TA), se puede rescatar que SA revierte segmentaciones y superficies de buena calidad final, pero presenta una pobre performance en tiempo para casos con volumen de búsqueda de $3 \times 3 \times 3$. Por otro lado, SA-TA obtuvo buenos tiempos de cálculo con segmentaciones de una calidad inferior que SA, pero satisfactorias en términos de llegar al borde del objeto.

Es importante analizar las zonas cóncavas, que contenía el nudo en el tronco para el volumen del área de la madera, lo cual implicó una segmentación poco exacta, pudiendo ser mejorada utilizando otras formas de manejo de la energía externa como es el caso Gradient Vector Flow (GVF) que soluciona esta problemática y que junto a un castigo en la energía a formas que salgan de alguna información a priori ingresada al funcional de energía, supondría una técnica estable y robusta para la segmentación de nudos en madera.

Finalmente se puede concluir, que la implementación del algoritmo base de contornos deformables funciona satisfactoriamente, dejando en claro, que es una primera aproximación. Sin embargo éste puede mejorarse en gran medida, como se ha mencionado, añadiendo energías que guíen de mejor forma la superficie, optimizando el código, modelando la superficie deformable de otra manera y utilizando tarjetas gráficas para los numerosos cálculos aminorando los tiempos de procesamiento de grandes volúmenes de datos como fue en el área médica, donde todo el proceso duró 12 minutos aproximadamente.

Bibliografía

- Aguilera C., Sanchez R., Baradit E. (2008-A), Internal wood inspection with active contour using data from ct-scanning.
- Aguilera C., Sanchez R., Baradit E. (2008-B), Detection of knots using X-Ray tomographies and deformable contours with simulated annealing.
- Allevatoa E., Pedarréa R. (2006), Segmentación de imágenes digitales 3d basado en regiones y contornos activos para la generación de mallas de superficie, *Mecánica Computacional*, Vol: 25, pp: 1149-1161.
- Baradit E., Aedo R., Correa J. (2006), Knots detection in wood using microwaves, *Wood science and technology*, Vol: 40, No: 2, pp: 118-123.
- Chan, T.F., Vese, L.A. (2001), Active contours without edges, *IEEE Transactions on Image Processing*, Vol: 10, no: 2, pp: 266-277.
- Cheng J., Xue R., Lu W., Jia R. (2008), Segmentation of medical images with Canny operator and GVF snake model, 7th World congress on intelligent control and automation, pp: 1777-1780.
- Cohen L. y Cohen I.(1991), Finite element methods for active contour models and balloons for 2D and 3D images, *IEEE Transactions on pattern analysis and machine intelligence*, Vol: 15, pp: 1131-1147.
- Fuchs, Theobald, Hanke, Randolph (2008), Task-driven design of X-ray systems for industrial inspection, *IEEE Nuclear Science Symposium Conference Record*, pp:523-527.
- Kass M., Witkin A., Terzopoulos D. (1988), Snakes: Active contour models, *International journal of computer vision*, Vol: 1, N: 4, pp: 321-331.
- Lakare S. (2000), 3D Segmentation Techniques for Medical Volumes.
- Li L., Ren Y., Gong X. (2008), Medical image segmentation based on modified ant colony algorithm with GVF snake model, *International seminar on future biomedical information engineering*, pp: 11-14.
- Nelson T. (2006), Three-dimensional Ultrasound Imaging , *Proceedings of Ultrasound Industry Association Annual Meeting*.

- Neubauer C. (1997), Intelligent X-ray inspection for quality control of solder joints, IEEE Transactions on components, packaging, and manufacturing technology, Vol:20, No:2, pp: 111-120.
- Otsu N. (1979), A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man and Cybernetics, Vol: 9, No: 1, pp: 62-66, 1979.
- Pan L., Gu L., Xu J. (2008), Implementation of medical image segmentation in CUDA, International Conference on technology and applications in biomedicine, pp:82-85.
- Rangayyan R.M., Kamenetsky I., Benediktsson H. (2008), Segmentation and analysis of the glomerular basement membrane using active contour models, 4th IET International conference on advances in medical, signal and information processing, pp:1-4.
- SHEN C., FAN J., PI L., LI F. (2006), Delineating lakes and enclosed islands in satellite imagery by geodesic active contour model, International journal of remote sensing Vol: 27, No: 23.
- Teramoto A., Murakoshi T., Tsuzaka M., Fujita H. (2007), Automated solder inspection method by means of x-ray oblique computed tomography, IEEE International conference on image processing, Vol: 5, pp:433-436.
- Terzopoulos D. y Metaxas D. (1991). Dynamic 3D models with local and global deformations: Deformable superquadrics. Pattern analysis and machine intelligence, pp: 703-714.
- Terzopoulos D., Vasilescu M. (1991), Sampling and reconstruction with adaptive meshes, IEEE computer society conference on computer vision and pattern recognition, Vol: 3, No: 6, pp.70-75.
- Tseng C., Jeng J. y Hsieh G. (2009), Active contour model via multi-population particle swarm optimization, Vol: 36, No: 3.
- Xu G. y Prince J. (1998), Snakes, shapes, and gradient vector flow, IEEE transactions on image processing, Vol: 7, No: 3, pp: 359-369.
- Xiaoping M., Boliang W., Shaohui H., Peishan D. (2008), Parametric shape modeling using deformable superellipses for segmentation of medical image, The 2nd international conference on bioinformatics and biomedical engineering, pp. 2673-2676.

Yim P., Foran D. (2003), Volumetry of hepatic metastases in computed tomography using the watershed and active contour algorithms, IEEE Symposium on computer-based medical systems, pp: 329-335.

Ying Z., Guangyao L., Xiehua S. (2008), Medical image segmentation based on wavelet transformation and IGGVF, The 2nd international conference on bioinformatics and biomedical engineering, pp: 2508-2511.

ANEXOS.

Índice.

- A) INSTALACION LIBRERÍA VTK
- B) UTILIZACION LIBRERÍA

ANEXOS A: Instalación VTK

Se debe tener el código pre-compilado de las Librerías VTK, esto se puede obtener de la página oficial <http://www.vtk.org> en la sección RESOURCES->Download. Luego se realiza la descompresión en una carpeta que identificaremos como source, creando al lado de esta una carpeta bin donde se dejara las librerías compiladas.

Aparte de VTK es necesario descargar Cmake desde <http://www.cmake.org/> el cual se instala en el computador a trabajar, ya sea con sistema operativo basado en Windows como en Linux.

Ejecutamos Cmake, para luego decirle donde se encuentra la fuente de las librerías y donde se construirá las librería a compilar, seguido presionamos Configure debiendo seleccionar el compilador a usar, en el caso que se muestra en el anexo se utilizo Visual Studio 2008 y luego Finish, de estar todo bien dependiendo lo que nos muestre el programa en la ventana de salida del compilador.

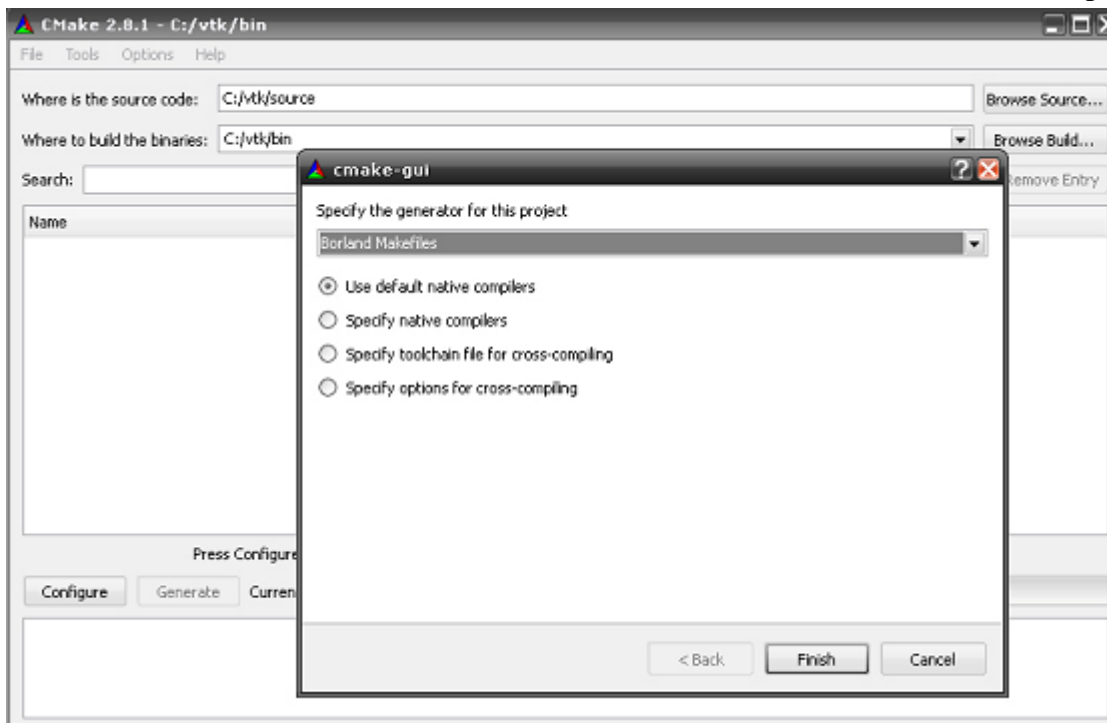


Figura N° A.1 Pantalla Cmake.

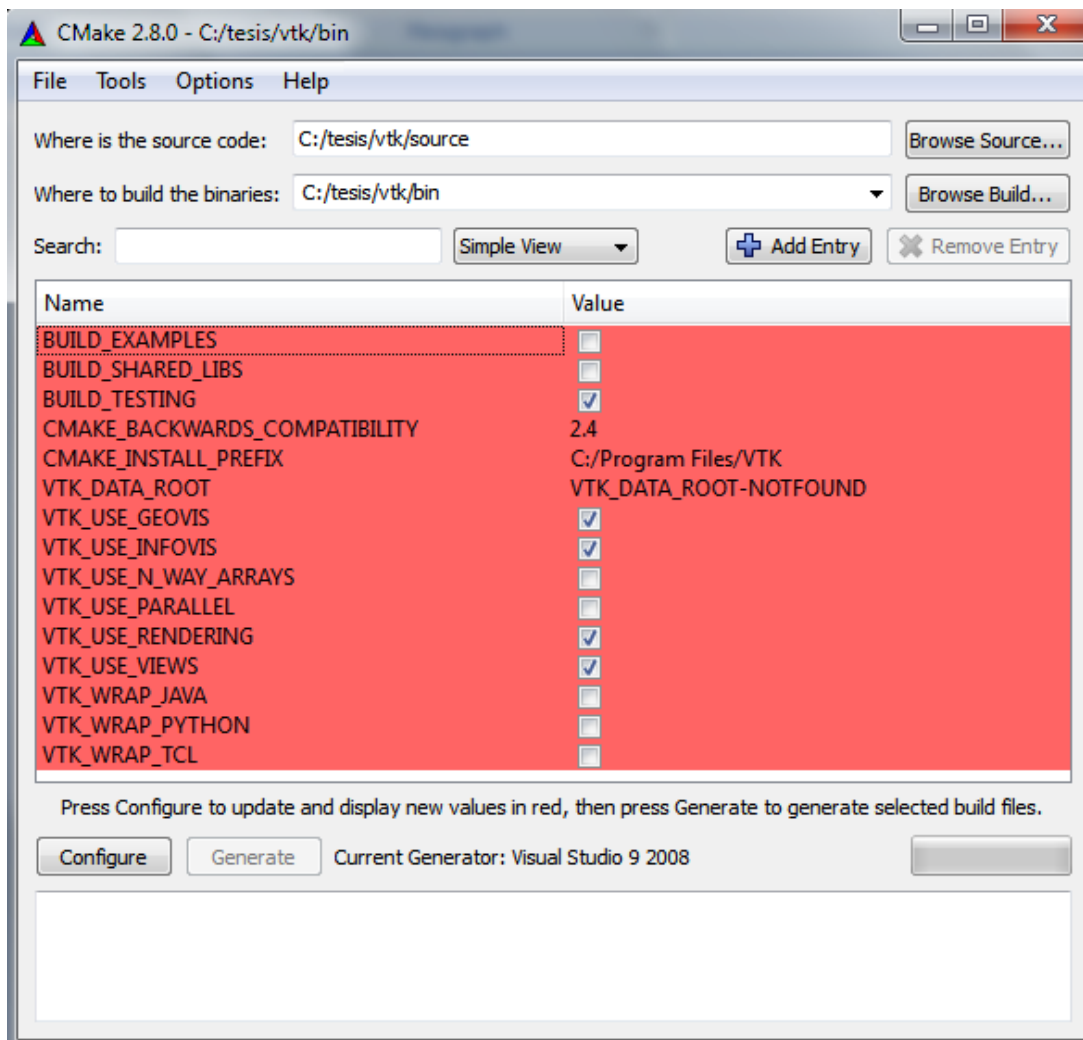


Figura N° A.2 Configuración parámetros de instalación de VTK .

Aquí seleccionaremos algunas configuraciones, como son procesamiento paralelo con tarjetas graficas, o compilar de inmediato los ejemplos que posee además de varias configuraciones posibles cambiando la vista de las opciones al presionar Simple View. En el caso de instalación que se seguirá seleccionaremos VTK_USE_RENDERING y VTK_USE_VIEWS y desmarcando BUIL_EXAMPLES, BUILD_SHARED_LIBS y BUILD_TESTING, para luego presionamos Generate.

Esto generara en la carpeta bin el código a compilar, en el cual abriremos el proyecto de Visual Basic, ALL_BUILD.vcproj, para luego compilar la solución como se muestra en la figura

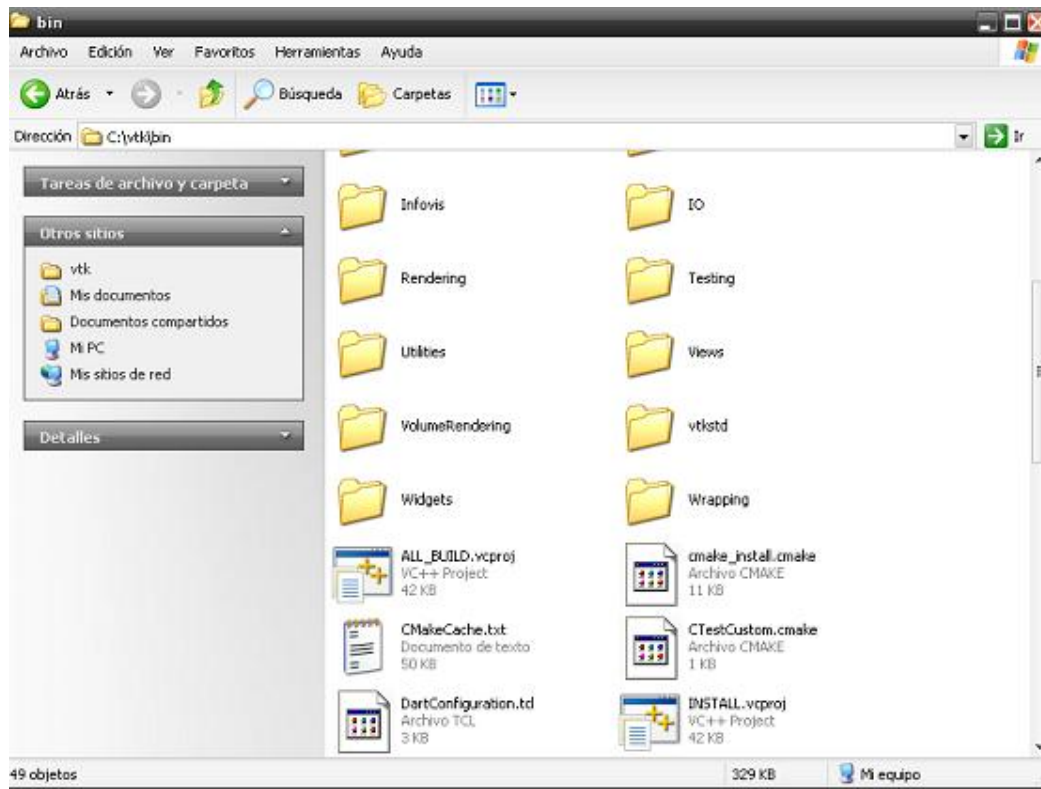


Figura N° A.3 Archivos generados por Cmake.

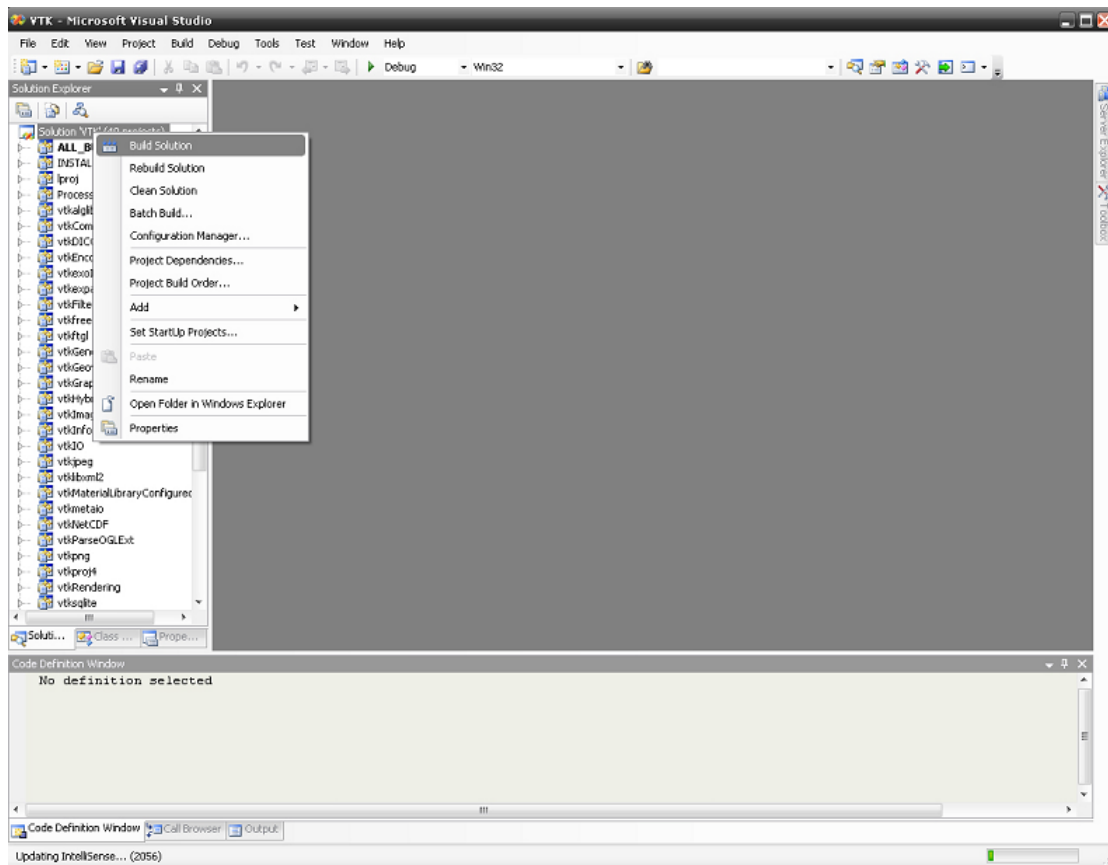


Figura N° A.4 Entorno Visual Studio y compilación librería VTK.

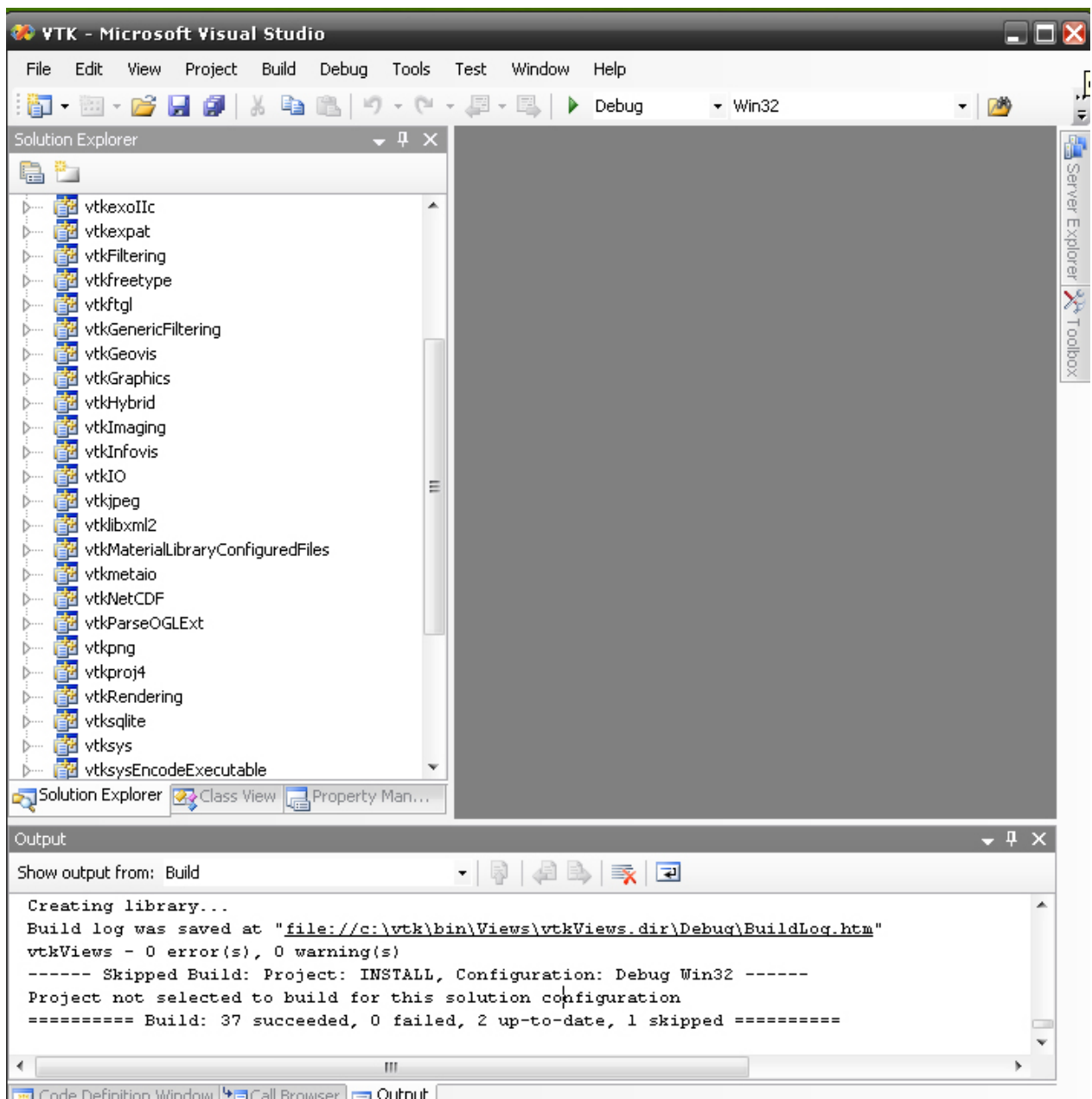


Figura N° A.5 Resultado fin compilación VTK.

Luego de esperar varios minutos, tendremos compiladas las librerías de VTK.

ANEXOS B: Utilización Librería

Al igual que las librerías VTK, el código a compilar con un programa ejemplo, es necesario que sea utilizado Cmake.

Un proyecto como el ejemplo que veremos a continuación, necesita de un CmakeList.txt el cual se encarga de crear el proyecto describiendo las librerías que necesitaremos y que buscare en el sistema para adjuntarlas a nuestro proyecto en el momento de la compilación.

Ejemplo código CmakeList.txt

```
CMAKE_MINIMUM_REQUIRED(VERSION 2.4)
PROJECT(ejemplo_cilindro)

FIND_PACKAGE ( VTK)
IF ( VTK_FOUND)
INCLUDE( ${USE_VTK_FILE} )
ENDIF( VTK_FOUND)

ADD_EXECUTABLE(cilindro Cylinder.cxx )

TARGET_LINK_LIBRARIES (cilindro
vtkRendering vtkVolumeRendering vtkGraphics vtkHybrid
vtkImaging vtkIO vtkFiltering vtkCommon)
```

Este a su vez necesita el código fuente del programa. El cual se encuentra en Cylinder.cxx y que es un ejemplo que da Kitware, los desarrolladores de VTK. Este programa creara un cilindro de color rojo, el cual será posible observar a través de una ventana de renderizado con controles en el mouse para observarlo a su alrededor.

Codigo ejemplo Cylinder.cxx


```
#include "vtkCylinderSource.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkRenderWindow.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkProperty.h"
#include "vtkCamera.h"

int main()
{
    // This creates a polygonal cylinder model with eight circumferential
    facets.
    //
    vtkCylinderSource *cylinder = vtkCylinderSource::New();
    cylinder->SetResolution(8);

    // The mapper is responsible for pushing the geometry into the graphics
    // library. It may also do color mapping, if scalars or other
    attributes
    // are defined.
    //
    vtkPolyDataMapper *cylinderMapper = vtkPolyDataMapper::New();
    cylinderMapper->SetInputConnection(cylinder->GetOutputPort());

    // The actor is a grouping mechanism: besides the geometry (mapper), it
    // also has a property, transformation matrix, and/or texture map.
    // Here we set its color and rotate it -22.5 degrees.
    vtkActor *cylinderActor = vtkActor::New();
    cylinderActor->SetMapper(cylinderMapper);
    cylinderActor->GetProperty()->SetColor(1.0000, 0.3882, 0.2784);
    cylinderActor->RotateX(30.0);
    cylinderActor->RotateY(-45.0);

    // Create the graphics structure. The renderer renders into the
    // render window. The render window interactor captures mouse events
    // and will perform appropriate camera or actor manipulation
    // depending on the nature of the events.
    //
    vtkRenderer *ren1 = vtkRenderer::New();
    vtkRenderWindow *renWin = vtkRenderWindow::New();
    renWin->AddRenderer(ren1);
    vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);

    // Add the actors to the renderer, set the background and size
    //
    ren1->AddActor(cylinderActor);
    ren1->SetBackground(0.1, 0.2, 0.4);
    renWin->SetSize(200, 200);
```

```

// We'll zoom in a little by accessing the camera and invoking a
"Zoom"
// method on it.
ren1->ResetCamera();
ren1->GetActiveCamera()->Zoom(1.5);
renWin->Render();

// This starts the event loop and as a side effect causes an initial
render.
iren->Start();

// Exiting from here, we have to delete all the instances that
// have been created.
cylinder->Delete();
cylinderMapper->Delete();
cylinderActor->Delete();
ren1->Delete();
renWin->Delete();
iren->Delete();

return 0;
}

```

Por lo tanto teniendo en una carpeta nuestros archivos iniciales, se debe crear una sub carpeta donde Cmake creara el proyecto

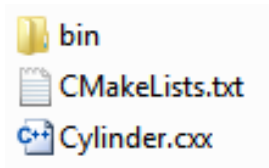


Figura N° B.1 Archivos necesarios para crear proyecto a través de Cmake.

Ejecutando Cmake configurando y generando el proyecto obtenemos algo como

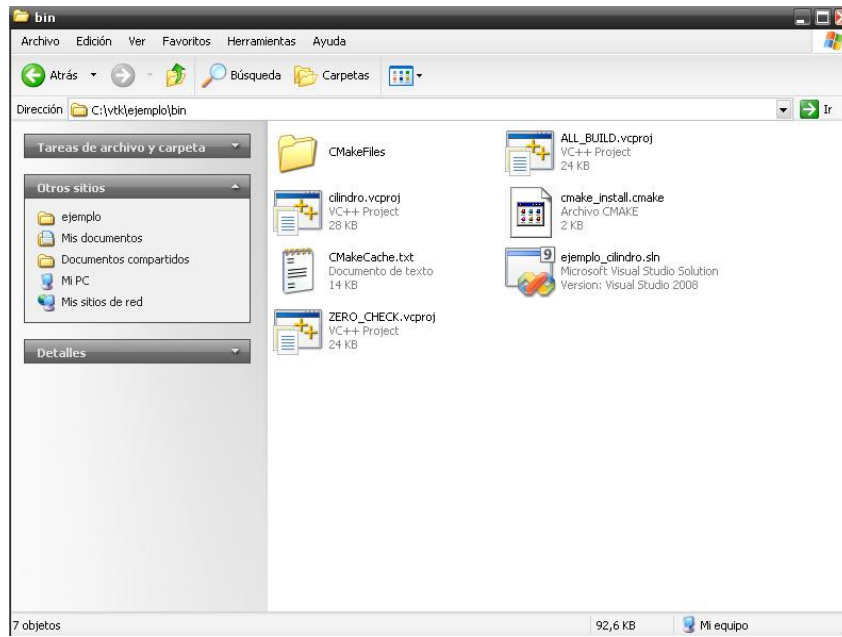


Figura N° B.2 Archivos generados por Cmake del proyecto en Visual Studio.

Para finalmente compilar y obtener el programa final.

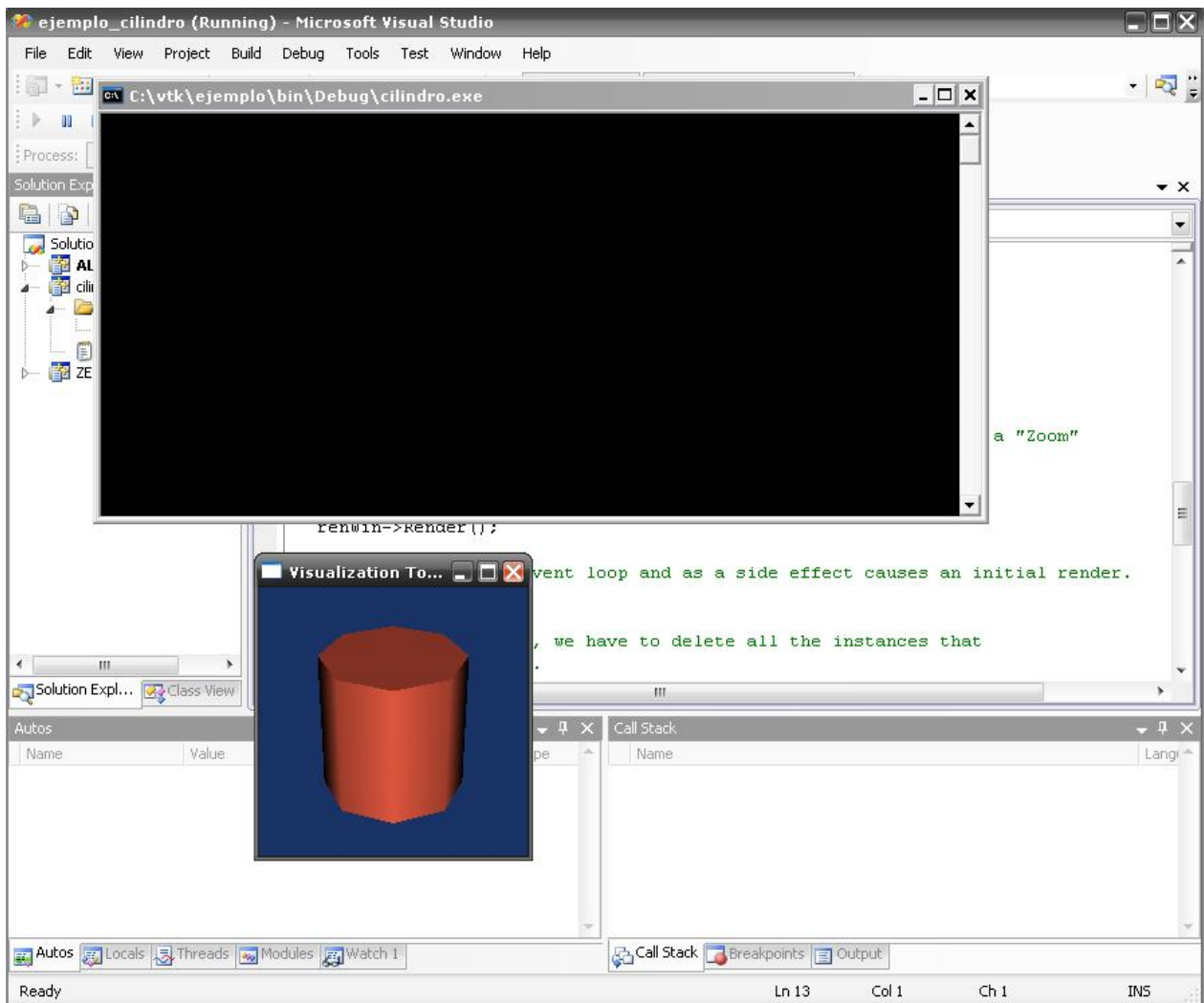


Figura N° B.3 Ventanas de resultado de compilación del ejemplo.

NOTA: Para poder realizar debug de los programas, es necesario compilar las librerías en modo debug (instalación librerías), para ejecutar los programas como release al momento de compilar las librerías se debe hacer en modo release.