Modelado 3D de tumores cerebrales empleando endoneurosonografía y redes neuronales artificiales

Andrés Felipe Serna Morales

Universidad Nacional de Colombia
Departamento de Ingeniería Eléctrica, Electrónica y Computación
Universidad Nacional de Colombia, Sede Manizales
Manizales, Colombia
2011

Modelado 3D de tumores cerebrales empleando endoneurosonografía y redes neuronales artificiales

Andrés Felipe Serna Morales

Tesis presentada como requisito parcial para optar al título de:
Magíster en Ingeniería - Automatización Industrial

Director:
Flavio Augusto Prieto Ortiz, PhD

Línea de Investigación:
Visión por Computador, Modelado 3D, Redes Neuronales
Grupo de Investigación:
Grupo de Percepción y Control Inteligente (PCI)

Departamento de Ingeniería Eléctrica, Electrónica y Computación
Universidad Nacional de Colombia, Sede Manizales
Manizales, Colombia
2011

3D modeling of brain tumors using endoneurosonography and artificial neural networks

Andrés Felipe Serna Morales

A Thesis presented for the degree of: :
Master of Engineering - Industrial Automation

Supervisor:
Flavio Augusto Prieto Ortiz, PhD

Research Areas::
Computer vision, 3D Modeling, Neural Networks
Research group:
Perception and Intelligent Control Group (PCI)

Department of Electrical, Electronic and Computer Engineering
Universidad Nacional de Colombia at Manizales
Manizales, Colombia
2011

A mis padres, José Jaime y Melva Irene, por su infinito amor, su apoyo incondicional y gran sabiduría para indicarme siempre el camino correcto.

A mi hermano, David, por su apoyo y compañía, por cuidar de la familia en los momentos en que no estuve presente.

A toda mi familia, por sus invaluables muestras de apoyo y cariño que me dieron la fuerza para continuar en los momentos difíciles.

A las personas maravillosas que conocí en México, quienes hicieron de mi estancia en ese país un punto de inflexión en mi experiencia de vida.

A mis amigos en la Universidad Nacional de Colombia y en el CINVESTAV Guadalajara, por su amistad, palabras de aliento, compañía, paciencia e invitaciones a jugar fútbol.

La carga más pesada es por lo tanto, a la vez, la imagen de la más intensa plenitud de la vida. Cuanto más pesada sea la carga, más a ras de tierra estará nuestra vida, más real y verdadera será.
Por el contrario, la ausencia absoluta de carga hace que el hombre se vuelva más ligero que el aire, vuele hacia lo alto, se distancie de la tierra, de su ser terreno, que sea real solo a medias y sus movimientos sean tan libres como insignificantes. Entonces, ¿Qué debemos elegir? ¿El peso o la levedad?

— *MILAN KUNDERA*

Agradecimientos

A mi Director, el Profesor Flavio Augusto Prieto Ortiz, por sus invaluables contribuciones y su incondicional apoyo durante el desarrollo de esta investigación.

A los profesores del CINVESTAV Guadalajara, Eduardo Bayro Corrochano y Edgar Nelson Sánchez Camperos, por sus valiosas asesorías durante mis estancias académicas en México.

Al estudiante de Ingeniería Mecatrónica de la sede Bogotá, Luis Felipe Montoya Franco, por su valiosa colaboración en el diseño e implementación de los algoritmos acelerados y otras pruebas experimentales.

Al Programa de Becas para Estudiantes Sobresalientes de Posgrado de la Universidad Nacional de Colombia, por financiar mis estudios de maestría durante estos 2 años.

Al programa de cooperación internacional entre Colciencias–Colombia y Conacyt–México, por apoyar económicamente mis estancias académicas en el Centro de Investigación y de Estudios Avanzados del IPN, CINVESTAV Unidad Guadalajara (México), bajo los Proyectos 460 y 483 de 2009: "Desarrollo e Implementación de herramientas robustas para la visualización y seguimiento en tiempo real de estructuras del cerebro empleando imágenes de ultrasonido 2D y endoscopía".

A la Dirección de Investigaciones de la Universidad Nacional de Colombia Sede Manizales, por su apoyo financiero bajo el proyecto DIMA 20201006025: "Modelado 3D de tumores cerebrales empleando endoneurosonografía y redes neuronales artificiales".

Resumen

Las cirugías mínimamente invasivas no precisan de heridas importantes para acceder a la zona del organismo que necesita ser operada, por lo que se han vuelto muy populares debido a que reducen los riesgos típicos de una intervención tradicional.
En el caso de las neurocirugías, las tendencias recientes sugieren la utilización conjunta de endoscopía y ultrasonido, una técnica llamada Endoneurosonografía (ENS), como herramienta de visualización en tiempo real de estructuras internas del cerebro, como tumores. La información adquirida mediante esta técnica debe ser utilizada para generar un modelo de representación 3D de tumores que se actualice durante la cirugía.
En este sentido, el modelado 3D con redes neuronales ofrece una ventaja, ya que la representación puede desarrollarse en dos etapas: primero, una etapa de entrenamiento fuera de línea, que permite generar un modelo inicial del tumor a partir de información ENS preliminar. Segundo, una etapa de entrenamiento en línea, que permite ajustar el modelo de acuerdo a la nueva información ENS recibida.
En este trabajo de investigación, se presenta una metodología de modelado 3D de tumores cerebrales empleando endoneurosonografía y redes neuronales artificiales. Específicamente, se analiza el uso de redes neuronales multicapa alimentadas hacia adelante (MLFFNN), mapas auto–organizados (SOM) y redes neuronales gas (NGN). Se realizan pruebas experimentales de modelado 3D con objetos de realidad virtualizada como tumores cerebrales fantasma, piezas arqueológicas, rostros, frutas y botellas. Las tres arquitecturas neuronales son comparadas en términos de desempeño, costo computacional, tiempo de procesamiento, conveniencia de la visualización y número de épocas de entrenamiento. Finalmente, se presentan resultados de modelado 3D de tumores sobre una base de datos ENS propia.

Palabras clave: Endoneurosonografía, endoscopía, ultrasonido, adquisición 3D, modelado 3D, redes neuronales

Abstract

Minimally invasive surgeries do not require significant injuries for accessing the area of the body that needs to be operated, so they have become very popular because they reduce the typical risks with respect to traditional interventions.
In neurosurgery, recent trends suggest the joint use of endoscopy and ultrasound, a technique called Endoneurosonography (ENS), for real–time visualization of brain structures, such as tumors. The ENS information can be used to generate a 3D representation model of brain tumors during surgery.
In this sense, 3D modeling using neural networks offers an advantage, since the representation can be developed in two stages: the first one, an off–line training, which generates an initial model of the tumor from preliminary ENS information. The second one, an on–line training, which adjusts the model according to the new information received from the ENS equipment.
In this research, a methodology for 3D modeling of brain tumors using endoneurosonography and artificial neural networks is presented. Specifically, the use of multi–layer feed–forward neural networks (MLFFNN), self–organizing maps (SOM) and neural gas networks (NGN) is studied. Experimental tests of 3D modeling are performed using virtualized reality objects as "phantom" brain tumors, archaeological pieces, faces, fruits and bottles. The three neural architectures are compared in terms of performance, computational cost, processing time, displaying convenience and number of training epochs. Finally, the results of 3D modeling of brain tumors from an ENS database are presented.

Keywords: Endoneurosonography, endoscopy, ultrasound, 3D acquisition, 3D modeling, neural networks

Contenido

Portada I

Agradecimientos XI

Resumen XIII

Abstract XV

Lista de Figuras XIX

Lista de Tablas XXIII

1. Introducción 1

2. Revisión de la literatura 5

2.1. Estado del arte: imagenología médica 5

2.2. Estado del arte: modelado 3D 7

3. Adquisición de información 3D de estructuras cerebrales a partir de ENS 21

3.1. Sistema endoneurosonográfico 25

3.2. Calibración de cámaras 25

3.3. Calibración seguidor–cámara ("hand–eye") 30

3.4. Procesamiento de imágenes endoscópicas 32

3.5. Procesamiento de imágenes de ultrasonido 40

3.6. Resultados: adquisición de información 3D 42

3.7. Resumen y conclusiones 46

4. Modelado 3D de objetos virtualizados usando computación neuronal 51

4.1. Modelado 3D empleando redes neuronales multicapa (MLFFNN) 52

4.2. Modelado 3D empleando redes neuronales auto–organizadas 57

4.3. Resultados: modelado 3D empleando computación neuronal 64

4.4. Resumen y conclusiones 67

5. Modelado neuronal acelerado por hardware usando GPU 71

5.1. Aceleración del algoritmo NGN 71

5.2. Pruebas experimentales 72

5.3. Resumen y conclusiones 75

6. Resultados: modelado 3D de tumores cerebrales 77

6.1. Adaptación o entrenamiento secuencial 77
6.2. Resultados 79
6.3. Conclusiones 86

7. Conclusiones y trabajo futuro 89

A. Apéndice: Filtro de Kalman 93

A.1. Proceso de estimación 93
A.2. Filtro discreto de Kalman 94
A.3. Filtro extendido de Kalman 96
A.4. Entrenamiento de redes neuronales con el filtro de Kalman 97

B. Apéndice: Procesamiento de imágenes de ultrasonido 101

B.1. Caracterización de la señal 102
B.2. Reducción de ruido y realce 105
B.3. Resultados 106
B.4. Resumen y conclusiones 108

C. Apéndice: Publicaciones 111

Bibliografía 113

Lista de Figuras

1-1. Esquema de modelado 3D usando endoneurosonografía y redes neuronales . . 3

2-1. Modelo masa–resorte 8
2-2. Modelado de una nube de puntos utilizando polígonos 9
2-3. Modelado 3D usando geometría sólida constructiva 10
2-4. Modelado 3D empleando FEM 11
2-5. Modelado 3D usando voxelización 12
2-6. Modelado 3D usando representaciones implícitas 13
2-7. Superficie activa ("snake") 14
2-8. Modelo 3D empleando superficies activas ("snakes") 14
2-9. Modelado 3D usando "b–splines" triangulares 16
2-10.Esquema de modelado 3D utilizando redes neuronales 16
2-11.Modelado 3D usando redes neuronales gas crecientes 18

3-1. Imágenes que componen la endoneurosonografía 22
3-2. Problemas a resolver con la metodología de adquisición propuesta 23
3-3. Sistema de adquisición endoneurosonográfica 24
3-4. Sistema de seguimiento visual POLARIS 25
3-5. Sonda de ultrasonido con su haz de emisión radial 26
3-6. Patrón plano empleado para la calibración del endoscopio estéreo 29
3-7. Calibración seguidor–cámara ("hand–eye") 31
3-8. Metodología espacio–temporal de segmentación de la sonda de ultrasonido . 33
3-9. Cálculo del flujo óptico empleando el algoritmo de Horn y Schunck 34
3-10.Esquema de segmentación de la sonda de ultrasonido 35
3-11.Segmentación de la sonda de ultrasonido empleando agrupamiento 36
3-12.Eje de la sonda de ultrasonido extraído mediante PCA 37
3-13.Resultados del seguimiento con completa visibilidad de la sonda 38
3-14.Diagrama de flujo de la restricción de movimiento aplicada 39
3-15.Resultados del seguimiento de la sonda sin restricción de movimiento 40
3-16.Resultados del seguimiento de la sonda con restricción de movimiento 40
3-17.Microtransductor de ultrasonido y su controlador 41
3-18.Segmentación morfológica de estructuras cerebrales 42
3-19.Determinación del eje de la sonda y la posición del sensor de ultrasonido . . 44

3-20.Ubicación 3D de los planos de las imágenes de ultrasonido 45
3-21.Información 3D de un tumor fantasma a partir de 790 cuadros ENS 46
3-22.Información 3D de un tumor fantasma a partir de 164 cuadros ENS 47
3-23.Información 3D de un tumor fantasma a partir de 280 cuadros ENS 48

4-1. Esquema de modelado 3D basado en MLFFNN 53
4-2. Generación de puntos extra a partir de una nube puntos 55
4-3. MLFFNN: evolución del entrenamiento con filtro de Kalman 56
4-4. MLFFNN: comparación del error cuadrático medio 57
4-5. MLFFNN: comparación del número de épocas de entrenamiento 58
4-6. Modelado 3D empleando MLFFNN 58
4-7. Tipos de mallas SOM 59
4-8. Proceso de adaptación SOM 61
4-9. Modelado 3D basado en Redes Neuronales Gas (NGN) 63
4-10.Objetos de realidad virtualizada empleados en los experimentos 65
4-11.Comparación del tiempo de entrenamiento 67
4-12.Modelado 3D de piezas arqueológicas empleando redes neuronales 68
4-13.Entrenamiento grueso y fino empleado para el modelado 3D 69
4-14.Comparación del tiempo de visualización 70

5-1. Comparación del tiempo de entrenamiento en GPU y CPU 73
5-2. Error del modelado 3D para diferentes épocas de entrenamiento 74
5-3. Evolución del modelado NGN durante el entrenamiento 74
5-4. Desempeño del modelado para diferentes números de neuronas de la red . . . 75

6-1. Esquema de modelado 3D de tumores cerebrales 78
6-2. Modelado 3D de tumores cerebrales usando ENS y redes neuronales 80
6-3. Modelado 3D de tumores cerebrales usando ENS y redes neuronales 80
6-4. Modelado 3D del tumor de la secuencia 1 usando SOM 81
6-5. Modelado 3D del tumor de la secuencia 2 usando SOM 82
6-6. Modelado 3D del tumor de la secuencia 3 usando SOM 82
6-7. Modelado 3D del tumor de la secuencia 4 usando SOM 83
6-8. Modelado 3D del tumor de la secuencia 1 usando NGN 83
6-9. Modelado 3D del tumor de la secuencia 4 usando NGN 84
6-10.Modelado 3D del tumor de la secuencia 5 usando NGN 84

7-1. Propuesta de trabajo futuro 90

A-1. Ecuaciones de predicción y corrección del filtro de Kalman 95
A-2. MLFFNN entrenado con filtro extendido de Kalman 98

B-1. Ejemplos de ruido “speckle” en imágenes de ultrasonido 102

B-2. Transformada rápida de Fourier y extracción de bordes 103
B-3. Filtrado gaussiano para hallar el orden de la señal 104
B-4. Función wavelet *daubechies 4* 105
B-5. Diagrama de flujo de la metodología de procesamiento de imágenes US . . . 106
B-6. Coeficientes de aproximación y detalle de la descomposición wavelet 107
B-7. Ecografía: Descomposición wavelet utilizando *daubechies 4* 108
B-8. Ultrasonido radial: Descomposición wavelet utilizando *daubechies 4* 109
B-9. Resultado del procesamiento empleando filtrado wavelet 110

Lista de Tablas

3-1. Parámetros intrínsecos del sistema endoscópico estéreo 28
3-2. Parámetros extrínsecos del sistema endoscópico estéreo 30
3-3. Errores de seguimiento con respecto al método manual ("ground-truth") . . 41

4-1. Comparación del error cuadrático medio de las arquitecturas MLFFNN . . . 54
4-2. Comparación de los parámetros de diseño de cada arquitectura neuronal . . 64
4-3. Distancia desde cada punto hasta la superficie del modelo generado 66

5-1. Aceleración del modelado 3D para diferentes objetos virtualizados 75

6-1. Secuencias ENS empleadas para las pruebas experimentales 79
6-2. Error topográfico y de cuantización del modelado 3D 85
6-3. Errores de cuantización del modelado 3D (milímetros) 86

Abreviaturas

Abreviatura	Término
ANN	Redes Neuronales Artificiales
BMU	Neurona de Mejor Ajuste
CAD	Diseño Asistido por Computador
CL	Aprendizaje Competitivo
CPU	Unidad Central de Procesamiento
CSG	Geometría Solida Constructiva
CT	Tomografía Computarizada
$CUDA$	Compute Unified Device Architecture
DBA	Diferencia Entre Ángulos
DOF	Grados de Libertad
EAM	Error Absoluto Medio
ECM	Error Cuadrático Medio
EBC	Distancia Euclidiana Entre Centroides
EKF	Filtro Extendido de Kalman
ENS	Endoneurosonografía
GPU	Unidad de Procesamiento Gráfico
$HIFU$	Ultrasonido Enfocado de Alta Intensidad
KF	Filtro de Kalman
LM	Levenberg–Marquardt
$MLFFNN$	Red Neuronal Multicapa Alimentada Hacia Adelante
MRI	Imagen de Resonancia Magnética
NCD	Difusión Coherente no–lineal
NGN	Red Neuronal Gas
NN	Red Neuronal
PC	Componente Principal
PCA	Análisis de Componentes Principales
PCI	Percepción y Control Inteligente
PET	Tomografía por Emisión de Positrones
R	Matriz de Rotación
RBF	Función de Base Radial
ROI	Región de Interés
RX	Rayos X

Abreviatura	**Término**
SOM	Mapa Auto–Organizado
T	Vector de Traslación
US	Ultrasonido
VQ	Cuantización de Vectores
$WSCE$	Adelgazamiento Wavelet con Ajuste de Contraste

1. Introducción

Las intervenciones mínimamente invasivas no precisan de heridas importantes para acceder a la zona del organismo que necesita ser operada, esto hace que el postoperatorio sea más corto, menos doloroso, con una corta hospitalización y una rápida incorporación a la vida normal. Este tipo de intervenciones han tomado mucho auge en los últimos años debido a que reducen los riesgos típicos de una intervención tradicional. En los primero años de aparición de esta técnica, el inconveniente era que no todos los órganos y tejidos del cuerpo podían ser operados de esta manera debido a restricciones técnicas. Sin embargo, los continuos desarrollos en herramientas quirúrgicas y sistemas de visualización disminuyen estas limitaciones día a día.
En la actualidad, existen casi tantos tipos de cirugías mínimamente invasivas como especialidades quirúrgicas, entre las cuales se destacan la angioscopia (vasos sanguíneos), artroscopia (articulaciones), broncoscopia (bronquios y pulmones), colposcopia (vagina y útero), endoscopía digestiva, laparoscopia (cavidad abdominal), laringoscopia (laringe) y cirugía video-asistida en general. El caso en que los videos intra–operativos son combinados interactivamente y en tiempo real con imágenes pre–operativas, se denomina Cirugía Guiada por Imágenes.
En el caso de las neurocirugías, las tendencias recientes sugieren la utilización conjunta de imágenes endoscópicas y de ultrasonido, una técnica llamada Endoneurosonografía (ENS), como herramienta de apoyo al neurocirujano en la visualización de estructuras internas del cerebro en escenarios intra–operativos.
Con las imágenes endoscópicas, el neurocirujano puede ver lo que pasa al interior del cráneo. Sin embargo, no puede ver lo que pasa más allá de las estructuras opacas, por lo que es necesario complementar su uso con otras técnicas de imagenología médica como resonancia magnética (MRI), tomografía computarizada (CT), tomografía por emisión de positrones (PET) o ultrasonido (US), siendo esta última la más adecuada y económica para escenarios intra–operativos.
La Endoneurosonografía permite la adquisición en tiempo real y en escenarios intra–operativos de información 3D de estructuras cerebrales, tales como tumores, a partir de videos endoscópicos y de ultrasonido. Esto significa que mientras el neurocirujano manipula el equipo, durante la intervención quirúrgica, la información 3D que define la morfología del tumor se actualiza constantemente y debe ser empleada para actualizar un modelo de representación. En este sentido, el modelado tridimensional con redes neuronales ofrece una ventaja, ya que la representación puede desarrollarse en dos etapas: una etapa de entrenamiento fuera de línea, que permite generar un modelo inicial del tumor a partir de información ENS preliminar; y

una etapa de entrenamiento en línea, que permite ajustar el modelo de acuerdo a la nueva información ENS recibida.

El objetivo de esta tesis es desarrollar un modelo de representación tridimensional, basado en redes neuronales artificiales, que permita la actualización de la forma de un tumor cerebral a partir de información recibida de un equipo médico de endoneurosonografía. En este sentido, el trabajo se divide en dos etapas: la primera, el desarrollo de una metodología de adquisición de información tridimensional a partir del uso combinado de endoscopía estéreo y ultrasonido 2D. La segunda, el desarrollo de una metodología de modelado 3D basado en redes neuronales artificiales.

Para la etapa de adquisición, se utilizan técnicas de visión estéreo que permiten la calibración, rectificación y reconstrucción del escenario intra–operativo visto por las cámaras endoscópicas. Las imágenes de ultrasonido son empleadas para segmentar las estructuras cerebrales de interés mediante técnicas de procesamiento morfológico de imágenes. Finalmente, se aplica una metodología de seguimiento espacio–temporal a la sonda de ultrasonido que permite estimar su pose en el espacio 3D y localizar la información de los tumores dentro de un espacio coordenado de referencia.

Para la etapa de modelado, se analiza el uso de redes neuronales multicapa alimentadas hacia adelante (MLFFNN), mapas auto–organizados (SOM) y redes neuronales gas (NGN) para el modelado 3D de tumores a partir de nubes de puntos 3D. Para comprobar las bondades de estas técnicas de modelado, se realizan pruebas experimentales con objetos de realidad virtualizada como tumores cerebrales fantasma, piezas arqueológicas, rostros, frutas, botellas y otros objetos.

Para el entrenamiento de la MLFFNN, se utilizan los algoritmos Levenberg–Marquardt (LM) y Filtro de Kalman (KF). Se realizan pruebas de modelado utilizando la misma arquitectura y los mismos conjuntos de entrenamiento y validación para ambos métodos de entrenamiento. El Error Cuadrático Medio (ECM) es la medida de desempeño empleada. De acuerdo con los resultados obtenidos, ambos métodos presentan errores comparables, con una leve superioridad del algoritmo LM. Sin embargo, para alcanzar un rendimiento dado, el entrenamiento por KF requiere menos épocas de entrenamiento que LM, lo que indica una convergencia más rápida.

Para el entrenamiento de las arquitecturas auto–organizadas se emplea un esquema de aprendizaje competitivo no supervisado. En este tipo de redes, las neuronas se ajustan asintóticamente a la superficie del objeto.

Las tres arquitecturas neuronales son comparadas en términos de desempeño, costo computacional, tiempo de procesamiento, conveniencia de la visualización y número de épocas de entrenamiento. Los resultados experimentales demuestran que las arquitecturas neuronales auto–organizadas son las más adecuadas para la tarea de modelado 3D.

Para estudiar la posible aplicación en tiempo real de los métodos de modelado neuronal, el algoritmo NGN es acelerado por hardware mediante una unidad de procesamiento gráfico (GPU). Para esta tarea, el algoritmo clásico NGN es implementado paralelamente en una

GPU, y se compara con su implementación serial en una Unidad Central de Procesamiento (CPU). Los resultados computacionales indican que el proceso acelerado es más veloz cuando se trabaja con redes neuronales con más de 10^4 neuronas.
Este documento está organizado como sigue. En el Capítulo 2 se hace una revisión del estado del arte de las técnicas de imagenología médica, aplicadas a la cirugía mínimamente invasiva, y del modelado tridimensional de objetos de realidad virtualizada. En el Capítulo 3 se explica la metodología de adquisición de información 3D de estructuras internas del cerebro empleando endoneurosonografía, se describe el montaje experimental, las características del laboratorio y se presentan resultados de adquisición con tumores cerebrales fantasma. En el Capítulo 4 se presenta un esquema de modelado 3D de objetos de realidad virtualizada mediante computación neuronal, se realizan comparaciones entre las arquitecturas MLFFNN, SOM y NGN, y se muestran los resultados de modelado utilizando objetos de realidad virtualizada adquiridos con diferentes sensores. En el Capítulo 5 se describe la implementación acelerada por hardware del algoritmo NGN con el fin de explorar la posibilidad del modelado tridimensional en tiempo real. En el Capítulo 6 se presentan resultados de modelado 3D sobre una base de datos endoneurosonográfica propia. Finalmente, en el Capítulo 7 se dan conclusiones sobre el trabajo y se indican las tendencias para la investigación futura.
La Figura **1-1** muestra un diagrama de bloques que esquematiza el trabajo desarrollado en esta tesis.

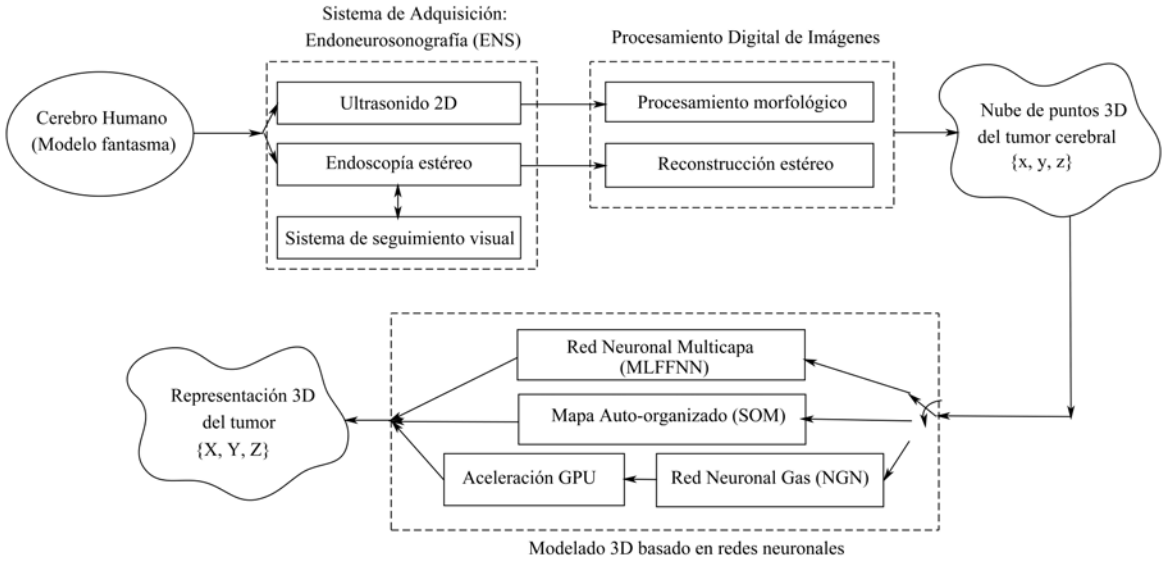


Figura 1-1.: Esquema de modelado 3D usando endoneurosonografía y redes neuronales

2. Revisión de la literatura

2.1. Estado del arte: imagenología médica

Recientes tendencias en cirugía cerebral mínimamente invasiva han sugerido el uso conjunto de endoscopía y ultrasonido para hacer reconstrucciones tridimensionales de las estructuras internas del cerebro, una técnica llamada Endoneurosonografía (ENS) [95]. Las imágenes endoscópicas son un componente esencial ya que permiten al neurocirujano ver lo que pasa al interior del cráneo. Sin embargo, las imágenes endoscópicas no permiten ver lo que pasa más allá de las estructuras opacas, por lo cual es necesario utilizar técnicas como resonancia magnética (MRI), tomografía computarizada (CT), tomografía por emisión de positrones (PET) o ultrasonido (US), siendo esta última la más adecuada para escenarios intra–operativos [77].

2.1.1. Imágenes endoscópicas

En los últimos años, la detección y el tratamiento del cáncer ha demandado la atención de los investigadores de todo el mundo, debido a la alta proliferación de la enfermedad y a que un diagnóstico temprano puede aumentar la probabilidad de éxito de los tratamientos existentes. Existen diferentes técnicas de imagenología médica que ayudan a la detección de este tipo de patologías, siendo la endoscopía una de las que ha tenido mayor crecimiento en los últimos años debido a su precisión, mínima invasión y su uso como herramienta de tratamiento en las etapas tempranas de la lesión [41].
En 1949, Uji y otros [48] desarrollaron en la Universidad de Tokio la primera cámara gástrica para aplicaciones clínicas. Un año más tarde, Olympus Corp. desarrolló el primer endoscópico gástrico comercial. En 1983, Welch–Allyn Inc. comercializó el primer video–endoscopio, y desde entonces muchas compañías se han dedicado a la fabricación de diversos tipos de endoscopios con mejores materiales de fabricación, mayor calidad de la imagen y más larga durabilidad.
El diagnóstico mediante imágenes endoscópicas consiste en detectar anormalidades en el color, la forma y la superficie del tejido, considerando la historia clínica del paciente. Los médicos deben tener la suficiente experiencia para diagnosticar, mediante la observación, el tipo de anomalía, el nivel de invasión y la posibilidad de metástasis. Si la lesión es identificada como un tumor, debe elegirse tratamiento por cirugía, quimioterapia, radioterapia o terapia endoscópica.

En 2005, Olympus Corp. lanzó el primer endoscopio multifuncional con alternativas de visualización que facilitan la tarea del médico realzando regiones de interés mediante el uso de diferentes iluminaciones, entre las que se encuentran iluminación blanca, iluminación fluorescente, iluminación de banda angosta e iluminación infrarroja [106, 58].
En el trabajo de Darabi y otros [27], se presenta una técnica de visualización llamada *endoscopía simulada* en la que se genera un escenario neuro–quirúrgico virtualizado a partir de información endoscópica. Esta técnica es una manera intuitiva de presentar información tridimensional para apoyar la planeación de neurocirugías; sin embargo, los expertos médicos deben ser cuidadosos y en lo posible tener algún conocimiento sobre reconstrucción 3D para saber reconocer las limitaciones de la técnica.

2.1.2. Imágenes de ultrasonido

Los escáneres médicos de ultrasonido son sistemas no invasivos, rápidos, seguros, y de relativo bajo costo que se han convertido en una de las herramientas de diagnóstico más utilizadas en los hospitales modernos.
Las imágenes de ultrasonido se obtienen exponiendo parte del cuerpo a ondas acústicas de alta frecuencia para producir imágenes del interior del organismo a partir del fenómeno de reflexión. Los exámenes por ultrasonido no utilizan radiación ionizante, por lo que no representan riesgo para el paciente. Debido a que las imágenes por ultrasonido se capturan en tiempo real, pueden mostrar la estructura y el movimiento de los órganos internos del cuerpo, así como también la sangre que fluye por los vasos sanguíneos. Estas imágenes son en general pruebas clínicas no dolorosas que ayudan al diagnóstico y tratamiento de muchas enfermedades.
Su aplicación más conocida son la ecografías prenatales, sin embargo, existe un amplio rango de aplicación, como por ejemplo la detección de tumores y anomalías arterio–venosas en varias partes del cuerpo, incluido el cerebro. Estas patologías se identifican como manchas o regiones irregulares en la imagen entregada por el escáner. Cuando el neurocirujano analiza estas imágenes, realiza su diagnóstico con base, entre muchas otras variables, en la geometría, la velocidad y la forma de crecimiento de la región. Esta tarea puede apoyarse efectivamente utilizando técnicas de procesamiento digital de imágenes e inteligencia artificial [95].
Un tipo especial de imagen de ultrasonido, llamada ultrasonido Doppler, evalúa la sangre mientras circula por los vasos sanguíneos, lo que permite diagnosticar, por ejemplo, anomalías arterio–venosas.
Para el tratamiento y diagnóstico de tumores cerebrales existe un tipo específico de ultrasonido llamado ultrasonido enfocado de alta intensidad (HIFU), cuya frecuencia se ubica entre 250 kHz y 2 MHz, que permite estudiar y tratar zonas específicas del cerebro sin afectar otras regiones que no son de interés [57, 44].
Las imágenes de ultrasonido son muy sensibles al ruido, lo que las hace difíciles de interpretar comparadas con otras técnicas de imagenología médica. Sin embargo, presentan la ventaja

de ser mucho más económicas y no tener efectos secundarios como las imágenes MRI y las imágenes de rayos X (RX), lo que representa una potencialidad para su uso masivo en el tratamiento y diagnóstico de diversas enfermedades. El éxito de este tipo de imágenes está en la aplicación de un preprocesamiento adecuado [121].
Una de las aplicaciones de las imágenes de ultrasonido es el diagnóstico de anomalías cerebrales como tumores y trastornos arterio–venosos. Con un preprocesamiento adecuado de las imágenes, las regiones donde se encuentran estos trastornos pueden ser segmentadas y sus contornos pueden ser extraídos [57, 44].

2.2. Estado del arte: modelado 3D

En el modelado 3D existen dos enfoques principales: a) el primero, llamado *realidad virtual*, que se refiere a la construcción de modelos artificiales de objetos reales o imaginarios para aplicaciones en videojuegos, diseño visual, diseño industrial, desarrollo de prototipos, entre otros; b) el segundo, llamado *realidad virtualizada*, sobre el que se enfoca este trabajo, consiste en desarrollar modelos de objetos reales cuya información es adquirida mediante visión estereoscópica, escáneres de rango, ENS, entre otros. La diferencia radica en que los modelos de realidad virtualizada utilizan información 3D proveniente directamente del mundo real (adquirida mediante sensores físicos), mientras que en la realidad virtual, la información utilizada para generar el modelo podría no provenir del mundo real y ser producto únicamente de la imaginación y el talento del diseñador.
Con el rápido avance de los sistemas de gráficos computacionales (tarjetas aceleradoras, memorias de mayor capacidad y procesadores más veloces), en los últimos años se han desarrollado una gran variedad de técnicas para el modelado 3D de objetos en entornos de realidad virtualizada. Algunos de ellos son: Sistemas masa–resorte, mallas de polígonos, parches paramétricos, geometría sólida constructiva (CSG), métodos de elementos finitos (FEM), modelos implícitos y redes neuronales. A continuación se hace una breve reseña bibliográfica de cada una de ellas.

2.2.1. Sistemas masa–resorte

Los modelos basados en sistemas masa–resorte son una manera simple de modelar objetos deformables a partir de un sistema dinámico completamente caracterizado. Estos modelos son fáciles de construir y pueden ser animados más sencillamente que los modelos basados en polígonos. Una de las principales aplicaciones es la animación de rostros humanos, mediante modelos masa–resorte que emulan las tres capas principales del tejido blando: músculo, tejido graso y epidermis. Adicionalmente, estos modelos son extensos pero simples, por lo que pueden realizarse simulaciones en tiempo real empleando hardware acelerador gráfico [37].

Los inconvenientes de este tipo de sistemas son: a) proveen una aproximación discreta de lo que realmente ocurre en un cuerpo continuo, por lo que su precisión depende del número de elementos de representación empleados; b) el comportamiento del modelo depende del número y valores de masa y constante elástica usadas en cada elemento de representación, y estos valores generalmente no son fáciles de deducir a partir de la observación y la experimentación; c) ciertas características físicas de objetos de volumen incompresible y objetos con superficies delgadas resistentes a la curvatura no pueden ser representadas; y, d) los sistemas masa–resorte exhiben problemas cuando se quieren modelar objetos rígidos, ya que en ese caso la constante elástica de los resortes debe ser alta, lo que genera inestabilidad en los algoritmos numéricos que se emplean para solucionar el modelo [4].
En la Figura **2-1** se muestra una porción de un modelo masa–resorte presentado en [37]. Las conexiones entre resortes ejercen fuerzas sobre todos los vecinos cuando alguna de las masas es desplazada de su posición original.

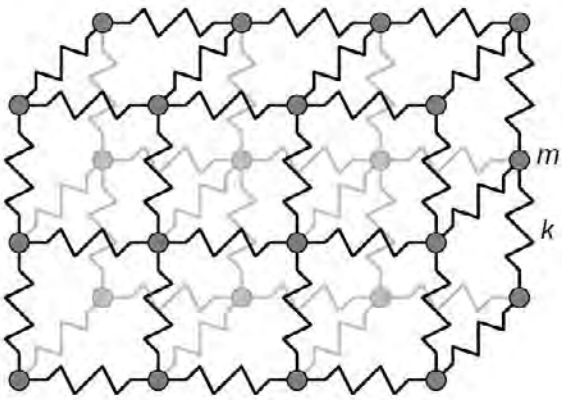
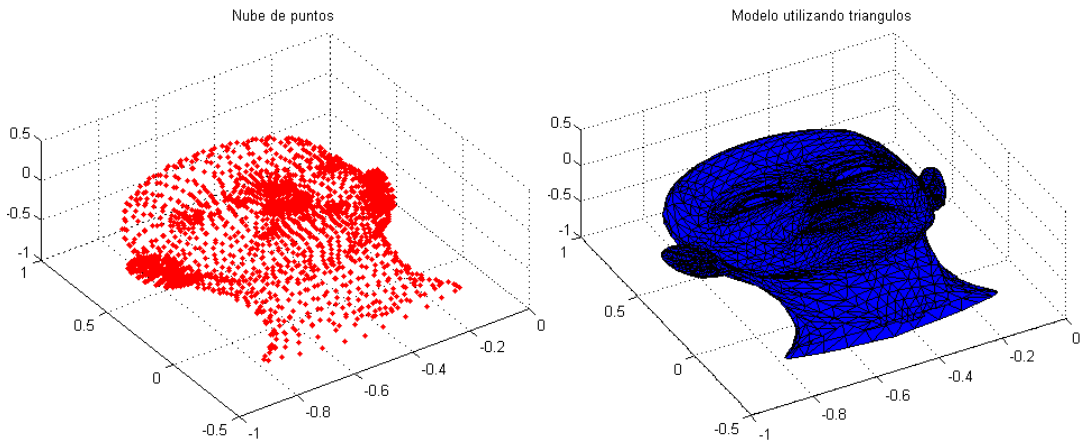


Figura 2-1.: Modelo masa–resorte. Tomado de [37]

2.2.2. Modelos basados en polígonos

Modelar objetos tridimensionales por polígonos es una de las metodologías más populares en la actualidad. Consiste en aproximar la superficie de un objeto por medio de una malla de figuras geométricas cerradas, generalmente triángulos. En este sentido, es posible aproximar cualquier superficie con la exactitud deseada. Las desventajas de esta técnica surgen cuando se desean modelar objetos de forma compleja, debido a que la precisión está directamente relacionada con el número de polígonos que se deben emplear, por lo que es común encontrar aplicaciones donde se emplean cientos de miles de polígonos [32]. Este problema influye en el costo computacional, el tiempo de visualización y los requerimientos de los equipos de cómputo. En la Figura **2-2** se observa un ejemplo de modelado 3D utilizando polígonos (triángulos). Nótese que este tipo de modelos requieren un algoritmo previo (como el método de triangulación de Delaunay [16]) que defina los triángulos a emplear por el modelo, los

cuales deben ser recalculados ante la llegada de nuevos datos.



(a) Nube de puntos (b) Modelos basado en polígonos

Figura 2-2.: Modelado de una nube de puntos utilizando polígonos

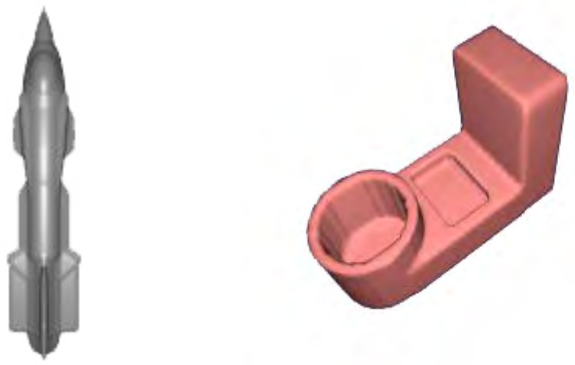
2.2.3. Geometría sólida constructiva (CSG)

La geometría sólida constructiva (CSG) se basa en el principio de que cualquier estructura geométrica puede ser representada como la combinación de figuras más simples [15]. El procedimiento consiste en combinar objetos de geometría primitiva mediante operaciones de unión, intersección y diferencia. Este procedimiento puede escribirse con topología de árbol de decisión, donde los nodos representan las formas básicas (esferas, cilindros, cubos, etc.), y las ramificaciones representan las operaciones.

En el trabajo de Goldfeather y otros [40], se describen un conjunto de algoritmos para el modelado de objetos 3D usando CSG y representación de vecindarios. Su desventaja es que requieren gran cantidad de memoria, mientras que la ventaja es que pueden emplear información de color para la representación.

En el trabajo de Breen y otros [13], se presenta una metodología para convertir modelos volumétricos CSG en modelos de distancia formados por una base de datos donde se almacenan las distancias más cortas entre cada objeto de representación CSG y la superficie del objeto. Este tipo de datos es muy útil en diversas aplicaciones gráficas, tales como representación 3D de información proveniente de escáneres MRI y CT, evaluación de superficies, transformaciones afines y “morphing”.

En la Figura **2-3** se muestra un ejemplo de modelado tridimensional empleando geometría sólida constructiva.



(a) (b)

Figura 2-3.: Modelado 3D usando geometría sólida constructiva. Tomado de [13]

2.2.4. Método de elementos finitos (FEM)

Los métodos basados en elementos finitos son usados para encontrar aproximaciones de una función continua que satisfaga alguna expresión de equilibrio, como por ejemplo una expresión de deformación. En FEM, un objeto se divide en puntos discretos unidos por nodos, luego se calcula una función que resuelva la ecuación de equilibrio para cada elemento, y su solución es modificada de acuerdo al vecindario de cada elemento hasta que se cumpla la condición de continuidad [37].

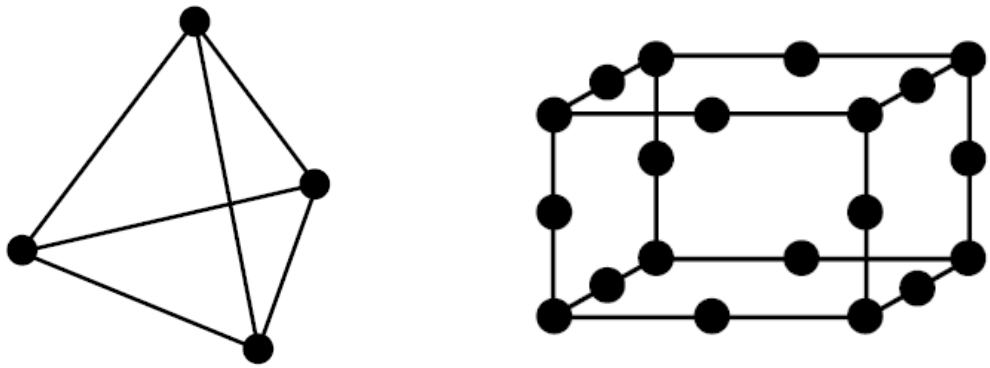
Una de las principales ventajas de este método es la versatilidad, debido a la variedad de objetos que pueden ser modelados. Tradicionalmente, FEM ha sido aplicado con éxito al modelado de materiales rígidos como metales, donde la deformación se limita al 1 % de las dimensiones del metal [18]; pero también ha sido aplicado al modelado de tejidos musculares, donde la deformación puede ser más del 100 % de las dimensiones iniciales del músculo [43].

El uso de FEM ha estado limitado por los requerimientos computacionales de los algoritmos desarrollados. En particular, existen dificultades técnicas cuando se desean realizar modelos en tiempo real. Esto se debe a que los vectores de fuerza y las matrices de masa y rigidez se calculan mediante integración numérica sobre el objeto, las cuales, en teoría, deben se recalculan ante cada deformación del objeto. Este recálculo es costoso computacionalmente, y en muchas de las aplicaciones reales se prefiere asumir que los objetos presentan pequeñas deformaciones para evitar dicho cálculo.

En el trabajo de Jaramillo y otros [59], se presenta una metodología de alineamiento y comparación de modelos tridimensionales aplicada a la inspección de piezas manufacturadas en la industria. El método propuesto utiliza FEM para estimar la deformación física del modelo real con respecto al modelo de referencia, y funciones de base radial (RBF) para acelerar el proceso y abrir la posibilidad de la inspección en tiempo real.

En la Figura **2-4** se muestra el modelado 3D de dos objetos simples empleando el método de elementos finitos. En la Figura **2-4**a se muestra un tetraedro representado mediante

4 elementos, y en la Figura **2-4**b se muestra un paralelepípedo representado mediante 20 elementos.



(a) Tetraedro: 4 elementos (b) Paralelepípedo: 20 elementos

Figura 2-4.: Modelado 3D empleando FEM. Tomado de [37]

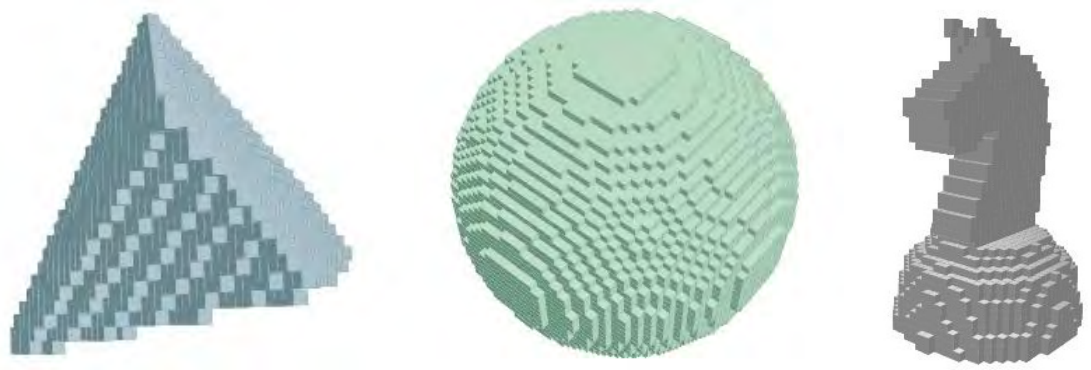
2.2.5. Subdivisión espacial: voxelización

Esta técnica consiste en dividir el espacio tridimensional en cubos elementales llamados vóxeles, donde cada uno es etiquetado como *lleno* o *vacío* de acuerdo a la forma del objeto que se desea representar. Esta representación es costosa computacionalmente, y su precisión depende del volumen de los vóxeles seleccionados; sin embargo, es una de las técnicas de representación más popular, y existen modificaciones al algoritmo original que ofrecen un rápido renderizado [91].

En el trabajo de Steinbach y otros [105], se presenta una metodología para reconstruir objetos del mundo real empleando vóxeles extendidos; la representación mínima de un vóxel es un punto, y es la representación que menor costo computacional demanda, pues solo se requiere conocer las coordenadas espaciales de su centro; sin embargo, presenta problemas de oclusión y huecos cuando se desean implementar representaciones volumétricas.

En el trabajo de Lin y otros [70], se propone una metodología llamada cubos marchantes para renderizar superficies 3D a partir de información volumétrica; la renderización superficial es un técnica de visualización en la que solo intervienen los vóxeles más externos del objeto, es decir, aquellos que son visibles y sólo representan su superficie. En general, esta metodología presenta 3 desventajas principales: a) La renderización demanda mucho tiempo debido a la cantidad de cubos requeridos; b) Durante la combinación de vóxeles adyacentes pueden producirse superficies erróneas debido a las diferentes configuraciones de vóxeles (existen 15 configuraciones diferentes); y, c) Es posible que aparezcan huecos en la representación debido a problemas de continuidad en vóxeles vecinos.

En la Figura **2-5** se muestran ejemplos de modelado tridimensional empleando voxelización. Estas imágenes fueron tomadas de [71].



(a) (b) (c)

Figura 2-5.: Modelado 3D usando voxelización. Tomado de [71]

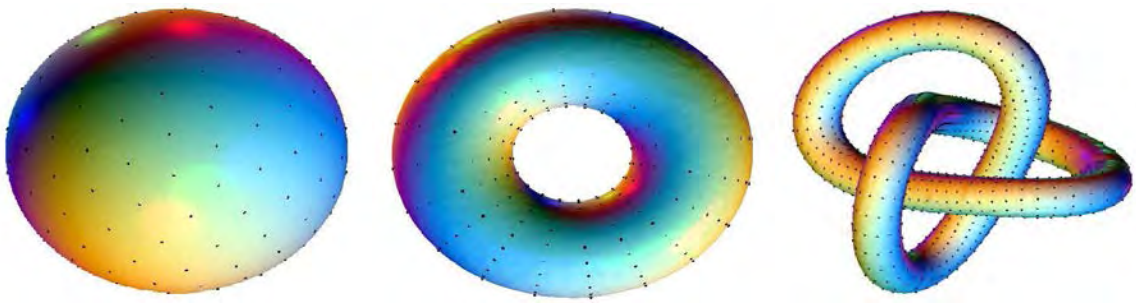
2.2.6. Representaciones implícitas

La representación implícita de objetos tridimensionales consiste en definir el objeto por medio de una expresión matemática implícita, como por ejemplo, una esfera, un cono o un paraboloide. La desventaja de esta representación es que solo un número limitado de formas pueden representarse mediante expresiones matemáticas, por lo que su aplicación en casos reales es limitada. La ventaja es que los modelos generados pueden manipularse y transformarse fácilmente [9].

En el trabajo de Liu y otros [72], se presenta un método para representación de superficies implícitas a partir de una nube de puntos tridimensionales. La metodología consiste en hallar una función continua $\phi(x, y, z)$, que represente la superficie envolvente de la nube de puntos a partir de la minimización de un funcional de energía.

En el trabajo de Ardon y otros [3], se introduce una metodología de segmentación basada en bordes mediante aproximaciones implícitas de objetos 3D aplicada a imágenes médicas. Esta representación es una extensión 3D del método de segmentación llamado contornos activos o "snakes", el cual se compone de una curva paramétrica deformable que se ajusta progresivamente a un contorno [62]; en la extensión 3D de este método, se trabaja con una superficie paramétrica deformable, por lo que su costo computacional es mayor.

En la Figura **2-6** se muestran ejemplos de modelado tridimensional empleando representaciones implícitas. Estas imágenes fueron tomadas de [72].



(a) (b) (c)

Figura 2-6.: Modelado 3D usando representaciones implícitas. Tomado de [72]

2.2.7. Contornos/Superficies activas ("snakes")

Kass y otros en [62] introdujeron el concepto de contornos activos o "snakes" para resolver diferentes problemas en visión de máquina y análisis de imágenes. Las "snakes" son curvas deformables empleadas para definir contornos o para hacer seguimiento de objetos en movimiento en secuencias de imágenes. Las "snakes" responden interactivamente a fuerzas internas que se resisten al estiramiento y a la curvatura, a fuerzas externas definidas por las características de la imagen, y a fuerzas definidas por el usuario.

Las "snakes" pueden ser entendidas como un modelo matemático de una curva deformable construida de un material flexible [7, 66]. En el campo del procesamiento de imágenes, una "snake" es un contorno activo que puede cambiar su forma dinámicamente para acoplarse a los bordes de un objeto determinado. Estos contornos se componen de puntos de control, llamados "snaxels", que se van moviendo hasta encontrar el mejor acople con el contorno real, como se muestra en la Figura **2-7**. Finalmente, mediante un proceso de optimización, la geometría de la "snake" se modifica hasta obtener el contorno del objeto analizado.

Si $v(s)$ es la posición parametrizada de la "snake", la fuerza de deformación interna se expresa como se muestra en la Ecuación 2-1

$$V = \frac{1}{2}\int \left[\alpha\left(s\right)\left\|\frac{dv(s)}{ds}\right\|^2 + \beta\left(s\right)\left\|\frac{d^2v(s)}{ds^2}\right\|^2\right] ds \tag{2-1}$$

La primera y segunda derivada corresponden a las deformaciones axiales y de curvatura, respectivamente, y los parámetros $\alpha(s)$ y $\beta(s)$ son sus ponderaciones. Por ejemplo, al hacer $\beta(s_0) = 0$ se permite que la "snake" tome forma de esquina en el punto s_0.

La fuerza externa, definida por la imagen, se integra sobre la longitud de la "snake" y la atrae hacia las regiones de interés. Una de las fuerzas externas más empleadas es el gradiente de la imagen. Con el objetivo de evitar mínimos locales en el camino a la solución deseada, el usuario define las fuerzas aplicadas con base en su conocimiento *a priori* de la

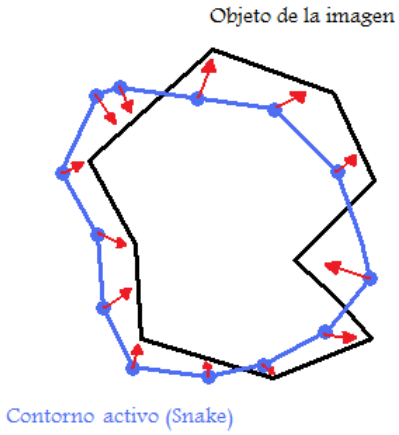


Figura 2-7.: Superficie activa ("snake"). Las flechas indican el movimiento de cada punto de control ("snaxel") hacia los bordes del objeto analizado

región. Para una solución numérica, la "snake" se discretiza como un conjunto de puntos $\{v_i\}$ parametrizados a lo largo de la curva. La deformación se simula empleando técnicas de integración de Euler que intentan minimizar la energía total del contorno activo.

Los contornos activos 2D han sido generalizados como *superficies activas* y *volúmenes activos*. Estas extensiones permiten aplicaciones 3D de segmentación de superficies, ajuste de modelos y seguimiento de regiones de interés. Una de las aplicaciones más comunes es la imagenología médica [96].

En la Figura **2-8** se muestra un ejemplo de modelado 3D empleando la metodología de superficies activas expuesta en [69].

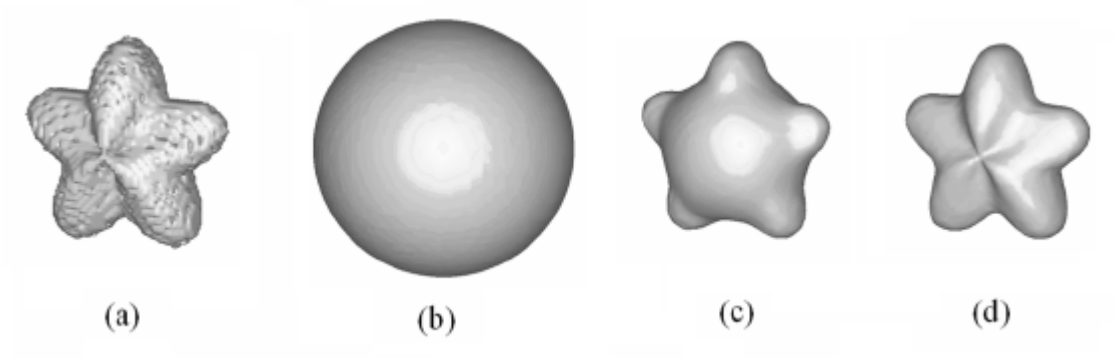


Figura 2-8.: Superficie activa. (a) Objeto a modelar. (b) Inicialización de la "snake". (c) Resultado después de 20 iteraciones. (e) Resultado después de 40 iteraciones. Tomado de [69]

2.2.8. Parches paramétricos

Son modelos netamente geométricos que computan la forma y el movimiento de objetos deformables. Generalmente, este tipo de técnicas son eficientes computacionalmente, pero dependen de la habilidad del diseñador o del usuario, y su comportamiento no obedece a principios físicos.
En el diseño asistido por computador (CAD), los diseñadores necesitan definir numéricamente curvas y superficies para construir la geometría de objetos, pero también requieren métodos intuitivos que permitan refinar esos objetos. Las "splines" o los parches paramétricos han surgido como una de las técnicas más populares de modelado geométrico. Estos métodos permiten representar curvas 2D y 3D, así como también parches 2D para especificar superficies 3D.
En estas representaciones, la curva (o la superficie) se define por un conjunto de puntos de control. El usuario debe ajustar el modelo a la forma del objeto moviendo los puntos de control, agregando puntos, quitando puntos, o modificando su ponderación dentro del modelo. Estos ajustes deben realizarse con base en la habilidad del usuario o en algún criterio de error.
La representación de objetos mediante estos métodos es computacionalmente eficiente y permite modificación interactiva debido a los parámetros de control. Sin embargo, este nivel de control puede ser una desventaja, pues la modificación precisa de la forma del objeto puede ser laboriosa debido a que cualquier cambio implica el ajuste de muchos puntos de control [5, 29].

2.2.8.1. Parches paramétricos bi–cúbicos ("b–splines")

Es un método muy similar al de los polígonos, con la diferencia que no se utiliza una malla de superficies planas sino que se emplean superficies curvas para construir la malla de representación del objeto. Cada superficie es representa por una expresión matemática parametrizada que indica su curvatura y posición en el espacio tridimensional. Al igual que el método de polígonos, es costoso computacionalmente y requiere largos tiempos de renderizado; además, presenta un problema de continuidad debido a la unión entre los parches [109], lo que genera una condición adicional en el proceso de ajuste.
Los parches paramétricos bi–cúbicos o "b–splines" consisten en definir una superficie paramétrica que se ajuste a una nube de puntos dada. Esta técnica es una de las más deseables en el tratamiento de imágenes médicas debido a su continuidad, suavidad en las fronteras, controlabilidad local y compatibilidad con transformaciones afines. Sin embargo, su uso aún no es aceptado ampliamente debido a las dificultades que existen para elegir los parámetros topológicos del modelo a partir de una nube puntos sin ordenar. En [17] se presenta una metodología de modelado de superficies 3D basada en "b–splines" que soluciona parcialmente este problema mediante la selección analítica de puntos de control.
En el trabajo de He y Qin [51], se presenta una técnica de reconstrucción 3D basada en super-

ficies “b–splines” triangulares. La diferencia de esta técnica con los algoritmos tradicionales, es que evita los complicados procedimientos de ajuste entre parches, pues esta metodología se basa en una triangulación previa para la generación de parches con continuidad C^{n-1} en las regiones suaves y continuidad C^0 en las esquinas, por lo que el usuario solo debe especificar el orden n del parche y el error de ajuste ϵ. Una ventaja importante es que la adaptación de los parches se realiza mediante la optimización de una función de costo.
En la Figura **2-9** se muestra un ejemplo de modelado tridimensional empleando “b–splines” triangulares.

(a) Triangulación tradicional (b) Modelo “b–spline” (c) Triángulos “b–spline”

Figura 2-9.: Modelado 3D usando “b–splines” triangulares. Tomado de [51]

2.2.9. Redes neuronales artificiales

Las redes neuronales ofrecen una solución al problema de modelado tridimensional que requiere muy poco espacio de almacenamiento, brinda buena precisión, no requiere conocimiento *a priori* del objeto que se desea modelar, y permite fácil manipulación mediante operaciones de traslación, rotación, escalamiento y deformación [93, 21, 24]. Sin embargo, tienen alto costo computacional debido al entrenamiento y a la inexistencia de una teoría bien definida para determinar la arquitectura óptima que debe tener la red [19, 23, 22, 20].
En la Figura **2-10** se muestra un diagrama de bloques que representa el proceso de modelado mediante redes neuronales.

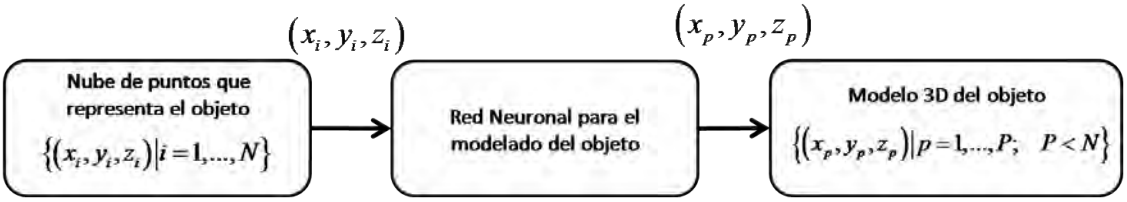


Figura 2-10.: Esquema de modelado 3D utilizando redes neuronales

En el trabajo de Liu y otros [73], se presenta una metodología de reconstrucción superficial basada en Redes Neuronales de Base Radial (RBF–NN) y superficies paramétricas "b-splines". La superficie se construye a partir de una nube de puntos 3D que corresponden al volumen que ocupa el objeto en el espacio. El primer paso consiste en ordenar la nube de puntos mediante una RBF–NN; después se genera un modelo matemático a partir de la combinación lineal de la función radial y los coeficientes de la capa oculta; finalmente, se utilizan los parámetros del modelo matemático para definir una superficie "b-spline".

En el trabajo de Yang y otros [120], se presenta una metodología de modelado 3D basado en procesamiento de imágenes y redes neuronales, cuyo objetivo es la inspección óptica automática de la calidad de la soldadura en tarjetas electrónicas. La arquitectura utilizada es una red neuronal multicapa entrenada mediante el algoritmo de retro–propagación (MLFFNN), cuyas entradas son características globales de la imagen ante diferentes condiciones y fuentes de iluminación (niveles de gris, bordes y métricas estadísticas); las salidas de la red son los pesos correspondientes a las regiones presentes en la imagen, de forma que se puede construir la superficie 3D. La validación del método se realiza comparando los resultados del modelado MLFFNN contra el método tradicional de inspección láser. La exactitud del modelo fue en promedio del 90 %.

En el trabajo de Cretu y otros [24], se presenta un esquema para la medición y representación de objetos tridimensionales deformables sin conocimiento *a priori* de su forma y/o composición. La solución propuesta combina las ventajas de las redes neuronales gas (NGN) y las redes neuronales multicapa (MLFFNN) para desarrollar un modelo capaz de ajustarse a la forma 3D del objeto y modelar sus características elásticas. Adicionalmente, las redes neuronales ofrecen salidas continuas que evitan el uso de métodos de interpolación cuando los patrones de entrada sean observaciones no presentes en el conjunto de entrenamiento de la red, y tienen potencial para usarse en aplicaciones en tiempo real debido a sus cortos tiempo de respuesta, siempre y cuando se realice un adecuado entrenamiento fuera de línea de la red. La información geométrica del objeto se obtiene mediante un escáner láser. Por medio de una NGN se genera un modelo comprimido de la nube de datos de entrada. Durante el entrenamiento, los nodos de la red codifican la geometría del objeto moviéndose asintóticamente hacia los puntos del espacio de entrada de acuerdo a la densidad de la nube de puntos. Este tipo de modelado garantiza un mayor número de nodos de representación en las zonas de mayor densidad de puntos y en las regiones de mayor variación en la forma geométrica del objeto, por lo que no solo se genera una identificación de los puntos relevantes de los datos de entrada, sino que también es posible determinar agrupaciones de puntos con características geométricas similares [20].

En [96] se presenta una aproximación de superficies basada en redes neuronales auto–organizadas e información de gradiente dentro del enfoque del álgebra geométrica. Esta metodología se aplica a una nube de puntos con el fin de encontrar una superficie que los aproxime. Para seleccionar los patrones de entrada que sirvan de conjunto de entrenamiento para la red, se emplea información del Vector de Flujo Gradiente Generalizado (GGVF),

mientras que los parámetros de la red representan diversas transformaciones en el espacio del algebra geométrica, lo que facilita transformaciones y deformaciones del modelo generado.
En el trabajo de Piastra [92], se presenta una red neuronal auto–organizada llamada SOM adaptativo (SOAM) para reconstruir curvas y superficies en el espacio 3D. Este algoritmo es una modificación del algoritmo clásico NGN que hace que la topología de la red sea curvada o superficial dependiendo de la dimensión del espacio de entrada. Este tipo de redes retiene la simplicidad computacional de los SOM ya que no requiere cálculos matemáticos de alta complejidad, sin embargo, por tratarse de una técnica en desarrollo, aún no se han aclarado la totalidad de sus fundamentos teóricos en cuanto a la convergencia del algoritmo ante datos de entrada defectuosos.
En la Figura **2-11** se muestra un ejemplo de modelado tridimensional empleando redes neuronales gas crecientes. Estas imágenes fueron tomadas de [33].

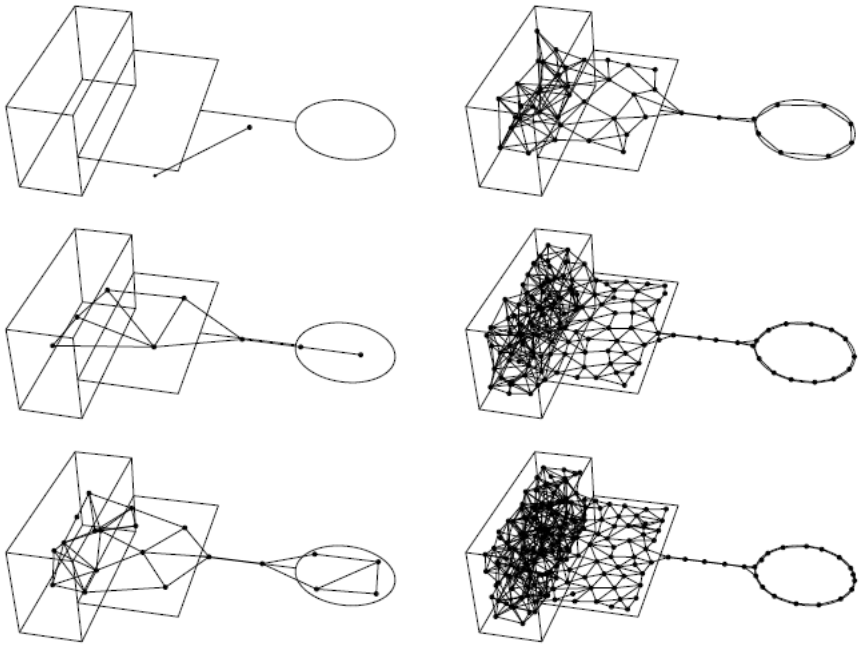


Figura 2-11.: Modelado 3D usando redes neuronales gas crecientes. Tomado de [33]

2.2.10. Resumen

La comparación entre las técnicas descritas anteriormente se puede hacer en términos de precisión, datos requeridos para el modelado, tipo de representación continua, complejidad computacional y espacio de almacenamiento en disco.
La precisión de los modelos basados en polígonos y vóxeles depende de la cantidad de elementos utilizados, donde debe existir un compromiso entre la precisión deseada y el costo computacional. En general, las representaciones de este tipo presentan una complejidad baja.

Los modelos basados en sistemas masa–resorte son fáciles de construir y pueden ser animados más sencillamente que los modelos basados en polígonos. El problema es que su precisión depende del número de elementos de representación y los valores de masa y elasticidad usados en cada uno.
Por el contrario, los modelos basados en "b–splines" requieren menos elementos de representación, lo cual es una ventaja en términos de tiempo de renderizado, sin embargo, es una representación mucho más compleja por la cantidad de parámetros requeridos.
La representación implícita es muy atractiva por el hecho de ser un modelo continuo, sin embargo, solo puede aplicarse a un número limitado de objetos.
La geometría sólida constructiva ofrece la ventaja de ser continua o discreta según se desee, el resultado es un modelo relativamente simple, aunque no funciona bien para objetos con formas complejas.
Finalmente, los modelos basados en redes neuronales artificiales pueden ofrecer representaciones continuas o discretas, volumétricas o superficiales, y con buena precisión empleando pocos recursos de memoria. Su desventaja es el alto costo computacional requerido en las etapas del entrenamiento, y la falta de una teoría bien definida sobre el diseño de las arquitecturas.

3. Adquisición de información 3D de estructuras cerebrales a partir de endoneurosonografía

La neurocirugía estereotáctica incluye el registro de imágenes médicas tanto pre–operativas como intra–operativas, típicamente en modalidades volumétricas como resonancia magnética (MRI), tomografía computarizada (CT) y ultrasonido (US), dentro de un sistema coordenado arbitrario [38]. En la mayoría de intervenciones quirúrgicas, el sistema coordenado se define por un conjunto de retículas (x, y, z) sobre un marco fijado al cráneo del paciente. En los sistemas modernos de cirugía asistida por ordenador, el marco se sustituye por un sistema de seguimiento (por lo general óptico, mecánico o magnético), que el equipo utiliza para rastrear las herramientas quirúrgicas, las cuales se presentan en un entorno virtual para proporcionar una guía al cirujano durante la intervención [39]. Con el fin de obtener buenos resultados, el cerebro no debe moverse de manera crítica durante la cirugía, lo cual sólo es posible mediante el uso de cirugía mínimamente invasiva, es decir, realizar la operación a través de un pequeño orificio en el cráneo [97].

Las tendencias recientes en la cirugía mínimamente invasiva del cerebro han sugerido la utilización conjunta de imágenes endoscópicas y de ultrasonido, una técnica llamada Endoneurosonografía (ENS) [76], debido a que es una técnica más barata que otras de imagenología médica como CT y MRI, y además, es más fácil de adquirir en un escenario intra–operativo [107]. En la literatura, se han reportado algunos trabajos que emplean ENS para extraer información tridimensional de estructuras cerebrales [39]. La Figura **3-1** muestra las imágenes que componen una secuencia de adquisición endoneurosonográfica.

Debido a sus ventajas, en este trabajo se emplea endoneurosonografía para la representación de estructuras internas del cerebro, específicamente tumores. En este capítulo, se plantea una metodología para la adquisición de información tridimensional de estructuras cerebrales utilizando endoneurosonografía.

La ENS es una técnica de adquisición 3D que combina la visión endoscópica estéreo y el ultrasonido 2D. Las imágenes endoscópicas se emplean para hacer una reconstrucción estéreo del escenario, en el que la sonda de ultrasonido es el elemento de mayor interés, mientras que las imágenes de ultrasonido se emplean para ubicar regiones biológicas, tales como tumores. Adicionalmente, existe un sistema de seguimiento visual externo (POLARIS) que permite conocer la pose del endoscopio durante la cirugía, por lo que la información 3D extraída

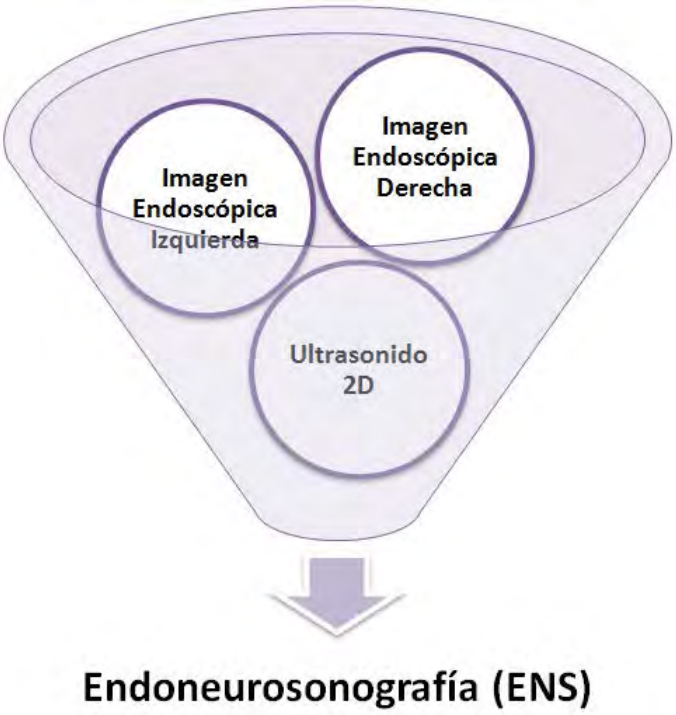


Figura 3-1.: Imágenes que componen la endoneurosonografía

está referenciada respecto a un sistema quirúrgico de coordenadas. Con esta metodología se pretenden resolver cinco problemas específicos: a) calibración del sistema endoscópico estéreo, b) calibración seguidor–cámara , comúnmente llamada calibración "hand–eye", entre el sistema de seguimiento visual y las cámara endoscópicas, c) seguimiento de la sonda de ultrasonidos a través de las imágenes endoscópicas, d) segmentación de estructuras cerebrales a partir de imágenes de ultrasonido, y e) extracción de información 3D a partir de la reconstrucción estéreo del escenario y las regiones segmentadas en las imágenes de ultrasonido.

En la Figura **3-2** se muestra un esquema con los problemas a resolver mediante la metodología propuesta, y a continuación e hace una breve descripción de cada uno de ellos.

En el primer paso, el sistema endoscópico se calibra intrínseca y extrínsecamente con el objetivo de caracterizar completamente el sistema de adquisición y poder hacer una reconstrucción estéreo del escenario intra–operativo. El sistema de visión estéreo se compone de dos cámaras CCD miniaturas que deben ser calibradas y rectificadas con el fin de extraer información 3D a partir de cada imagen 2D. Con este fin, se utilizó la técnica de calibración propuesta por Zhang [123], la cual sólo requiere de un patrón de calibración plano y al menos dos capturas desde orientaciones diferentes de la cámara. No es necesario que la trayectoria de movimiento sea conocida y lo que importa es el movimiento relativo, por lo que es indiferente si es el patrón o la cámara lo que se mueve.

En el segundo paso, se aborda el problema de la calibración "hand–eye" que permite estimar

Figura 3-2.: Problemas a resolver con la metodología de adquisición propuesta

la transformación geométrica desde el sistema de seguimiento visual POLARIS hasta el sistema coordenado de las cámaras endoscópicas. Esta transformación permite que la información endoscópica adquirida se referencie con respecto a un punto del mundo real.

En el tercer paso, se realiza un seguimiento de la sonda de ultrasonido a través de una secuencia de imágenes endoscópicas y se calcula su pose en el espacio 3D empleando únicamente información de las imágenes, es decir, no se requiere ningún sistema de seguimiento externo (ni óptico ni magnético). Para obtener un seguimiento confiable de la sonda, se aprovecha la información contextual de la escena como el movimiento de la sonda y el material de construcción de la misma. En este sentido, se propone una metodología de segmentación basada en flujo óptico y luminancia. A continuación, se aplica Análisis de Componentes Principales (PCA) sobre la región segmentada con el fin de determinar el eje principal de la sonda. Por último, se define una restricción de movimiento a fin de evitar errores de seguimiento. La determinación y seguimiento del eje principal de la sonda de ultrasonido es una de las tareas esenciales en la reconstrucción 3D, dado que las imágenes de ultrasonido que captura el sensor se ubican en el plano perpendicular a dicho eje.

En el cuarto paso, se segmentan las estructuras de interés (tumores) en las imágenes de ultrasonido. Se emplean técnicas morfológicas con el fin de rellenar huecos y eliminar ruido causado por el sub–muestreo y el ruido "speckle" propio del sistema de ultrasonido utilizado [83]. Por último, se aplica un umbral automático utilizando el método de Otsu para segmentar las regiones de interés [90].

En el último paso, las metodologías de las etapas anteriores se integran con el fin de hacer una reconstrucción 3D de las estructuras del cerebro con respecto a un sistema coordenado dentro de la sala de operación quirúrgica. Para lograr esto, se debe calibrar el equipo de adquisición, calcular la pose de la sonda de ultrasonido en el espacio 3D a partir de las

imágenes endoscópicas, extraer el plano perpendicular al eje de la sonda (que contiene las imágenes de ultrasonido), segmentar las regiones de interés en las imágenes de ultrasonido, y finalmente, localizar estas regiones en el espacio coordenado de referencia.
La configuración del equipo de adquisición endoneurosonográfica es el siguiente: Se dispone de una camisa de trabajo con varios canales por los que se introducen los elementos quirúrgicos, como se muestra en la Figura **3-3**. El endoscopio estéreo y la sonda de ultrasonido se introducen por alguno de los canales de la camisa de trabajo, de manera que la sonda sea vista por las dos cámaras endoscópicas. Un sistema de seguimiento visual (POLARIS) [87], instalado en la sala de cirugía como se muestra en la Figura **3-4**, entrega la pose 3D de la punta del endoscopio. Adicionalmente, con la información de las imágenes endoscópicas se requiere calcular la pose del sensor de ultrasonido, ubicado en la punta de la sonda. El seguimiento de la pose de la sonda es una tarea delicada debido a que la sonda es flexible, gira sobre su propio eje y puede moverse hacia adelante y hacia atrás a través del canal. Después de que el seguimiento es computado, es posible hacer la reconstrucción tridimensional de esta información. Para esto, deben conocerse las transformaciones de rotación y traslación desde el sistema coordenado de la imagen de ultrasonido hasta el sistema coordenado de referencia en el mundo real. Con estas transformaciones, la información 3D del tumor se referencia respecto a la posición del sistema POLARIS dentro de la sala de cirugía.

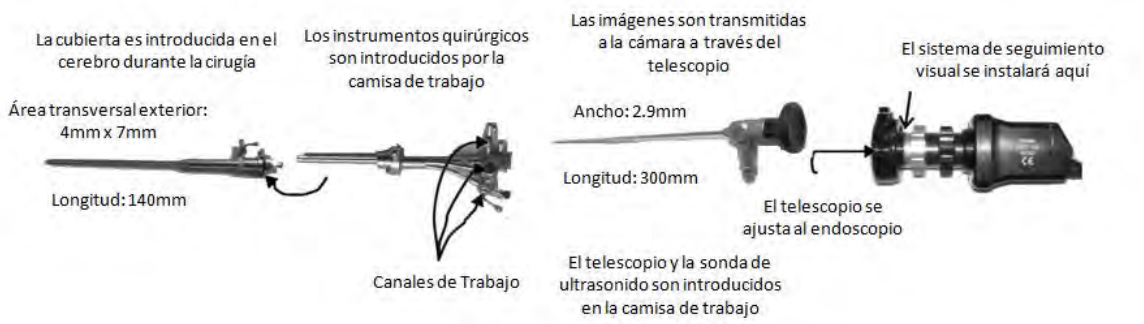
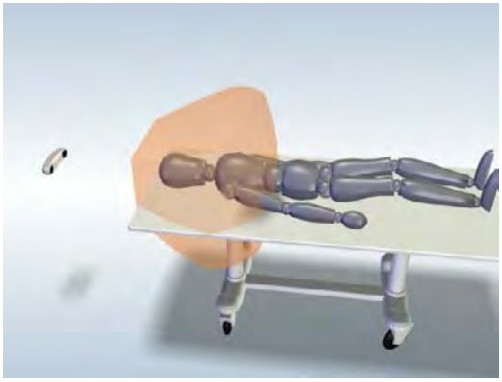
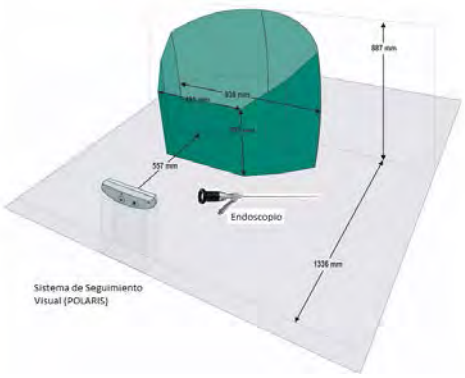


Figura 3-3.: Sistema de adquisición endoneurosonográfica

Este capítulo está organizado como sigue: la Sección 3.1 describe el montaje experimental utilizado en la adquisición endoneurosonográfica y discute los aspectos técnicos del equipo. Las Secciones 3.2 y 3.3 explican las técnicas de calibración utilizadas para la caracterización de las cámaras endoscópicas y del sistema seguidor–cámaras. La Sección 3.4 presenta las metodologías de segmentación y seguimiento espacio–temporal de la sonda de ultrasonido y la determinación de la pose de su eje a través de imágenes endoscópicas. Adicionalmente, se presentan los experimentos realizados sobre una base de datos endoneurosonográfica propia y se argumenta la utilización de una restricción de movimiento para el seguimiento de la sonda. La Sección 3.5 ilustra las técnicas morfológicas empleadas para la segmentación de estructuras cerebrales en imágenes de ultrasonido. La Sección 3.6 aplica el marco teórico–práctico de las secciones anteriores para presentar formalmente la metodología de adquisición



(a) Rango de Acción



(b) Seguimiento visual del endoscopio

Figura 3-4.: Sistema de seguimiento visual POLARIS. Tomado de [87]

de información 3D de estructuras cerebrales a partir de endoneurosonografía. Finalmente, la Sección 3.7 está dedicada a resumir y concluir el capítulo.

3.1. Sistema endoneurosonográfico

El equipo ENS mostrado en la Figura **3-3** se compone de una camisa de trabajo, un endoscopio estéreo con un telescopio acoplado y una sonda de ultrasonido radial. El endoscopio estéreo captura simultáneamente dos imágenes de color mediante dos sensores CCD miniaturas. El telescopio se introduce por uno de los canales de trabajo de la camisa y transmite las imágenes cerebrales hasta el endoscopio. Por último, la sonda de ultrasonido radial obtiene imágenes US de las estructuras cerebrales. La sonda de ultrasonido tiene conectado un micro transductor que gira sobre su propio eje para generar un haz de 360° a 10 MHz. La imagen que captura el sensor US es ortogonal al eje de la sonda [76, 101], por lo que hacer el seguimiento de su pose a través de las imágenes de ultrasonido es fundamental para realizar la triangulación estéreo que permita computar la pose 3D de la sonda US con respecto a las coordenadas del endoscopio. En la Figura **3-5** se muestra la sonda de ultrasonido, la cual gira sobre su propio eje por medio de un motor.
En las siguientes secciones se presentan las técnicas usadas para la calibración estéreo de las cámaras endoscópicas y la calibración “hand–eye” del sistema seguidor–cámaras.

3.2. Calibración de cámaras

El proceso de visión por computador se inicia con la detección de la luz que proviene de alguna fuente y se refleja en los objetos del escenario. La luz reflejada pasa por la lente de la cámara y se recoge en su sensor fotoelectrónico [11].
El modelo de cámara más simple es el de la cámara estenopeica (“pinhole”), el cual se

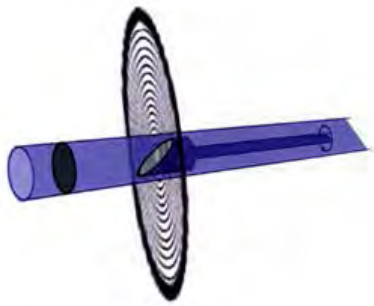


Figura 3-5.: Sonda de ultrasonido con su haz de emisión radial. Tomado de [95]

compone de una pared imaginaria con un pequeño agujero en el centro que bloquea todos los rayos de luz, excepto los que pasan a través de la pequeña abertura del centro. Este modelo de cámara es útil para entender la geometría básica de los rayos de luz, sin embargo, la pequeña abertura no recoge suficiente luz para tomar imágenes con corto tiempo de exposición. Por este motivo, las cámaras utilizan lentes que reúnen más luz que la que estaría disponible en un solo punto, por lo que su caracterización no sólo debe incluir parámetros geométricos de la cámara, sino también los parámetros de distorsión de la lente.

En la visión por computador, la calibración de la cámara es el proceso mediante el cual se determinan las características internas de la cámara (parámetros intrínsecos), y la pose tridimensional de la cámara con respecto a un sistema de coordenadas determinado (parámetros extrínsecos) [115].

El sistema de visión estéreo utilizado en este trabajo se compone de dos cámaras endoscópicas que deben ser calibradas con el fin de extraer información 3D del escenario a partir de cada par de imágenes 2D. En general, las técnicas de calibración pueden clasificarse en dos categorías: a) calibración fotogramétrica, la cual se realiza mediante la observación de un patrón de calibración cuya geometría en el espacio 3D se conoce con muy buena precisión. Por lo general, el objeto de calibración consta de dos o tres planos ortogonales entre sí [110]; y, b) auto–calibración, que no utiliza ningún objeto de calibración. Los parámetros de la cámara se calculan a partir de diferentes puntos de vista de una escena estática [74].

En este trabajo se utiliza la técnica de calibración propuesta por Zhang [123], la cual requiere que la cámara capture imágenes de un patrón de calibración plano en al menos dos orientaciones diferentes. O bien la cámara o el patrón de plano se puede mover con la mano y el movimiento no tiene por qué conocerse. Este enfoque es un método híbrido entre los dos tipos de calibración existentes, ya que, al contrario que la auto–calibración, sí se requiere un patrón de calibración, pero este patrón no es un arreglo de patrones 3D, como en la fotogrametría, sino que es un patrón plano. La elección de esta técnica de calibración se debe a su facilidad de implementación, velocidad y buenos resultados reportados en la literatura [10, 12].

3.2.1. Parámetros intrínsecos

El modelo de cámara utilizado en este trabajo está inspirado en el modelo propuesto por Heikkila y Silven [53]. La cámara se modela mediante cuatro conjuntos de parámetros intrínsecos: la *distancia focal* ($fc \in \Re^2$), el *punto principal* ($cc \in \Re^2$) o también llamado centro de imagen, el *coeficiente de inclinación* ($\alpha \in \Re^1$), y los *coeficientes de distorsión* ($kc \in \Re^5$).
Sea P un punto en el espacio con coordenadas $(Xc, Yc, Zc)^T$ en el mismo marco de referencia de la cámara, donde $(x, y)^T = (Xc/Zc, Yc/Zc)^T$ es su proyección estenopeica. Usando los modelos de distorsión radial y tangencial propuestos por Brown [14], se define la corrección de distorsión de la Ecuación 3-1 para las coordenadas normalizadas $(x_d, y_d)^T$, donde $r^2 = x^2+y^2$.

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \left(1 + kc_1 r^2 + kc_2 r^4 + kc_5 r^6\right) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2kc_3 xy + kc_4 \left(r^2 + 2x^2\right) \\ kc_3 \left(r^2 + 2y^2\right) + 2kc_4 xy \end{bmatrix} \tag{3-1}$$

Una vez que se aplica la corrección de distorsión, las coordenadas $(x_p, y_p)^T$ de los píxeles correspondientes a la proyección de P sobre el plano de la imagen se relacionan a través de la Ecuación 3-2.

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} fc_1 & \alpha fc_1 & cc_1 \\ 0 & fc_2 & cc_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \tag{3-2}$$

3.2.2. Parámetros extrínsecos

Los parámetros extrínsecos de la cámara son las transformaciones geométricas que permiten pasar del sistema coordenado del plano de la imagen al sistema coordenado centrado en el punto principal de la cámara. La transformación rígida desde el sistema de coordenadas del plano de la imagen $(x_w, y_w, z_w)^T$ hasta el sistema coordenado de la cámara $(x, y, z)^T$ se muestra en la Ecuación 3-3. Esta transformación es única y se define como tres rotaciones separadas (Elevación, Cabeceo y Balanceo), seguidas de una traslación 3D. Los parámetros estimados tras la calibración son: la *Matriz de Rotación* (R) y el *Vector de Traslación* (T) [110].

$$\begin{bmatrix} x & y & z \end{bmatrix}^T = R \begin{bmatrix} x_w & y_w & z_w \end{bmatrix}^T + T \tag{3-3}$$

3.2.3. Calibración estéreo

Con el fin de calibrar el sistema endoscópico, se utilizó el Toolbox de MATLAB para calibración de cámaras desarrollado por Bouguet [10]. Este Toolbox permite calibrar intrínseca y extrínsecamente las dos cámaras utilizando el procedimiento propuesto por Zhang [123]. Después de cada calibración individual, se calculan los parámetros intrínsecos y extrínsecos

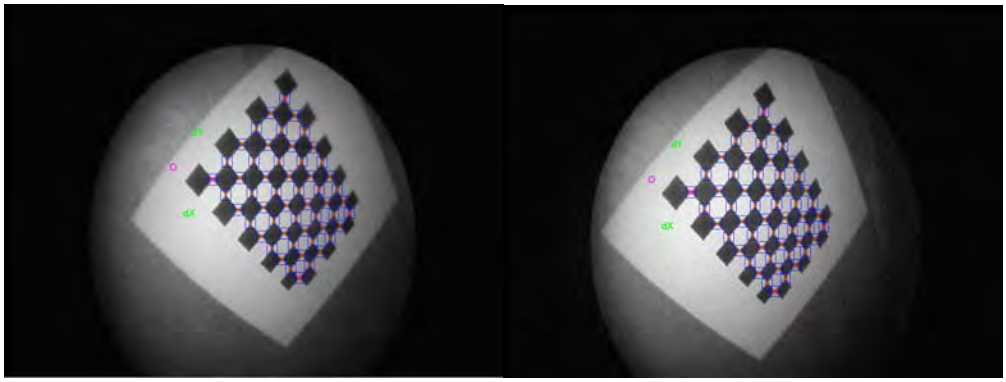
del sistema estéreo. Para esta calibración se aplica una optimización estéreo global sobre un conjunto mínimo de parámetros desconocidos. En particular, sólo se considera una pose desconocida (6 grados de libertad: 3 de posición y 3 de orientación) para la ubicación del patrón de calibración. Esto asegura la rigidez estructural de la transformación desde el sistema de coordenadas de la cámara izquierda hasta el de la cámara derecha. De forma predeterminada, la optimización estéreo reajusta los parámetros intrínsecos de las cámaras antes de iniciar la calibración.

La correspondencia entre cada par de imágenes es muy importante, y significa que el mismo conjunto de puntos debe seleccionarse tanto en la imagen izquierda como en la derecha. Para esta calibración se probaron dos patrones de calibración diferentes: el primero, el clásico patrón de ajedrez de las Figuras **3-6**a y **3-6**b cuyas medidas son dx=dy=5 mm; y el segundo, un patrón de puntos mostrado en las Figuras **3-6**c y **3-6**d, diseñado por Wengert y otros, especialmente para la calibración de endoscopios [119]. Los puntos están equidistanciados 2 mm vertical y horizontalmente. Después de la calibración, es posible calcular la posición 3D del conjunto de puntos de calibración dadas las proyecciones en las imágenes izquierda y derecha, como se muestra en la Figura **3-6**. Para este cálculo, son necesarios los parámetros intrínsecos.

Las Tablas **3-1** y **3-2** muestran los parámetros intrínsecos y extrínsecos del sistema endoscópico después de la calibración, respectivamente. Los errores numéricos reportados corresponden a tres veces la desviación estándar. Las Figuras **3-6**e se muestra la representación gráfica de los parámetros extrínsecos en un espacio tridimensional en el que el origen coordenado se ha definido arbitrariamente en el punto principal de la cámara derecha. En la Figura **3-6**f se muestra una representación equivalente en el que el origen coordenado se ha definido en el centro del patrón de calibración.

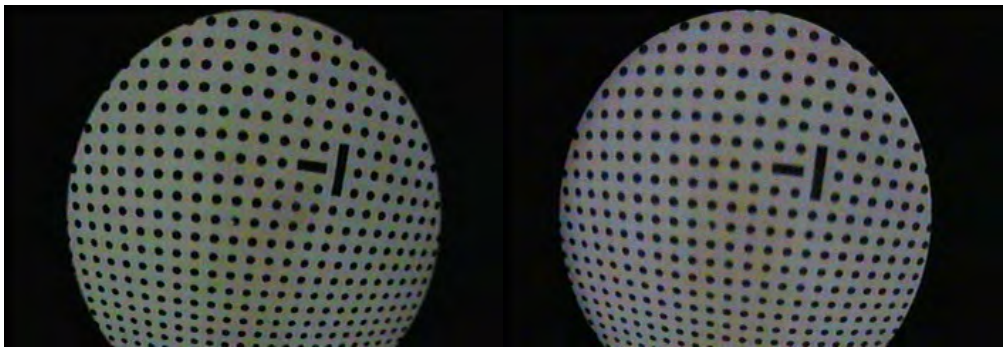
Tabla 3-1.: Parámetros intrínsecos del sistema endoscópico estéreo

	Cámara Izquierda	**Cámara Derecha**
$fc =$	$\begin{bmatrix} 164{,}33 & 177{,}16 \end{bmatrix}^T \pm \begin{bmatrix} 14{,}16 & 15{,}36 \end{bmatrix}^T$	$\begin{bmatrix} 230{,}74 & 220{,}04 \end{bmatrix}^T \pm \begin{bmatrix} 10{,}23 & 13{,}93 \end{bmatrix}^T$
$cc =$	$\begin{bmatrix} 191{,}91 & 133{,}37 \end{bmatrix}^T \pm \begin{bmatrix} 9{,}79 & 10{,}68 \end{bmatrix}^T$	$\begin{bmatrix} 183{,}86 & 196{,}90 \end{bmatrix}^T \pm \begin{bmatrix} 7{,}97 & 14{,}29 \end{bmatrix}^T$
$\alpha =$	$90{,}00 \pm 0{,}00$	$90{,}00 \pm 0{,}00$
$kc =$	$\begin{bmatrix} -0{,}37 & 0{,}15 & 0{,}01 & 0{,}02 & 0{,}00 \end{bmatrix}^T$ $\pm \begin{bmatrix} 0{,}09 & 0{,}19 & 0{,}01 & 0{,}02 & 0{,}00 \end{bmatrix}^T$	$\begin{bmatrix} -0{,}75 & 0{,}23 & -0{,}13 & -0{,}04 & 0{,}00 \end{bmatrix}^T$ $\pm \begin{bmatrix} 0{,}12 & 0{,}11 & 0{,}04 & 0{,}02 & 0{,}00 \end{bmatrix}^T$
	fc: *distancia focal* α: *coeficiente de inclinación*	*cc*: *punto principal* *kc*: *coeficientes de distorsión*



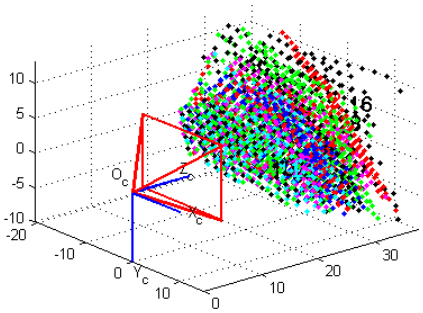
(a) Patrón Ajedrez: Vista Cámara Izquierda

(b) Patrón Ajedrez: Vista Cámara Derecha

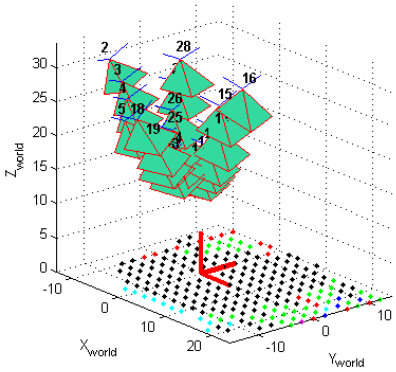


(c) Patrón Puntos: Vista Cámara Izquierda

(d) Patrón Puntos: Vista Cámara Derecha



(e) Parámetros Extrínsecos. Centro en la cámara



(f) Parámetros Extrínsecos. Centro en el patrón

Figura 3-6.: Patrón plano empleado para la calibración del endoscopio estéreo

Tabla 3-2.: Parámetros extrínsecos del sistema endoscópico estéreo

Matriz de Rotación	
$R =$	$\begin{bmatrix} 0{,}988 & -0{,}002 & -0{,}153 \\ -0{,}036 & 0{,}970 & -0{,}242 \\ 0{,}149 & 0{,}245 & 0{,}958 \end{bmatrix} \pm \begin{bmatrix} 0{,}9981 & -0{,}007 & 0{,}061 \\ 0{,}012 & 0{,}998 & -0{,}074 \\ -0{,}060 & 0{,}075 & 0{,}996 \end{bmatrix}$
Vector de Traslación	
$T =$	$\begin{bmatrix} 7{,}1879 & -5{,}7485 & 25{,}3582 \end{bmatrix}^T \pm \begin{bmatrix} 1{,}7641 & 2{,}6040 & 6{,}7322 \end{bmatrix}^T$

3.3. Calibración seguidor–cámara ("hand–eye")

El problema de la calibración "hand–eye" es un problema de robótica en el que se tiene una cámara ("Eye") montada sobre un manipulador robótico ("Hand"). El problema consiste en estimar la transformación geométrica desde el sistema coordenado de la cámara hasta el sistema coordenado del manipulador. De forma general, este problema de calibración puede extenderse a 3 aplicaciones:

- Cámara instalada sobre el efector final de un manipulador robótico.
- Cámara con sensor de movimiento y posicionamiento interno: giroscopio o acelerómetro.
- Cámara con un sistema de seguimiento externo: óptico, magnético o mecánico.

Como se explica en la Sección 3.1, la configuración experimental del sistema de adquisición ENS incluye un sistema de seguimiento visual externo (POLARIS) que entrega la pose del endoscopio con respecto a un sistema de referencia en el mundo real. En este caso, la calibración "hand–eye" es computada a partir de la información intrínseca y extrínseca de las cámaras, y la información de seguimiento que entrega el sistema POLARIS. El resultado es la transformación geométrica desde la pose del endoscopio hasta la pose de las cámaras endoscópicas.
En la Figura **3-7** se muestra el problema de calibración como la búsqueda de las transformaciones geométricas H_i^j que permiten pasar del sistema coordenado j al sistema coordenado i. El objetivo final, es encontrar la transformación $H_{patrón}^{mundo}$ como la aplicación sucesiva de las transformaciones $H_{marcador}^{mundo}$, $H_{cámara}^{marcador}$ y $H_{patrón}^{cámara}$. La matriz de transformación homogénea $H_{marcador}^{mundo}$ está dada por el sistema de seguimiento visual, mientras que la matriz $H_{patrón}^{cámara}$ corresponde a los parámetros extrínsecos de las cámaras endoscópicas obtenidos con el procedimiento de la Sección 3.2.

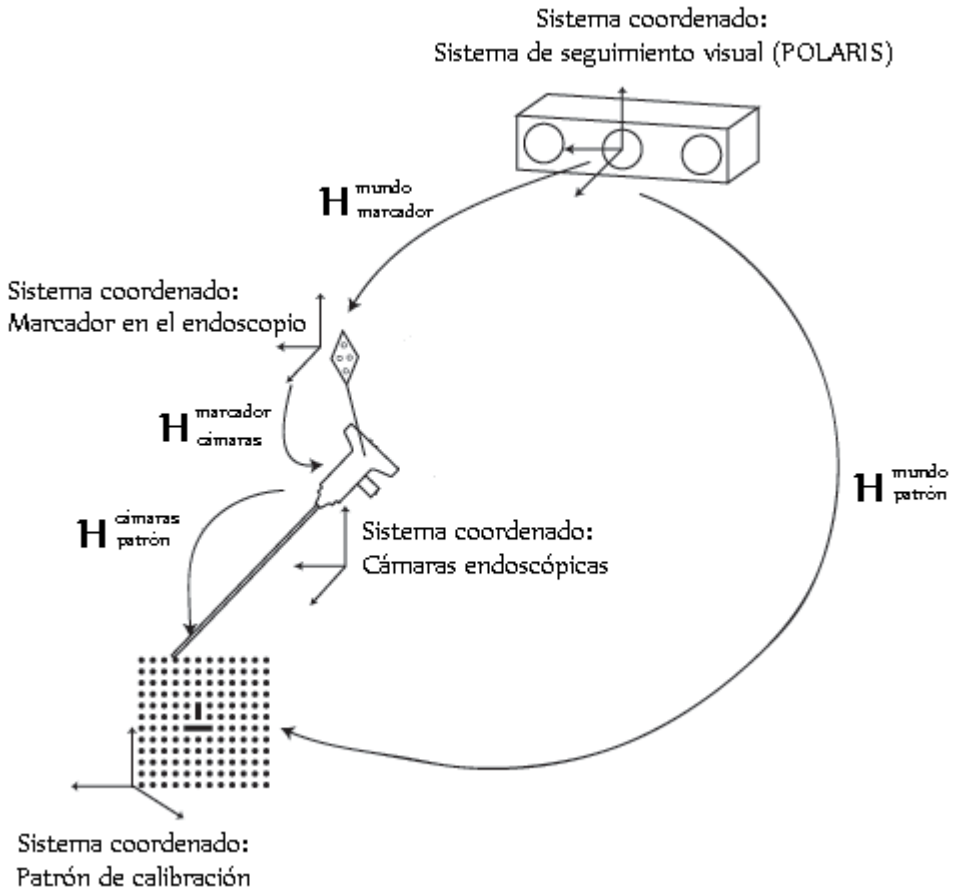


Figura 3-7.: Calibración seguidor–cámara ("hand–eye"). Adaptado de [30]

En la literatura se han planteado varias soluciones a este problema de calibración. El enfoque clásico consiste en dividir el problema en dos partes: primero, descomponer la matriz $H^{marcador}_{cámara}$ en su componente rotacional y traslacional, y entonces optimizar la primera componente y luego la segunda [111, 102]. Existen otras soluciones que determinan simultáneamente las componentes de rotación y traslación usando cuaterniones [25, 26], y trabajos en los que se plantean metodologías basadas en movimiento que no requieren el uso de patrones de calibración [99].

La calibración permite establecer la relación $H^{marcador}_{cámara}$ que define la transformación desde la pose de la cámara endoscópica hasta la pose del marcador en el endoscopio. Para esto, se establece la ecuación lineal $A[H^{marcador}_{cámara}] = [H^{marcador}_{cámara}]B$, donde A es el movimiento relativo entre el marcador y el sistema de seguimiento, y B es el movimiento relativo entre la cámara y el patrón de calibración, tal como se muestra en la Ecuación 3-4.

$$
\begin{aligned}
A &= [H^{mundo}_{marcador}(t_0)] \cdot [H^{marcador}_{mundo}(t_1)] \\
B &= [H^{cámara}_{patrón}(t_0)] \cdot [H^{patrón}_{cámara}(t_1)]
\end{aligned}
\tag{3-4}
$$

Este problema se resuelve mediante la metodología propuesta por Tsai y Lenz [112], usando el toolbox de calibración de cámaras de MATLAB [10, 119, 52].

La Ecuación 3-5 muestra los resultados de calibración "hand–eye" para una sesión de adquisición ENS. Cabe aclarar que estos parámetros son diferentes en cada sesión de adquisición debido a que el equipo debe ser recalibrado antes de su funcionamiento.

$$
H^{cámara}_{marcador} = \begin{bmatrix} 0{,}9868 & -0{,}1155 & 0{,}1133 & 47{,}0462 \\ 0{,}1597 & 0{,}8077 & -0{,}5675 & 126{,}2160 \\ -0{,}0260 & 0{,}5781 & 0{,}8155 & 658{,}3127 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{3-5}
$$

3.4. Procesamiento de imágenes endoscópicas

En cada instante de tiempo se obtienen dos imágenes de color de cada cámara endoscópica y una imagen de ultrasonido de la sonda US. Las cámaras están completamente caracterizadas (intrínseca y extrínsecamente), por lo que es posible realizar una reconstrucción estéreo de la escena quirúrgica. Las imágenes endoscópicas tienen una dimensión de 352×240 píxeles. El objeto de interés es la sonda de ultrasonido, que es visible por las cámaras endoscópicas. La sonda es metálica (superficie especular), gira sobre su propio eje y se mueve hacia delante y hacia atrás a través del canal de trabajo, por lo que la luminancia y el flujo óptico son características importantes para realizar su segmentación y su seguimiento.

En general, los métodos de seguimiento visual de objetos en visión por computador se dividen en dos clases: enfoques de *arriba hacia abajo* ("top–down") y de *abajo hacia arriba* ("bottom–up"). El enfoque *arriba hacia abajo* genera hipótesis de objetos en la escena y trata de verificarlas usando información de la imagen. Por ejemplo, el *filtro de partículas* sigue este enfoque en el sentido de que las hipótesis se evalúan sólo en algunas posiciones de prueba de la imagen [88, 89]. Por otra parte, el enfoque *abajo hacia arriba* consiste en segmentar los objetos de interés en cada imagen disponible con el fin de computar el seguimiento. En este trabajo se propone una metodología espacio–temporal de seguimiento de la sonda de ultrasonido que corresponde a un enfoque *abajo hacia arriba*.

3.4.1. Segmentación espacio–temporal de la sonda de ultrasonido

El objetivo es determinar la ubicación de la sonda de ultrasonidos a través de las imágenes endoscópicas. Debido a los movimientos de la sonda y al material especular del que está construida, el flujo óptico y la luminancia son las características utilizadas para realizar la segmentación. En la Figura **3-8** se muestra la metodología espacio–temporal de segmentación basada en flujo óptico, luminancia y agrupamiento [36, 101].

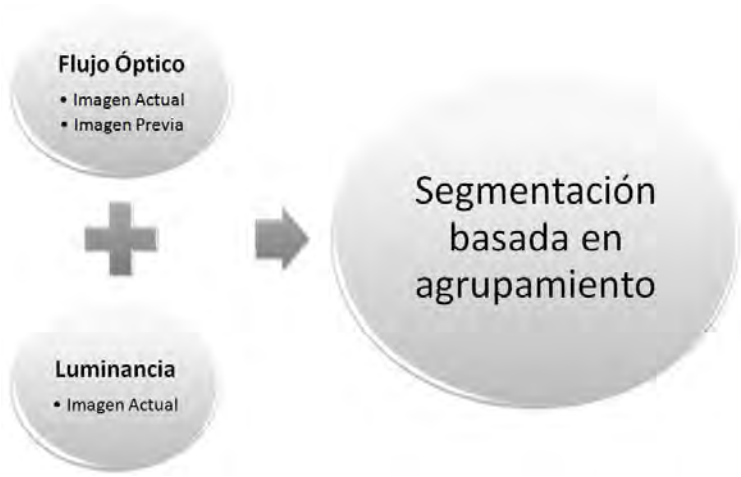


Figura 3-8.: Metodología espacio–temporal de segmentación de la sonda de ultrasonido

3.4.1.1. Flujo Óptico

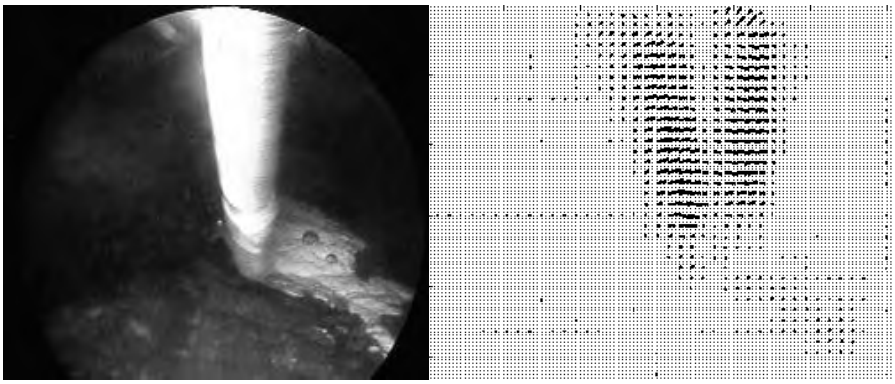
En este trabajo se utiliza el algoritmo propuesto por Horn y Schunck para computar el flujo óptico entre imágenes consecutivas de la secuencia endoscópica [56]. Este algoritmo determina el flujo óptico como una solución de la ecuación diferencial parcial que se muestra en la Ecuación 3-6, donde $L(x, y, t)$ representa la luminancia del píxel (x, y) en el instante t. Este algoritmo hace la suposición de que los patrones de brillo no varían en intervalos de tiempo corto, y lo único que varía es su posición (x, y) dentro de la imagen.

$$\frac{\partial L}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial L}{\partial y}\frac{\partial y}{\partial t} + \frac{dL}{dt} = 0 \tag{3-6}$$

La solución de la Ecuación 3-6 se obtiene minimizando una función de error mediante la aplicación de métodos numéricos. La función de error E se define en términos de gradientes espaciales y temporales del campo de vector de flujo, que se compone de los términos L_c y L_b que se muestran en la Ecuación 3-7.

$$
\begin{aligned}
&E = \int\int [\alpha^2 L_c^2 + L_b^2]\, dx\, dy \\
&L_b = \tfrac{\partial L}{\partial x} u + \tfrac{\partial L}{\partial y} v + \tfrac{dL}{dt} \\
&L_c^2 = \left(\tfrac{\partial u}{\partial x}\right)^2 + \left(\tfrac{\partial u}{\partial y}\right)^2 + \left(\tfrac{\partial v}{\partial x}\right)^2 + \left(\tfrac{\partial v}{\partial y}\right)^2 \\
&u = \tfrac{dx}{dt} \quad ; \quad v = \tfrac{dy}{dt}
\end{aligned}
\tag{3-7}
$$

Para resolver el problema de minimización se utiliza un método acelerado de gradiente descendente para calcular la mejor dirección de búsqueda del mínimo de la función. El algoritmo de flujo óptico tiene dos fases principales: en la primera, los coeficientes de gradiente de $\partial L/\partial x$, $\partial L/\partial y$, dL/dt se calculan a partir de las imágenes de entrada; en la segunda, se calculan los vectores de flujo óptico u y v, definidos por la Ecuación 3-7.
La Figura **3-9** muestra el cálculo del flujo óptico para una imagen endoscópica. Nótese que los valores máximos de flujo corresponden a la región de la imagen en que la sonda de ultrasonido se está moviendo.



(a) Imagen Endoscópica (b) Flujo Óptico

Figura 3-9.: Cálculo del flujo óptico empleando el algoritmo de Horn y Schunck

3.4.1.2. Segmentación basada en agrupamiento

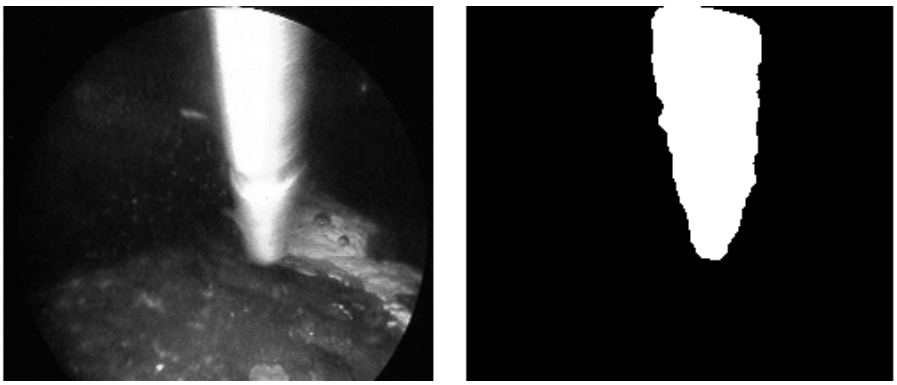
La mayoría de las técnicas clásicas de segmentación tienen en cuenta únicamente información de la imagen actual para segmentar objetos de interés [113]. Sin embargo, el movimiento relativo del objeto con respecto al fondo es una característica útil para segmentar objetos no estáticos dentro de una imagen [36]. En este caso, el objeto que deseamos segmentar es la sonda de ultrasonido, la cual gira sobre su propio eje y se mueve hacia adelante y hacia atrás a través del canal de trabajo. En este enfoque, la segmentación no se realiza de manera simple empleando información de un único fotograma, sino que utiliza información de la imagen actual y de las imágenes precedentes. Con este objetivo, se construye un espacio de dos características: la primera, la luminancia de la imagen actual, debido a que la sonda

de ultrasonido está construida de un material especular que hace que sea más brillante que el resto de objetos de la escena; y la segunda, la magnitud del flujo óptico, dado que la sonda de ultrasonido es el objeto con más movimiento relativo dentro de la escena. Con estas características se obtiene tanto información espacial como temporal de la imagen.
Una vez construido el espacio de características, se aplica el algoritmo de agrupamiento de "k–means clustering" [100], el cual es un método numérico, no supervisado, no determinístico e iterativo, para clasificar datos de entrada en grupos naturales. Es simple, rápido, efectivo en tareas de clasificación y ampliamente utilizado en aplicaciones prácticas [86]. El algoritmo "k–means" consiste en dividir iterativamente el espacio de características en agrupamientos naturales con el fin de minimizar la suma, sobre todos los agrupamientos, de las distancias del centroide de cada región a cada punto del espacio de entrada. En este caso, el espacio de características se divide en dos agrupamientos correspondientes a dos regiones de la imagen: la sonda de ultrasonido y el fondo. Después de que se aplica el algoritmo de agrupamiento, la imagen es abierta morfológicamente con el fin de reducir el ruido y eliminar las pequeñas regiones que no son de interés [60]. La Figura **3-10** muestra este esquema de segmentación. La Figura **3-11** muestra el resultado de la segmentación mediante el procedimiento descrito anteriormente.

Figura 3-10.: Esquema de segmentación de la sonda US usando agrupamiento y morfología

3.4.2. Cálculo del eje de la sonda de ultrasonido

El objetivo es determinar el eje de la sonda de ultrasonido a través de las imágenes endoscópicas. Después de que se realiza la segmentación explicada en la subsección anterior, se obtiene un conjunto de puntos (x, y) de píxeles pertenecientes a la región de la sonda de



(a) Imagen Endoscópica (b) Segmentación de la sonda US

Figura 3-11.: Segmentación de la sonda de ultrasonido empleando agrupamiento

ultrasonido dentro de la imagen. Con el propósito de obtener la línea del eje de la sonda, se extrae la primera componente principal (PC) de la región segmentada mediante PCA [42].

PCA consiste en calcular los ejes de mayor dispersión o vectores propios de una nube de datos n–dimensional. En el caso de imágenes, se puede utilizar para alinear objetos, regiones o fronteras mediante dichos vectores [113].

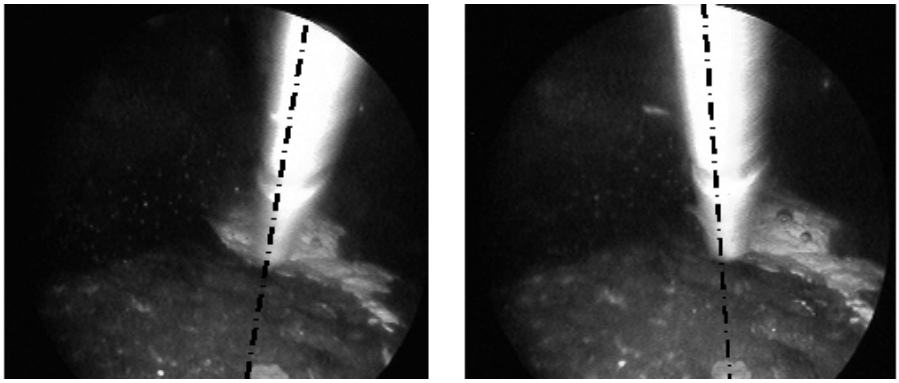
La sonda de ultrasonido tiene una forma alargada y delgada, por lo que su eje longitudinal corresponde con el eje que se quiere encontrar. Si se entiende la región segmentada de la sonda US como un conjunto de datos bivariados (x_1, x_2), el eje longitudinal es el que tiene mayor dispersión de datos. Por esta razón, el cálculo de la primera PC de los píxeles en la región segmentada corresponde a la línea del eje de la sonda. Con la información del eje de la sonda US se hace seguimiento de su pose a través de las imágenes endoscópicas.

Considérese la variable $X' = [x_1, x_2]$ que corresponde a las coordenadas cartesianas de los píxeles que conforman la región de la sonda US segmentada en la imagen, la matriz de covarianza Σ y valores propios $\lambda_1 \geq \lambda_2$. Con esta información, se puede construir la combinación lineal que se muestra en la Ecuación 3-8. El Análisis de Componentes Principales consiste en hallar la combinación lineal $Y' = [y_1, y_2]$ que maximiza la varianza $Var(y_i)$ y la covarianza $Cov(y_i, y_k)$ de los datos (Ver Ecuación 3-9). El resultado del PCA es una matriz de dos elementos con las componentes de mayor y segunda mayor varianza, respectivamente [46].

$$\begin{aligned} y_1 &= a_1 X = a_{11}x_1 + a_{12}x_2 \\ y_2 &= a_2 X = a_{21}x_1 + a_{22}x_2 \end{aligned} \tag{3-8}$$

$$\begin{aligned} Var(y_i) &= a_i' \sum a_i \quad ; \quad i = 1, 2 \\ Cov(y_i, y_k) &= a_i' \sum a_k \quad ; \quad i = 1, 2 \end{aligned} \tag{3-9}$$

La Figura **3-12** muestra el eje de la sonda de ultrasonido extraído mediante PCA en cada imagen endoscópica.



(a) Imagen endoscópica izquierda (b) Imagen endoscópica derecha

Figura 3-12.: Eje de la sonda de ultrasonido extraído mediante PCA

3.4.3. Errores de seguimiento y restricción de movimiento

Para probar la metodología de seguimiento de la sonda de ultrasonido se utilizó una base de datos de 2900 imágenes endoscópicas de fantasmas médicos de tumores cerebrales. Con el fin de comparar el desempeño de la metodología de seguimiento propuesta, se aplicó la metodología de seguimiento espacio–temporal y el tradicional filtro de partículas [88] a todas las imágenes disponibles, y se obtuvieron resultados numéricos tras la comparación con 100 imágenes segmentadas manualmente mediante el método de "ground–truth".

El eje de la sonda de ultrasonido se define por dos parámetros: en primer lugar, el centroide de la región segmentada, que determina el punto donde la línea del eje debe cruzar la sonda, y en segundo lugar, el ángulo del eje, que se calcula con el primer vector propio, extraído mediante PCA, de la región segmentada.

La Tabla **3-3** muestra los errores de cálculo de los ejes de la sonda US extraídos mediante el procedimiento espacio–temporal propuesto y el filtro de partículas, con respecto al método "ground–truth". Los errores computados corresponden a la *distancia euclidiana entre centroides* (EBC) medida en píxeles, y la *diferencia entre ángulos* (DBA) medida en grados. Las imágenes endoscópicas tienen una dimensión de 352 $\times$ 240 píxeles. Se presenta la media (μ_{error}), la desviación estándar (σ_{error}), el valor mínimo (min_{error}) y el valor máximo (max_{error}) de los errores de seguimiento a través de 100 imágenes endoscópicas tomadas a 4 cuadros por segundo.

Los resultados en la Figura **3-13** indican que el seguimiento es correcto cuando se tiene completa visibilidad de la sonda de ultrasonido. Desafortunadamente, como se muestra en la Figura **3-15**, en algunos casos la sonda de ultrasonido puede salir del rango de visión de las cámaras endoscópicas, lo que genera un seguimiento erróneo y es la razón de los altos

errores reportados en la Tabla **3-3** para el sistema sin restricción de movimiento. Con el fin de reducir estos errores, se incluye una restricción de movimiento para el eje de la sonda. Esta restricción consiste en definir las variaciones máximas de ángulo y desplazamiento permitidas para la sonda entre imágenes consecutivas de la secuencia de video. Para esto, se define el vector de estado que se muestra en la Ecuación 3-10, que codifica la pose del eje de la sonda de ultrasonido como la variación en posición y rotación en cualquier instante de tiempo t. En esta ecuación, $d_x(t)$, $d_y(t)$ y $d_\theta(t)$ son las primeras derivadas de la pose de la sonda con respecto a x, y y θ, respectivamente.

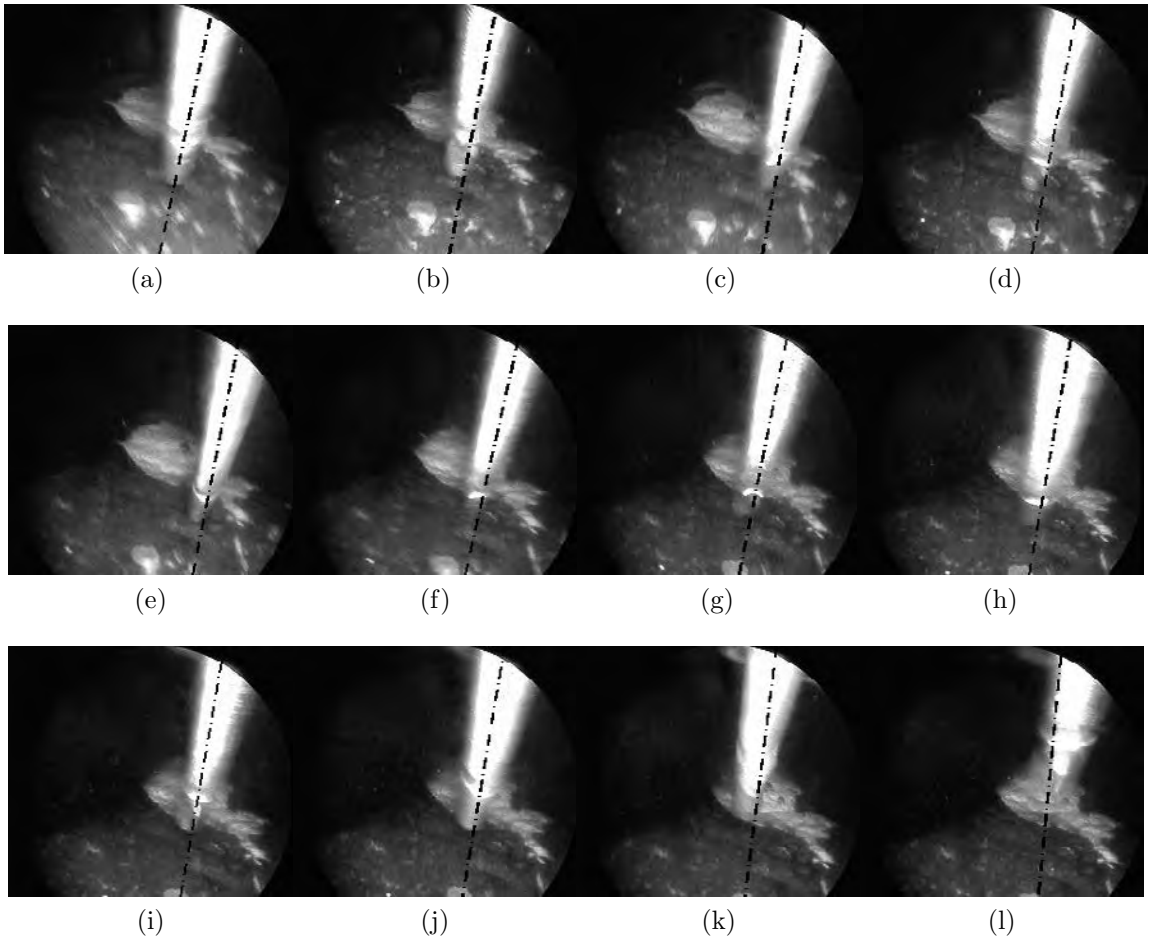


Figura 3-13.: Resultados del seguimiento con completa visibilidad de la sonda

El primer paso de la restricción de movimiento consiste en estimar el eje de la sonda de ultrasonido utilizando la metodología propuesta en las subsecciones anteriores. Esta estimación se acepta si dos condiciones se cumplen: a) la distancia euclidiana entre los centroides (EBC) de la sonda en el cuadro actual y en el cuadro anterior no debe exceder el umbral u_d; b) la diferencia entre ángulos (DBA) del eje de la sonda en el cuadro actual y en el cuadro anterior no debe exceder el umbral u_θ. Los umbrales u_d y u_θ se ajustaron heurísticamente

en 35 píxeles y 30°, respectivamente. En la Figura **3-14** se muestra un diagrama de flujo explicando la restricción de movimiento aplicada.

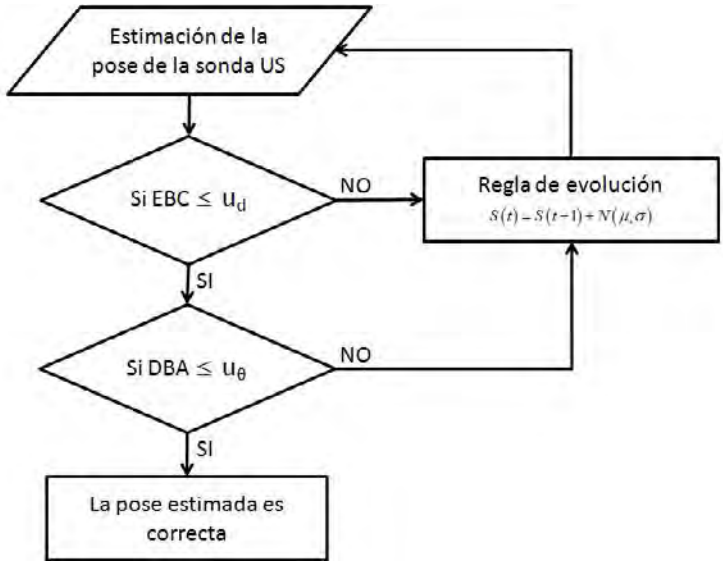


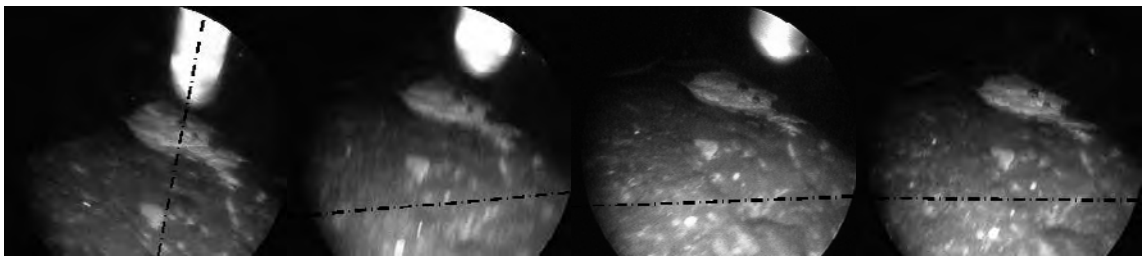
Figura 3-14.: Diagrama de flujo de la restricción de movimiento aplicada

Si estas condiciones no se cumplen, es probable que se haya cometido un error en el seguimiento de la sonda de ultrasonido debido que ésta está saliendo del rango de visión de las cámaras endoscópicas. Dadas las características erráticas y aleatorias del movimiento de la sonda, la variación del vector de estado se define por la regla de evolución de la Ecuación 3-10, donde $N(\mu_n, \sigma_n)$ es ruido blanco gaussiano. Después de analizar los movimientos de la sonda US durante las secuencias endoscópicas, los parámetros de las funciones gaussianas se fijaron en $\mu_n = 0$ y $\sigma_n = 5$.

Esta regla de evolución asegura que la variación de la pose del eje de la sonda US no sea abrupta entre una imagen y la siguiente, eliminando los errores de seguimiento causados por la pérdida de visibilidad de la sonda en la imágenes endoscópicas, como se muestra en la Figura **3-16**.

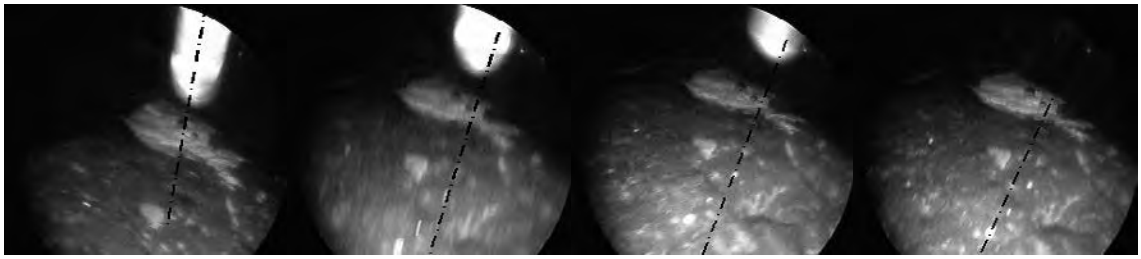
En la Tabla **3-3** se observa una reducción de los errores de seguimiento si se compara la metodología espacio–temporal de seguimiento con y sin restricción de movimiento. Comparado con el filtro de partículas, la metodología propuesta obtuvo errores de seguimiento más bajos tanto para el ángulo del eje como para la posición del sensor. Una de las justificaciones es que el filtro de partículas implementado [88] utiliza una aproximación elipsoidal de la región de interés en lugar de utilizar la forma exacta de la sonda, como lo hace la metodología de seguimiento espacio–temporal propuesta.

$$\begin{aligned} S(t) &= [d_x(t),\ d_y(t),\ d_\theta(t)]^T \\ S(t) &= S(t-1) + N(\mu, \sigma) \end{aligned} \tag{3-10}$$



(a) (b) (c) (d)

Figura 3-15.: Resultados del seguimiento de la sonda sin restricción de movimiento



(a) (b) (c) (d)

Figura 3-16.: Resultados del seguimiento de la sonda con restricción de movimiento

3.5. Procesamiento de imágenes de ultrasonido

La adquisición de ultrasonido durante la intervención se realiza mediante un dispositivo de ultrasonido Aloka SSD-5000 con un micro–transductor radial de 10 MHz conectado a la estación de neuronavegación. En la Figura **3-17** se muestra el microtransductor de ultrasonido, el cual gira sobre su propio eje, gracias a su controlador, para lograr un ángulo de escaneo de 360°.

El objetivo en esta etapa es segmentar estructuras cerebrales de interés, como tumores, a partir de imágenes de ultrasonido. Las imágenes entregadas por el sistema de adquisición ALOKA están contaminadas por ruido "speckle" y contienen pequeños agujeros debido al muestreo que realiza el equipo [83, 2]. Para eliminar estos problemas, se aplica el operador de cierre morfológico sobre la imagen I_{us} con el elemento estructurante M, como se muestra en la Ecuación 3-11, donde $\oplus$ y $\ominus$ representan la dilatación y la erosión morfológica, respectivamente. El elemento estructurante M es una máscara en forma de disco circular de 32 píxeles de radio. Después de estas operaciones, se aplica un umbral global a la imagen utilizando el método de umbralización automática de Otsu [90] con el fin de segmentar las regiones de interés.

Tabla 3-3.: Errores de seguimiento con respecto al método manual ("ground-truth")

	μ_{error}	σ_{error}	min_{error}	max_{error}
	Seguimiento espacio–temporal			
EBC (píxeles)	28.1	27.4	0.6	108.6
DBA (grados)	11.7	23.8	0.1	88.8
	Seguimiento espacio–temporal con restricción de movimiento			
EBC (píxeles)	18.8	13.5	0.6	75.8
DBA (grados)	3.9	3.5	0.1	13.4
	Filtro de partículas de color			
EBC (píxeles)	49.6	29.5	3.2	109.8
DBA (grados)	14.1	10.3	0.0	31.0

EBC: distancia euclidiana entre centroides
DBA: diferencia entre ángulos

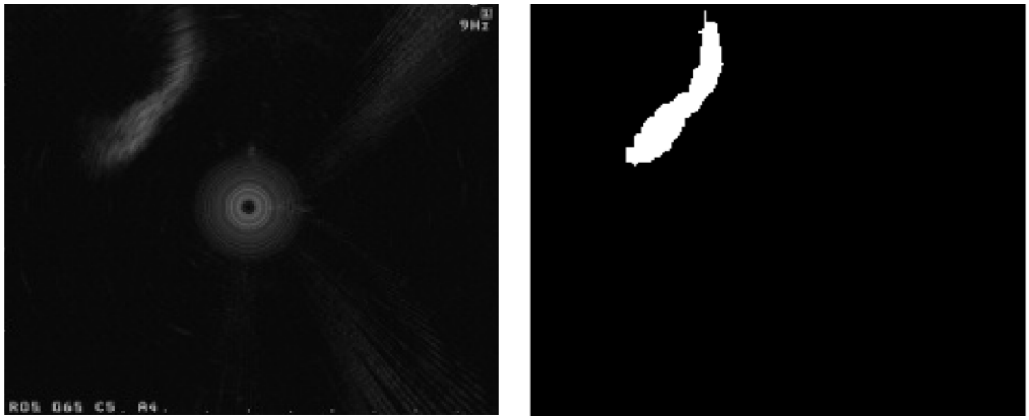
(a) Controlador (b) Microtransductor radial

Figura 3-17.: Microtransductor de ultrasonido y su controlador

$$I \bullet M = (I_{us} \oplus M) \ominus M \tag{3-11}$$

La metodología para el procesamiento de imágenes de ultrasonido fue tomada del trabajo de Machucho y otros [76], que consiste en: a) Excluir la parte central de la imagen ya que sólo contiene ruido. b) Aplicar un umbral automático a los niveles de gris de la imagen, usando el método de Otsu, para seleccionar sólo las regiones más brillantes. c) Aplicar un cierre morfológico para llenar agujeros y eliminar ruido "speckle". d) Utilizar el código de cadena para encontrar y eliminar las regiones con área más pequeña. e) Seleccionar la región de interés (ROI). f) Finalmente, aplicar el operador lógico "and" entre la región de interés y la imagen original. La imagen resultante es el tumor segmentado. La Figura **3-18**a muestra una imagen de ultrasonido típica adquirida por el equipo ALOKA SSD-5000 con un micro–transductor radial tipo B a 10 MHz, y la Figura **3-18**b muestra el resultado de la

segmentación utilizando el procedimiento explicado anteriormente.



(a) Imagen de ultrasonido (b) Región segmentada

Figura 3-18.: Segmentación morfológica de estructuras cerebrales

El centro de la imagen de ultrasonido corresponde a la ubicación del sensor radial US y es el origen del sistema de coordenadas de la imagen. La parte central de la imagen debe excluirse debido al ruido causado por la rotación del micro–transductor. Utilizando la información técnica del sistema ALOKA [2] se conoce la correspondencia entre las longitudes dadas en píxeles en la imagen de ultrasonido y las longitudes en milímetros en el espacio coordenado 3D del mundo real. Cada imagen de ultrasonido de dimensión 320×240 píxeles equivale a una imagen de 60×50 milímetros en medidas del mundo real.
En el Apéndice B se presenta una metodología alternativa de procesamiento de imágenes de ultrasonido basada en realce y filtrado wavelet, la cual presenta buenos resultados en el realce de características de interés y reducción de ruido "speckle". Sin embargo, se prefirió el uso de la metodología propuesta en esta Sección debido a su menor costo computacional y buenos resultados reportados en investigaciones anteriores [77].

3.6. Resultados: adquisición de información 3D

Usando las metodologías presentadas en este capítulo, es posible obtener información 3D de la morfología de un tumor con respecto a un sistema de coordenadas arbitrario. El origen de este sistema coordenado corresponde a la ubicación del sistema de seguimiento visual POLARIS dentro de la sala quirúrgica. Sin embargo, para facilitar la visualización, se utilizará un sistema coordenado $OXYZ$ referenciado a la posición inicial de la cámara derecha del endoscopio al comenzar la secuencia de captura ENS. Con la representación en coordenadas homogéneas, la transformación desde un sistema de coordenadas hasta otro sólo requiere la aplicación de una matriz de rotación ($R_{3\times3}$) y un vector de traslación ($T_{3\times1}$), acopladas en una matriz $TH_{4\times4}$ como se muestra en la Ecuación 3-12. Gracias a esta representación, un

sistema de coordenadas se puede transformar en otro de una manera sencilla con una simple multiplicación de matrices [34].

$$TH_{4\times4} = \begin{bmatrix} \text{Matriz de Rotacion} & \text{Vector de Traslacion} \\ \text{Transf. de Perspectiva} & \text{Escalado} \end{bmatrix} = \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{3-12}$$

El Algoritmo 1 muestra la metodología propuesta para obtener información 3D de tumores cerebrales a partir de endoneurosonografía. Cada paso del algoritmo se explica a continuación.

Algoritmo 1 Adquisición de información 3D a partir de ENS

[***Paso 1:***] Segmentación morfológica de las regiones de interés en las imágenes de ultrasonido.
[***Paso 2:***] Calibración estéreo del sistema de adquisición endoscópica
[***Paso 3:***] Calibración "hand–eye" de las cámaras endoscópicas y el sistema de seguimiento visual POLARIS.
[***Paso 4:***] Seguimiento espacio–temporal de la sonda de ultrasonido a través de las imágenes endoscópicas.
[***Paso 5:***] Transformación homogénea para pasar la información adquirida del tumor desde el sistema coordenado del sistema ENS hasta el sistema coordenado del mundo real.
[***Paso 6:***] La información tridimensional del tumor cerebral adquirida mediante ENS es almacenada y visualizada.

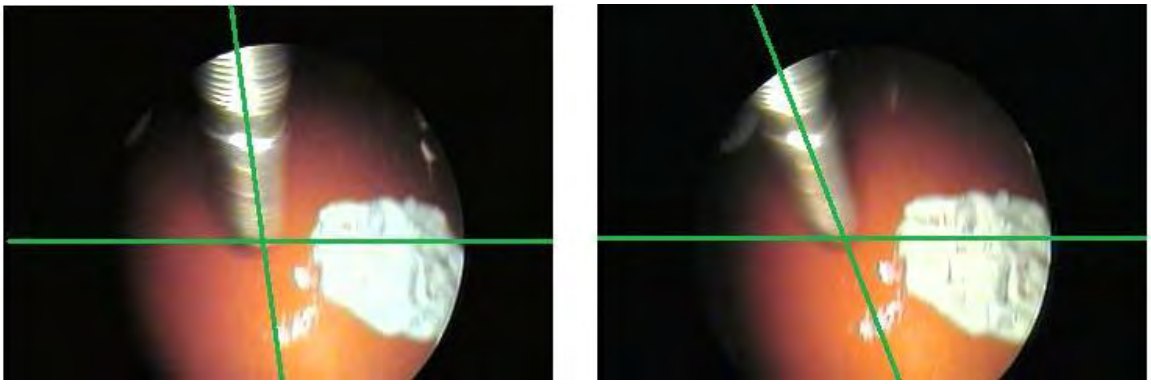
Utilizando la metodología de segmentación de imágenes de ultrasonido que se explica en la Sección 3.5, es posible obtener las coordenadas $(\hat{x}_{us}, \hat{y}_{us}, 0)^T$ de los píxeles que conforman las estructuras cerebrales en la imagen de ultrasonido. Utilizando la transformación homogénea de la Ecuación 3-13 se transforman las coordenadas de los píxeles de la imagen de ultrasonido, con origen en la esquina superior izquierda de la imagen, al sistema coordenado con origen en el centro de la imagen de ultrasonido, que es la ubicación del sensor US. Esta transformación es una traslación pura, es decir, no se requiere rotación sobre ningún eje, y las coordenadas resultantes $(x_{us}, y_{us}, z_{us})^T$ están sobre el mismo plano de la imagen. Las dimensiones de la imagen del ultrasonido, medidas en píxeles, se denotan L_x y L_y, respectivamente.

$$\begin{bmatrix} x_{us} & y_{us} & z_{us} \end{bmatrix}^T = \begin{bmatrix} \hat{x}_{us} & \hat{y}_{us} & 0 \end{bmatrix}^T - \frac{1}{2}\begin{bmatrix} l_x & l_y & 0 \end{bmatrix}^T \tag{3-13}$$

Utilizando el procedimiento de la Sección 3.4, se calcula la pose del eje de la sonda de ultrasonido para cada par de imágenes endoscópicas, como se muestra en la Figura **3-19**, donde la intersección entre las líneas denota la posición del sensor US cuyas coordenadas son $(x_{sensor}, y_{sensor}, z_{sensor})^T$. Utilizando los parámetros intrínsecos y extrínsecos de la cámara,

obtenidos mediante el procedimiento de calibración de la Sección 3.2, se hace la rectificación y reconstrucción estéreo de la escena endoscópica.
Específicamente, se desea conocer la pose tridimensional de la sonda de ultrasonido, ya que el plano en el que se encuentra la imagen de ultrasonido es perpendicular a la sonda e interseca la posición del sensor. Por lo tanto, se define el vector normal al plano $\hat{k}_w = (\hat{k}_{w,1}, \hat{k}_{w,2}, \hat{k}_{w,3})^T$ como un vector unitario con la misma dirección del eje de la sonda y con origen en la posición del sensor. Con esta información se definen los planos de cada imagen US, mediante la Ecuación 3-14, como se ilustra en la Figura **3-20** [117].

$$f_{USplane}(x, y) = -\frac{\hat{k}_{w,1}(x - x_{sensor}) + \hat{k}_{w,2}(y - y_{sensor})}{\hat{k}_{w,3}} + z_{sensor} \tag{3-14}$$



(a) Imagen izquierda (b) Imagen derecha

Figura 3-19.: Determinación del eje de la sonda y la posición del sensor de ultrasonido

Se definen dos sistemas de coordenadas: el primero, el sistema UVW correspondiente a las coordenadas de la imagen de ultrasonido centradas en la posición del sensor; el segundo, el sistema $OXYZ$ correspondiente a las coordenadas endoscópicas centradas en el punto principal de la cámara derecha. El espacio coordenado UVW se define por los vectores ortonormales $(\hat{i}_u, \hat{j}_v, \hat{k}_w)^T$, donde $\hat{i}_u, \hat{j}_v$ son vectores sobre el plano de la imagen de ultrasonido y $\hat{k}_w$ es el vector normal al plano definido por el eje de la sonda. La Ecuación 3-15 define la transformación de los puntos $(x_{us}, y_{us}, z_{us})^T$ en el espacio coordenado UVW hasta el sistema coordenado $OXYZ$ definido con referencia al endoscopio.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \hat{i}_x \cdot \hat{i}_u & \hat{i}_x \cdot \hat{j}_v & \hat{i}_x \cdot \hat{k}_w \\ \hat{j}_y \cdot \hat{i}_u & \hat{j}_y \cdot \hat{j}_v & \hat{j}_y \cdot \hat{k}_w \\ \hat{k}_z \cdot \hat{i}_u & \hat{k}_z \cdot \hat{j}_v & \hat{k}_z \cdot \hat{k}_w \end{bmatrix} \begin{bmatrix} x_{us} \\ y_{us} \\ z_{us} \end{bmatrix} + \begin{bmatrix} x_{sensor} \\ y_{sensor} \\ z_{sensor} \end{bmatrix} \tag{3-15}$$

Con la anterior transformación y conociendo las coordenadas $(x_{us}, y_{us}, z_{us})^T$ de los píxeles

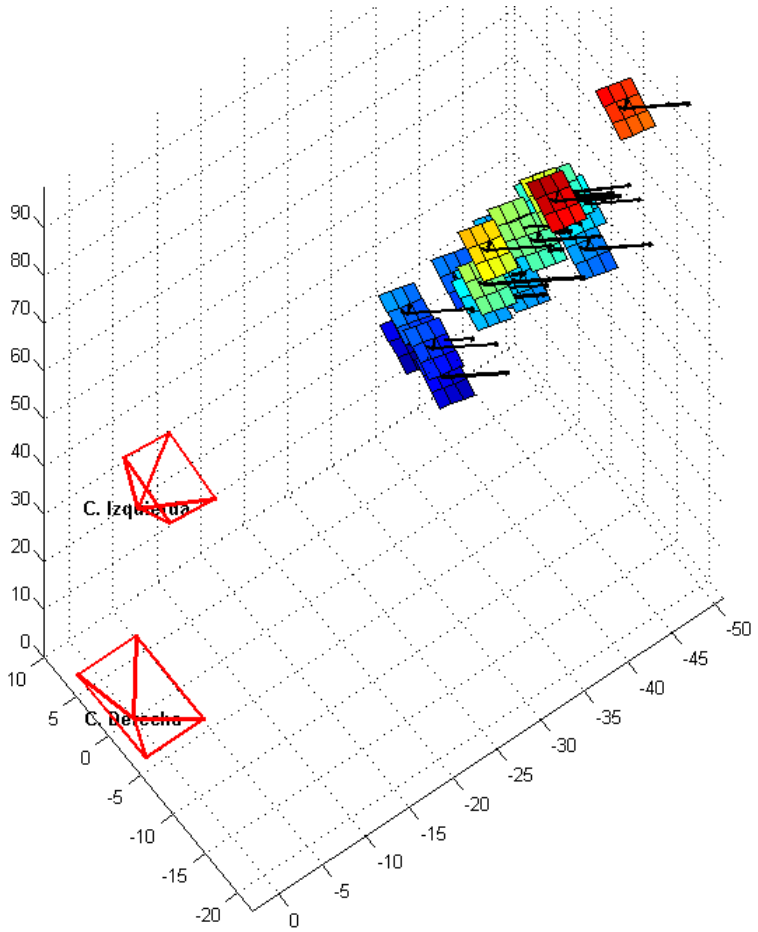


Figura 3-20.: Ubicación 3D de los planos de las imágenes de ultrasonido

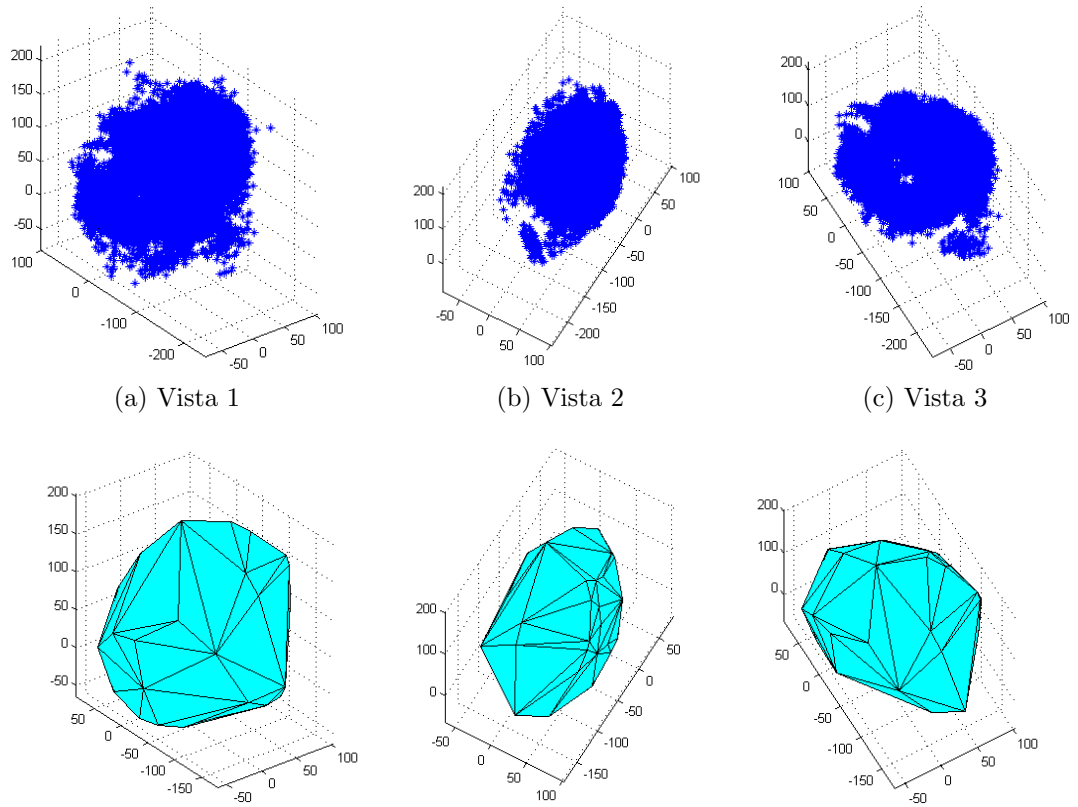
pertenecientes a las regiones de interés segmentadas en las imágenes de ultrasonido, se calcula la posición de estas estructuras cerebrales en el espacio tridimensional $OXYZ$ a través de la transformación de la Ecuación 3-15.

La Figura **3-20** muestra una secuencia endoneurosonográfica de 20 cuadros en la que se grafica cada una de las poses del eje de la sonda US y sus correspondientes planos perpendiculares sobre los que se encuentra cada imagen de ultrasonido. Esta información se grafica con respecto a la posición de las cámaras endoscópicas con origen en la cámara derecha.

La Figura **3-21** muestra la información 3D (nube de puntos y su envolvente convexa) de un tumor cerebral fantasma adquirido de una secuencia ENS de 790 cuadros tomada a 4 cuadros por segundo. Como se explicó anteriormente, cada cuadro ENS consta de dos imágenes endoscópicas y una imagen de ultrasonido, por lo que esta representación implica el procesamiento de 2370 imágenes. Las Figuras **3-22** y **3-23** muestran otras representaciones 3D de tumores cerebrales fantasma adquiridos mediante secuencias ENS de 164 y 280 cuadros, respectivamente.

Antes de iniciar cualquier preprocesamiento de datos, la secuencia endoneurosonográfica de

la Figura **3-22** proporciona información de aproximadamente 10^6 puntos tridimensionales de las regiones segmentadas que pertenecen a tumores cerebrales. Nótese que el tumor de la Figura **3-22** tiene un lado plano, esto se debe a que con la técnica de ultrasonido empleada no se puede reconocer bien la unión órgano–tumor, por lo que se ha empleado un plano para su representación.



(a) Vista 1 (b) Vista 2 (c) Vista 3

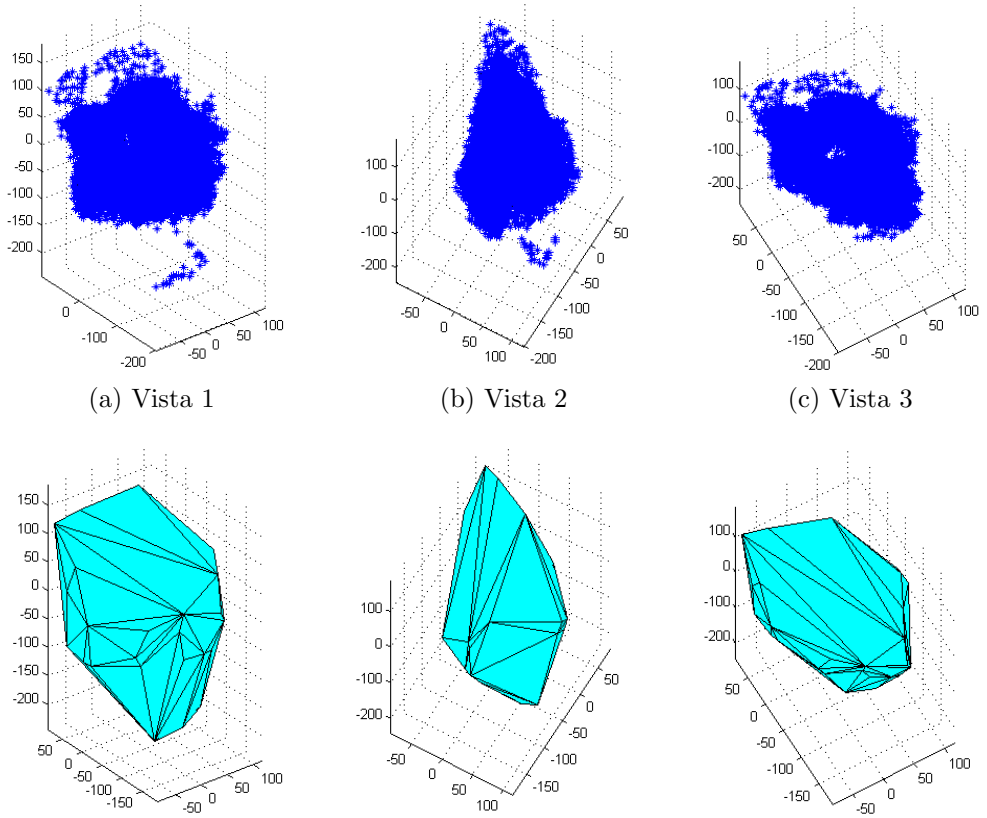
(d) Envolvente convexa 1 (e) Envolvente convexa 2 (f) Envolvente convexa 3

Figura 3-21.: Información 3D de un tumor fantasma a partir de 790 cuadros ENS

3.7. Resumen y conclusiones

Se ha presentado una solución sencilla y eficaz al problema de adquisición de información tridimensional de estructuras cerebrales a partir de secuencias de imágenes endoneurosonográficas. El método es simple, eficiente, y razonablemente robusto a la oclusión, a los desplazamientos de la sonda y a la salida de ésta del rango de visión de las cámaras endoscópicas.

El sistema endoscópico se calibra con el procedimiento propuesto Zhang [123], en el que sólo se requieren vistas de un patrón plano de calibración en diferentes orientaciones. En primer lugar, cada cámara se calibra individualmente, y luego se computan los parámetros



(a) Vista 1 (b) Vista 2 (c) Vista 3

(d) Envolvente convexa 1 (e) Envolvente convexa 2 (f) Envolvente convexa 3

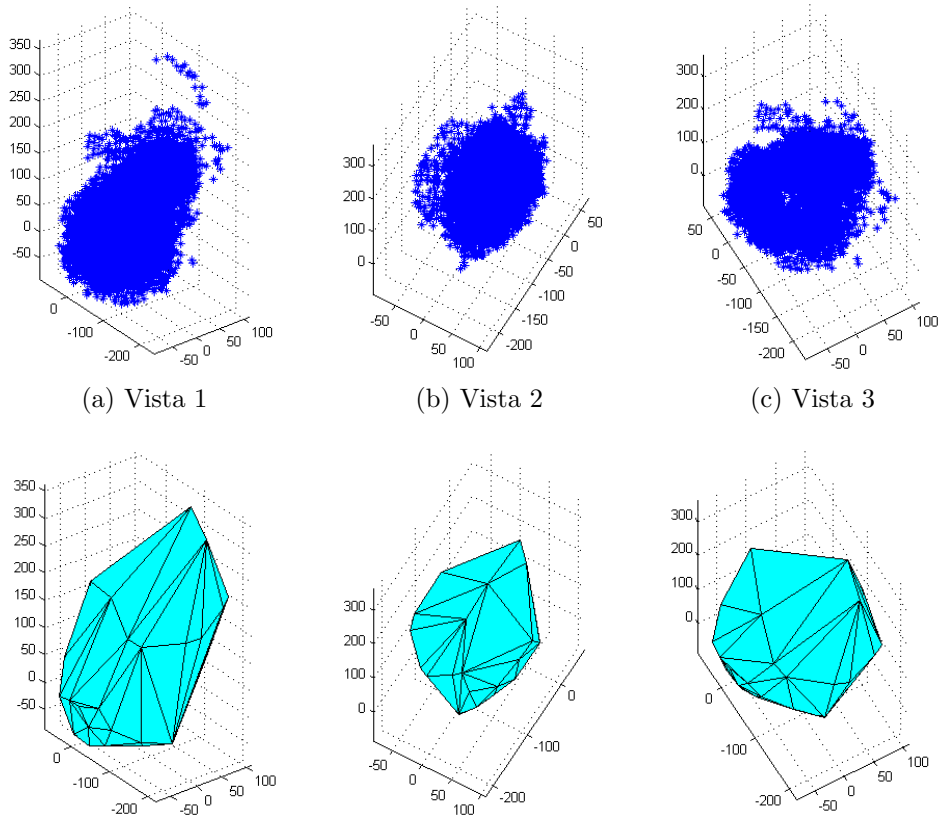
Figura 3-22.: Información 3D de un tumor fantasma a partir de 164 cuadros ENS

intrínsecos y extrínsecos del sistema estéreo. Con los parámetros resultantes se realiza la triangulación estéreo que permite determinar la pose 3D del eje de la sonda de ultrasonido a través de la secuencia de imágenes.

Con los parámetros de la calibración estéreo y la información entregada por el sistema de seguimiento visual POLARIS es posible computar la calibración "hand–eye" que permite conocer la transformación geométrica desde el sistema coordenado del patrón de calibración hasta el sistema coordenado del sistema de seguimiento. De esta manera, la información tridimensional adquirida puede referenciarse respecto a un punto arbitrario en el mundo real.

La segmentación de la sonda de ultrasonido se realiza empleando información de luminancia y flujo óptico. La elección de la luminancia como característica se debe a que la sonda de ultrasonido está construida de un material metálico que la hace más brillante que el resto de los objetos de la escena; mientras que la utilización de flujo óptico se debe a que es una clave importante para la detección de los movimientos continuos y erráticos de la sonda.

Después de implementar la metodología de seguimiento espacio-temporal de la sonda de ultrasonido, se presentaron errores debido a la oclusión y a la falta de visibilidad de la sonda



(a) Vista 1 (b) Vista 2 (c) Vista 3

(d) Envolvente convexa 1 (e) Envolvente convexa 2 (f) Envolvente convexa 3

Figura 3-23.: Información 3D de un tumor fantasma a partir de 280 cuadros ENS

cuando ésta sale del rango de las cámaras. Por esta razón, se define un vector de estados que codifica la pose del eje de la sonda en cualquier instante de tiempo a partir de sus variaciones de traslación y rotación, y se introdujo una restricción de movimiento asociada a las rotaciones y desplazamientos máximos permitidos entre imágenes consecutivas de la secuencia. La comparación de resultados que se muestra en la Tabla **3-3** demuestra que esta restricción es eficaz en la reducción de los errores medios y sus desviaciones estándar.

Con el fin de evaluar el desempeño de la metodología de seguimiento propuesta, los resultados con restricción y sin restricción de movimiento se compararon contra uno de los métodos de seguimiento más populares, el filtro de partículas [88]. En la Sección 3.4 se muestra que en el seguimiento utilizando este filtro no hay grandes errores en la determinación del ángulo de la línea que define el eje de la sonda US, sin embargo, hay errores significativos en la determinación de su centroide. Este error se debe a que el filtro de partículas implementado define la región de interés y los puntos de prueba dentro de una elipse. Por el contrario, la metodología aquí propuesta define la región exacta de la sonda, mediante una segmentación espacio–temporal, para hacer su seguimiento, por lo que se reportan menores errores. De acuerdo con los resultados presentados, el método propuesto proporciona una mejor solución

de seguimiento para este problema específico.
En los capítulos siguientes, se plantea una metodología para la representación tridimensional de estructuras cerebrales empleando redes neuronales a partir de información endoneurosonográfica. Las metodologías propuestas en este capítulo se emplean para localizar puntos en el espacio 3D que hacen parte del tumor. Esta información es la que se utiliza como entrada de las redes neuronales para ejecutar el modelado. Como se ha explicado, la endoneurosonografía brinda información en tiempo real de estructuras internas del cerebro a partir de tres videos, dos endoscópicos y uno de ultrasonido. Esta información se aprovecha para modelar la morfología de estas estructuras mediante redes neuronales artificiales, y de su correcto desempeño depende el éxito de las siguientes etapas de este trabajo.

4. Modelado 3D de objetos virtualizados usando computación neuronal

En el modelado tridimensional existen dos enfoques principales: el primero, llamado *Realidad Virtual*, se refiere a la construcción de modelos artificiales de objetos reales o imaginarios para aplicaciones en juegos de vídeo, diseño gráfico, diseño industrial, desarrollo de prototipos, entre otros. Generalmente, estos escenarios se construyen utilizando software de diseño asistido por computador (CAD). El segundo enfoque, llamado *Realidad Virtualizada*, se refiere al desarrollo de modelos de objetos del mundo real a partir de información obtenida mediante algún sensor. Los objetos de realidad virtualizada de este trabajo se adquieren mediante uno de tres sensores: un sistema endoneurosonográfico (ENS), un sistema de visión estereoscópica o un escáner de rango.

En el contexto de la computación gráfica, el modelado 3D es el proceso de dar una representación fiel de los objetos presentes en un escenario para llevar a cabo estudios de sus propiedades [19]. Uno de los mayores retos es reducir la complejidad de manera que se tengan modelos simples con precisión adecuada [47].

Hay varios aspectos que deben tenerse en cuenta al construir un modelo 3D, entre los cuales se pueden destacar: la adquisición, el propósito del modelado, la complejidad de la implementación, el costo computacional, la conveniencia de la visualización y la facilidad de manipulación. Todos estos aspectos influyen en la elección, el diseño y el desempeño del modelo.

Este capítulo presenta una metodología de modelado tridimensional utilizando redes neuronales artificiales. Se presentan comparaciones experimentales en términos de rendimiento, tiempo computacional, épocas de entrenamiento, facilidad de visualización y complejidad de la arquitectura. Su organización es como sigue: En la Sección 4.1 se describe el uso de las redes neuronales multicapa alimentadas hacia adelante (MLFFNN) en el modelado volumétrico de objetos de realidad virtualizada, y se discuten los resultados obtenidos empleando los algoritmos de entrenamiento Levenberg–Marquardt (LM) y Filtro de Kalman (KF). En la Sección 4.2 se detalla el uso de arquitecturas auto–organizadas, como los mapas auto–organizados (SOM) y las redes neuronales gas (NGN). El enfoque de estas arquitecturas es diferente al de las MLFFNN, ya que se trata de redes con esquemas de entrenamiento no supervisado en el cual las neuronas se ajustan asintóticamente a la superficie del objeto. En la Sección 4.3 se presentan y discuten los resultados del modelado 3D de objetos virtualizados usando computación neuronal. Se exponen comparaciones experimentales entre los tres métodos utilizados

y se plantea su aplicación al modelado 3D en línea de tumores cerebrales. Finalmente, la Sección 4.4 está dedicada a resumir y concluir el capítulo.

4.1. Modelado 3D empleando redes neuronales multicapa alimentadas hacia adelante (MLFFNN)

Una MLFFNN puede ser utilizada para modelar volumétricamente un objeto de realidad virtualizada a partir de una nube de puntos 3D. La red neuronal se entrena a manera de clasificador con el fin de que aprenda la morfología de un objeto. La entrada χ de la red está compuesta por las coordenadas (x, y, z) de cada punto de la nube y las combinaciones no–lineales que se establecen en la Ecuación 4-1. Estas entradas brindan robustez a la red en el sentido en que una representación MLFFNN de un objeto con superficies planas debería tener fuertes conexiones neuronales entre las entradas lineales (x, y, z) y la siguiente capa de la red; mientras que una representación MLFFNN de un objeto con superficies curvas debería tener fuertes conexiones neuronales entre las entradas no–lineales $(x^2, y^2, z^2, xy, yz, xz)$ y la siguiente capa de la red.

$$\chi = \left[\ x^2,\ \ y^2,\ \ z^2,\ \ x,\ \ y,\ \ z,\ \ xy,\ \ yz,\ \ xz\ \right]^T \tag{4-1}$$

La salida de la MLFFNN es un escalar $o \in \Re$ proporcional a la distancia euclidiana entre el punto de entrada y la superficie del objeto. La salida de la red se determina mediante la Ecuación 4-2.

$$o : \begin{cases} > 0, & \text{si } \chi \text{ esta fuera del objeto} \\ = 0, & \text{si } \chi \text{ esta sobre la superficie} \\ < 0, & \text{si } \chi \text{ esta dentro del objeto} \end{cases} \tag{4-2}$$

La representación tridimensional mediante MLFFNN es simple, compacta y precisa, siempre y cuando se esté dispuesto a pagar el costo computacional requerido en la etapa de entrenamiento. En teoría, tiene resolución infinita, ahorra espacio de almacenamiento en disco y ofrece una representación continua del objeto a modelar. Con respecto a otros métodos, sus ventajas son: resuelve los problemas de almacenamiento y largos tiempos de representación de los modelos basados en polígonos y vóxeles, resuelve la incapacidad de las superficies implícitas para especificar puntos concretos sobre la superficie del objeto, permite representar detalles finos si se usa un conjunto de datos de entrenamiento adecuado, permiten operaciones geométricas, transformaciones de apariencia y detección de colisiones [22].
La Figura **4-1** presenta el esquema de modelado tridimensional utilizando Redes Neuronales Artificiales. La entrada de la red es la nube de puntos 3D de un objeto virtualizado, mientras

que la salida es la representación tridimensional generada por la red. Como regla general, el número de neuronas en la arquitectura es menor que el número de puntos en el conjunto de datos de entrada.

Figura 4-1.: Esquema de modelado 3D basado en MLFFNN

4.1.1. Arquitectura MLFFNN

Uno de los principales desafíos en el uso de redes neuronales es el diseño de su arquitectura. En este trabajo, se ha utilizado una MLFFNN con nueve neuronas en la capa de entrada, una neurona en la capa de salida, y se ha explorado el uso de una, dos y tres capas ocultas variando el número de neuronas entre 5 y 10 neuronas por capa. Se utilizan tangentes hiperbólicas para las funciones de activación en las capas ocultas y funciones lineales para la activación de la capa de salida. La Ecuación 4-3 presenta la función de salida de una MLFFNN de dos capas ocultas, donde χ es la entrada de la red neuronal, w_i y b_i son los pesos sinápticos y el umbral de polarización de las neuronas en la capa i, respectivamente.

$$o = w_3^T \left[\tanh\left(w_2^T \left[\tanh\left(w_1^T \chi + b_1\right)\right] + b_2\right)\right] + b_3 \tag{4-3}$$

En la Tabla **4-1** se presenta una comparación del Error Cuadrático Medio (ECM) entre 6 arquitecturas MLFFNN empleadas en el modelado de 11 objetos de realidad virtualizada. Las pruebas fueron realizadas con los mismos conjuntos de entrenamiento, validación y prueba, e inicialización aleatoria de los pesos sinápticos. Con base en estos resultados se seleccionó la arquitectura de 3 capas ocultas y 10 neuronas por capa por ser la que presenta los menores errores.

4.1.2. Entrenamiento MLFFNN

La red neuronal requiere una etapa de entrenamiento fuera de línea, a través de la cual se genera una representación volumétrica de los objetos 3D a partir de la nube de puntos de entrada.

Tabla 4-1.: Comparación del error cuadrático medio de las arquitecturas MLFFNN

Capas Ocultas	**1**		**2**		**3**	
Neuronas por capa	**5**	**10**	**5**	**10**	**5**	**10**
Alca	8.3472	6.6954	9.0402	5.1753	4.7823	3.8870
Alfil	4.1466	1.4493	1.4073	1.3213	1.6673	1.2532
Banana	3.9288	4.3596	3.4376	3.3088	4.1337	3.2683
Botella	1.7846	1.0330	1.4725	1.6540	1.3482	1.2847
Campana	4.8519	3.8452	5.1698	3.4382	4.7736	2.6966
Caracol	5.2743	3.4974	3.9485	2.1631	3.2531	1.6949
Cascabel	0.0613	0.0289	0.0379	0.0173	0.0218	0.0197
Felino	3.8013	4.3518	3.1802	1.8727	3.2001	1.8361
Cabeza	0.2983	0.1970	0.1587	0.1229	0.1876	0.1188
Naranja	0.1188	0.1269	0.2051	0.0574	0.1953	0.0769
Rostro	7.2434	3.5949	4.5881	1.5788	1.6824	1.3209

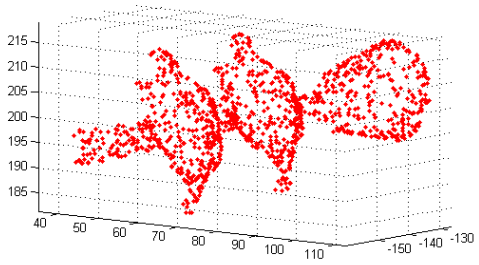
Los datos de entrenamiento empleados corresponden a nubes de puntos 3D de tumores cerebrales fantasmas, objetos triviales y piezas arqueológicas, adquiridas mediante un equipo ENS, un sistema de visión estereoscópica y un escáner de rango, respectivamente.

Para evitar un entrenamiento inadecuado que conduzca a la solución trivial, se deben agregar puntos sintéticos adicionales al conjunto de entrenamiento, tal como se muestra en la Figura **4-2**. Estos puntos deben estar ubicados al interior y al exterior del objeto para conformar un conjunto de entrenamiento suficientemente representativo. Con estos puntos adicionales, la red neuronal alcanza una mejor generalización.

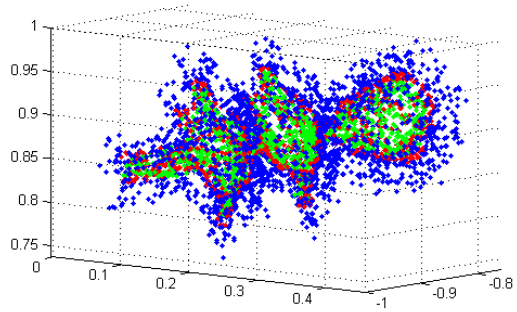
Este modelo neuronal utiliza un esquema de entrenamiento supervisado, por lo que se requiere que los datos de entrenamiento estén etiquetados. Cada etiqueta es un escalar correspondiente a la distancia euclidiana desde cada punto de entrenamiento hasta la superficie del objeto. En ese sentido, los puntos sobre la superficie del objeto tienen etiqueta cero, los puntos exteriores tienen etiqueta positiva y los puntos interiores tienen etiqueta negativa. Todos los datos de entrenamiento se normalizaron en el rango [-1, 1] y se utiliza la metodología de *validación y prueba* para evaluar el desempeño de la red [28]. El 75 % de los datos de entrada fue empleado para el entrenamiento y el 25 % para la validación. El criterio de desempeño es el Error Cuadrático Medio (ECM). Para la determinación de los pesos de la red neuronal, se utilizaron dos algoritmos de entrenamiento: Levenberg–Marquardt (LM) [45] y Filtro de Kalman (KF) [50].

4.1.2.1. Entrenamiento mediante Levenberg–Marquardt (LM)

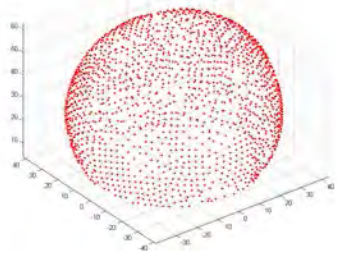
Es un algoritmo de optimización no lineal por mínimos cuadrados incorporado al algoritmo de retropropagación con el fin de resolver el problema de entrenamiento de redes neuronales. El algoritmo Levenberg–Marquardt es más eficiente que otros métodos de entrenamiento como



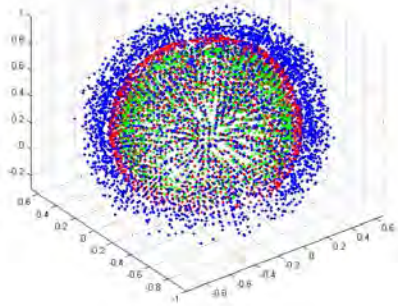
(a) Alfil: Nube de puntos original



(b) Puntos Extra. En azul: puntos exteriores. En rojo: puntos superficiales. En verde: puntos interiores.



(c) Naranja: Nube de puntos original



(d) Puntos Extra. En azul: puntos exteriores. En rojo: puntos superficiales. En verde: puntos interiores.

Figura 4-2.: Generación de puntos extra a partir de una nube puntos

el gradiente conjugado, la retropropagación de paso variable y métodos de Newton [45]. Si bien es un algoritmo acelerado de gradiente descendente, también puede ser entendido como una aproximación al método de Newton [78].

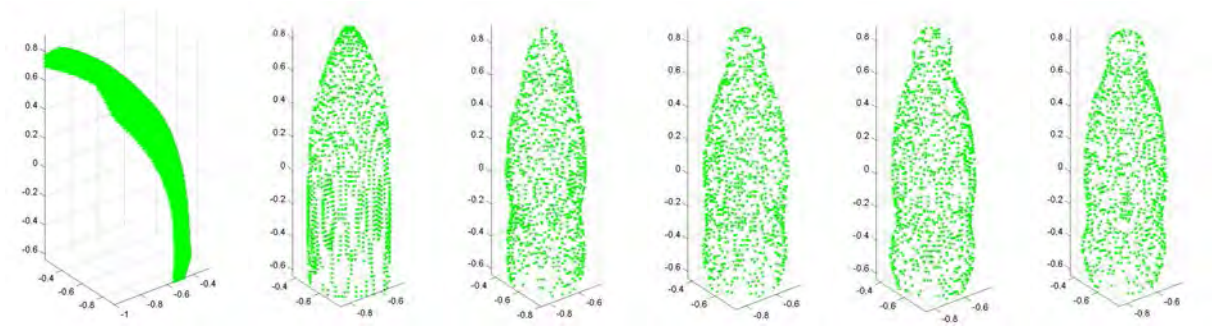
El método de Levenberg–Marquardt usa una dirección de búsqueda que es la solución de la Ecuación 4-4, donde el escalar λ_k controla la magnitud y la dirección de d_k. Cuando $\lambda_k = 0$, la dirección d_k es idéntica a la que proporciona el método de Newton. Por otro lado, cuando $\lambda_k \to \infty$, la dirección de búsqueda tiende a la dirección más rápida de convergencia, sin embargo, su magnitud tiende a cero. En este sentido, un elección adecuada de λ_k, hace que el método de Levenberg–Marquardt brinde una dirección de búsqueda intermedia entre el método de Gauss–Newton y la dirección de búsqueda ideal [68].

$$\left[J\left(x_k\right)^T J\left(x_k\right) + \lambda_k I\right] d_k = -J\left(x_k\right)^T F\left(x_k\right) \tag{4-4}$$

4.1.2.2. Entrenamiento por Filtro de Kalman (KF)

El uso del Filtro de Kalman como algoritmo de entrenamiento para redes neuronales se basa en el hecho de que los algoritmos de gradiente descendente, mínimos cuadrados recursivos y retropropagación, son casos específicos del Filtro de Kalman [98]. Durante el entrenamiento mediante filtro de Kalman, los pesos sinápticos son los estados a estimar y la salida de la red es la medida utilizada para la corrección. En este sentido, el problema de entrenamiento de una red neuronal puede ser visto como un problema de filtrado óptimo [103]. En la Figura **4-3** se muestra la evolución de una MLFFNN entrenada con Filtro de Kalman durante el modelado de un objeto de realidad virtualizada.

Para tener más detalles sobre el Filtro de Kalman y su aplicación al entrenamiento de redes neuronales, ver Apéndice A.



(a) Época 1 (b) Época 3 (c) Época 6 (d) Época 14 (e) Época 22 (f) Época 31

Figura 4-3.: MLFFNN: evolución del entrenamiento con filtro de Kalman

4.1.2.3. Comparación entre LM y KF

Para comparar el rendimiento de ambos métodos de entrenamiento, se realizaron pruebas de modelado utilizando la misma arquitectura y los mismos conjuntos de entrenamiento y validación, tanto para Levenberg–Marquardt como para el Filtro de Kalman. El Error Cuadrático Medio (ECM) es la medida de desempeño empleada. De acuerdo con la Figura **4-4**, la MLFFNN entrenada con Levenberg-Marquardt alcanza menor error que el Filtro de Kalman en seis de los once modelos probados. Sin embargo, los desempeños de ambos métodos son comparables, la superioridad de un método sobre otro no es clara, y parece haber una relación con la morfología del objeto que se desea modelar. La Figura **4-5** presenta la comparación entre ambos métodos de acuerdo al número de épocas de entrenamiento requeridas. Para alcanzar un rendimiento dado, el entrenamiento por filtro de Kalman requiere menos épocas de entrenamiento que Levenberg–Marquardt en ocho de los once casos presentados.

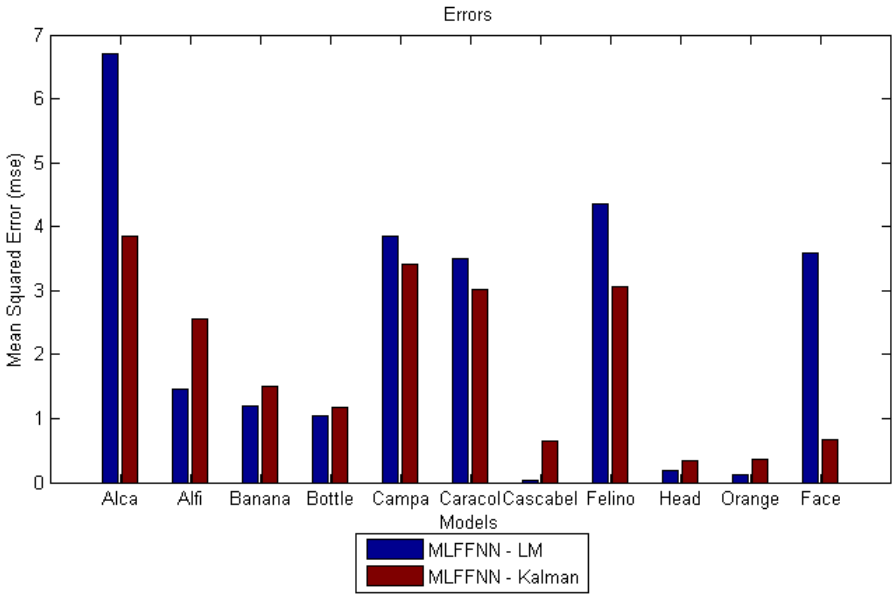


Figura 4-4.: MLFFNN: comparación del error cuadrático medio

4.1.3. Visualización MLFFNN

Para el esquema de modelado 3D usando MLFFNN, la red neuronal es un clasificador entrenado para aprender una representación volumétrica de un objeto de realidad virtualizada. La salida de la red neuronal es un escalar proporcional a la distancia euclidiana entre los patrones de entrada y la superficie del objeto.
En este sentido, es necesario definir un conjunto de datos para evaluar y visualizar el modelo generado. Se define un conjunto de 216000 puntos de prueba (x_p, y_p, z_p) en el intervalo $[-1, 1]$, que se utiliza como entrada del clasificador con el fin de conocer los puntos que corresponden a la superficie, al interior o al exterior del objeto. A continuación, se grafican únicamente los puntos que se encuentran en la superficie del objeto, es decir, aquellos puntos para los que la respuesta de la red es cero. La Figura **4-6** muestra un resultado de modelado 3D usando MLFFNN. Nótese que los modelos generados con ambos métodos de entrenamiento son similares.

4.2. Modelado 3D empleando redes neuronales auto–organizadas

Las arquitecturas auto–organizadas son red neuronales de aprendizaje no supervisado que pueden ser empleadas para el modelado 3D a partir de nubes de puntos de objetos virtualizados. Dos de las arquitecturas auto–organizadas más populares son los *Mapas Auto–Organizados (SOM)* y las *Redes Neuronales Gas (NGN)*. Ambas arquitecturas se caracterizan por estar basadas en entrenamiento competitivo [79, 33].

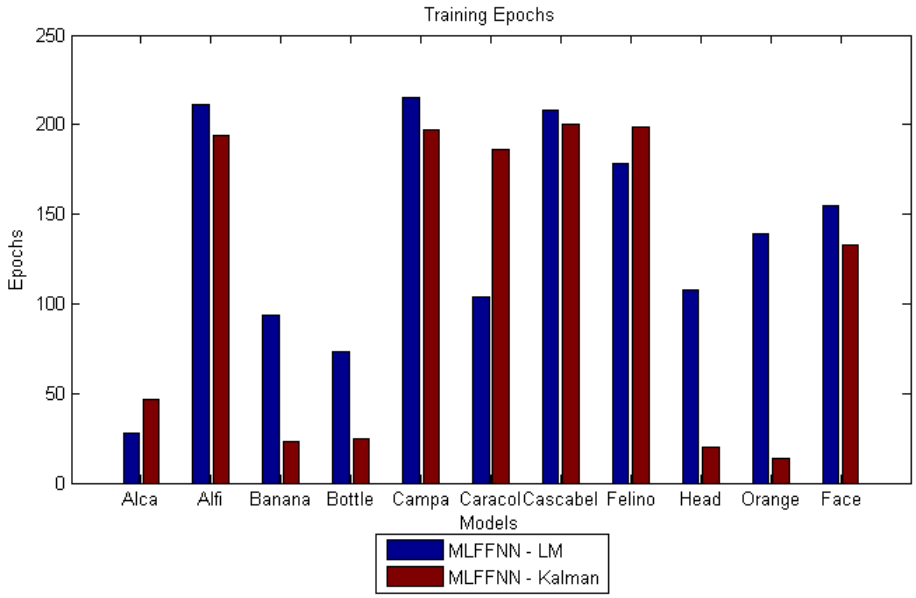
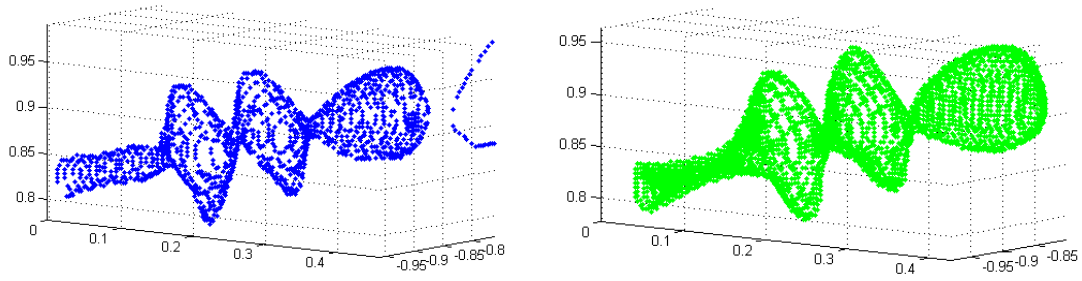


Figura 4-5.: MLFFNN: comparación del número de épocas de entrenamiento



(a) Entrenamiento: Levenberg–Marquardt (b) Entrenamiento: Filtro de Kalman

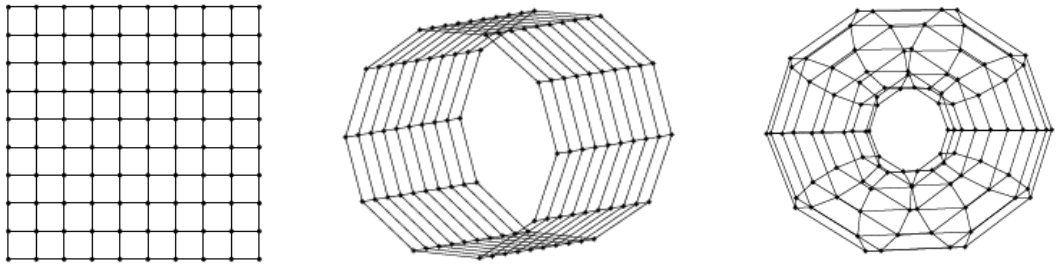
Figura 4-6.: Modelado 3D empleando MLFFNN

El espacio de entrada se agrupa mediante la asignación de neuronas a regiones específicas del espacio. El número de entradas de cada neurona es igual a la dimensión del espacio de entrada, por lo que los pesos sinápticos se interpretan como ubicaciones en este espacio. Cada región es un grupo de puntos cercanos con respecto a las neuronas asignadas. Después del entrenamiento, dos vectores de entrada que pertenezcan a la misma región deberían estar representados por una o dos neuronas cercanas dentro del espacio de representación.

Tanto los SOM como las NGN pueden utilizarse para modelar superficialmente y de manera comprimida objetos de realidad virtualizada a partir de nubes de puntos 3D. Cada vector de peso sináptico codifica las coordenadas 3D de un agrupamiento de puntos sobre la superficie del objeto. Durante el entrenamiento, el modelo se ajusta asintóticamente hacia los puntos en el espacio de entrada de acuerdo a una función de densidad de puntos, y por lo tanto toma la forma del objeto codificado en la nube de puntos de entrada.

4.2.1. Mapas Auto–Organizados (SOM)

Un Mapa Auto–Organizado (SOM - "Self Organizing Map") es un conjunto de neuronas organizadas en una malla regular con conexiones rectangulares o hexagonales [114]. La malla puede tener diferentes configuraciones dependiendo del objeto que se desea modelar, como se indica en la Figura **4-7**.



(a) Hoja (b) Cilindro (c) Toroide

Figura 4-7.: Tipos de mallas SOM. Tomado de [114]

Los datos de entrada de la red son un conjunto de coordenadas (x, y, z) de una nube de puntos 3D que representa un objeto virtualizado. Por lo tanto, cada neurona está representada por un vector de pesos tridimensional $\mathbf{m}_i \in \Re^3$.
El objetivo del algoritmo SOM es aproximar el espacio de entrada ξ por prototipos o punteros a manera de pesos sinápticos $\mathbf{m}_i$, de forma que el mapa característico ϕ dé una representación fiel de los vectores de entrada [64].

4.2.1.1. Entrenamiento SOM

Los SOM son entrenados iterativamente como se explica a continuación. En cada etapa de entrenamiento, un vector $\mathbf{x}$ se elige aleatoriamente del conjunto de datos de entrada. Se calcula la distancia entre $\mathbf{x}$ y todos los vectores de peso sináptico del SOM. La neurona cuyo vector de peso sináptico sea el más cercano al vector de entrada $\mathbf{x}$ se llama Neurona de Mejor Ajuste (BMU - "Best Match Unit") o neurona ganadora, y se denota con la letra c, según se determina por la Ecuación 4-5. En general, se puede utilizar cualquier medida de distancia, sin embargo, dada la naturaleza geométrica del problema (realidad virtualizada), se utiliza distancia euclidiana para encontrar la BMU de cada vector de entrada. Finalmente, se añade una variable de peso a_k a cada neurona, de modo que la medida de distancia se puede reescribir como se indica en la Ecuación 4-6.

$$\|\mathbf{x} - \mathbf{m}_c\| = \min_i \{\|\mathbf{x} - \mathbf{m}_i\|\} \tag{4-5}$$

$$\|\mathbf{x} - \mathbf{m}\|^2 = \sum_{k \in K} a_k \left(\mathbf{x}_k - \mathbf{m}_k\right)^2 \tag{4-6}$$

Después de determinar la neurona ganadora, los vectores de peso sináptico del SOM se actualizan de modo que la BMU se mueve asintóticamente hacia el vector $\mathbf{x}$ en el espacio de entrada. Los vecinos topológicos de la BMU, es decir, los vecinos de acuerdo a la configuración de la malla, también se actualizan de forma similar, pero en menor medida que la neurona ganadora, de modo que la morfología del SOM se ajusta asintóticamente a la morfología de la nube de puntos de entrenamiento de acuerdo a un criterio de error. La regla de actualización del vector de pesos de la neurona i se presenta en la Ecuación 4-7, donde $\alpha(t)$ es la razón de aprendizaje, que por lo general disminuye con el tiempo, y $h_{ci}(t)$ es la máscara del vecindario alrededor de la neurona ganadora c. Esta máscara es una función que decrece con el tiempo y depende de la distancia de la BMU a cada una de las neuronas sobre la malla del SOM. Además, $h_{ci}(t)$ define la región de influencia del vector de entrada en el mapa. En este trabajo se ha utilizado una función de vecindad gaussiana, como se indica en la Ecuación 4-8, donde σ_t es una función decreciente que define el radio de influencia en la vecindad y d_{ci} es la distancia entre la neurona c y la neurona i a través de las conexiones de la malla. La Figura **4-8** ilustra el proceso de adaptación SOM.

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h_{ci}(t) \left[\mathbf{x}(t) - \mathbf{m}_i(t)\right] \tag{4-7}$$

$$h_{ci}(t) = \exp\left(-\frac{d_{ci}^2}{2\sigma_t^2}\right) \tag{4-8}$$

Antes de iniciar el entrenamiento, los datos de entrada se normalizan y los pesos sinápticos se inicializan linealmente a lo largo de las dos componentes principales de la distribución de los datos, extraídos mediante Análisis de Componentes Principales (PCA), lo que garantiza una convergencia más rápida. La relación de vecindad es definida por una malla hexagonal.

Las Figuras **4-9**b y **4-9**e muestran dos ejemplos de modelado tridimensional usando Mapas Auto–Organizados. Los objetos virtualizados corresponden a piezas arqueológicas [31]. Debido a las restricciones de la malla hexagonal, no es posible eliminar las conexiones entre cada neurona y sus vecinos, por lo que los agujeros de los objetos no se pueden modelar correctamente.

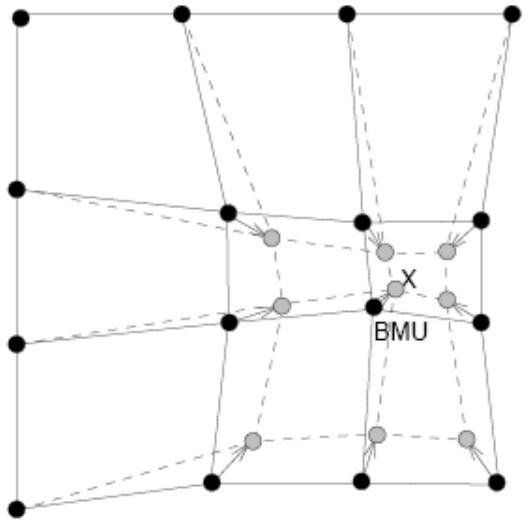


Figura 4-8.: Proceso de adaptación SOM. Tomado de [114]

4.2.2. Red Neuronal Gas (NGN)

Los errores del modelado SOM, causados por las restricciones de la malla, son resueltos por el modelado con Redes Neuronales Gas (NGN - "Neural Gas Networks"). Una NGN está compuesta por neuronas, de manera similar que un SOM, que se mueven en el espacio de entrada durante el entrenamiento hasta ajustarse a una nube de puntos de entrada. La posición de la neurona ganadora y sus vecinas se actualiza durante cada etapa de entrenamiento. La diferencia con la arquitectura SOM, es que en las NGN la relación de vecindad se define calculando la distancia euclidiana a todas las neuronas de la red, y no mediante una malla topológica predefinida. Es decir, mientras en un SOM los vecinos de cada neurona se conocen de antemano dada la configuración de la red, en una NGN los vecinos de cada neurona deben calcularse en cada época de entrenamiento con respecto a todas las neuronas de la red.

Comparado con los SOM, las NGN no usan una relación de vecindario definida por una malla, sino que usa un ranking de vecindario basado en el cálculo de las neuronas más cercanas con respecto a todas las neuronas de la red. Por este motivo, el proceso de adaptación NGN es mejor que el de SOM, ya que las neuronas de la NGN pueden moverse libremente en el espacio de entrada sin ningún tipo de restricción topológica. Así, el error topográfico[1] de una NGN es cero [114].

El algoritmo NGN puede entenderse como un algoritmo de Cuantización de Vectores (VQ) en el sentido en que un conjunto de M vectores de entrada se aproxima por un conjunto de N neuronas (o vectores de referencia), cuyos pesos sinápticos ofrecen una representación

[1]El error topográfico es una medida cuantitativa de la preservación topológica del SOM después del entrenamiento, y se calcula como la proporción de todos los datos de entrada para los cuales la neurona más cercana y la segunda neurona más cercana no son adyacentes dentro de la malla de la red.

aproximada de la distribución de los vectores de entrada, cumpliéndose que $N \ll M$ [79]. Comparado con otros métodos de VQ como SOM, "k–means clustering" y agrupamiento de máxima entropía, las NGN convergen más rápidamente, alcanzan menores errores de distorsión, tienen más alto costo computacional, minimizan un funcional de costo durante el entrenamiento y pueden alcanzar mejores desempeños empleando conjuntos de entrenamiento más pequeños [79].
NGN es un algoritmo de compresión de datos basado en una regla de adaptación "soft–max" con el objetivo de minimizar una función de distorsión. Esta regla de adaptación es una extensión del algoritmo estándar "k–means clustering" y tiene en cuenta un ranking de vecindario de pesos sinápticos calculado sobre todas las neuronas de la red.
La dinámica de los pesos sinápticos durante el entrenamiento consta de dos pasos: en el primero, el proceso de adaptación está determinado por el gradiente de una función de costo dependiente de los parámetros de vecindad de la red; en el segundo paso, hay un ajuste similar al de la dinámica de las partículas brownianas moviéndose a través de un potencial determinado por la densidad de la nube de puntos de entrada [6].

4.2.2.1. Entrenamiento NGN

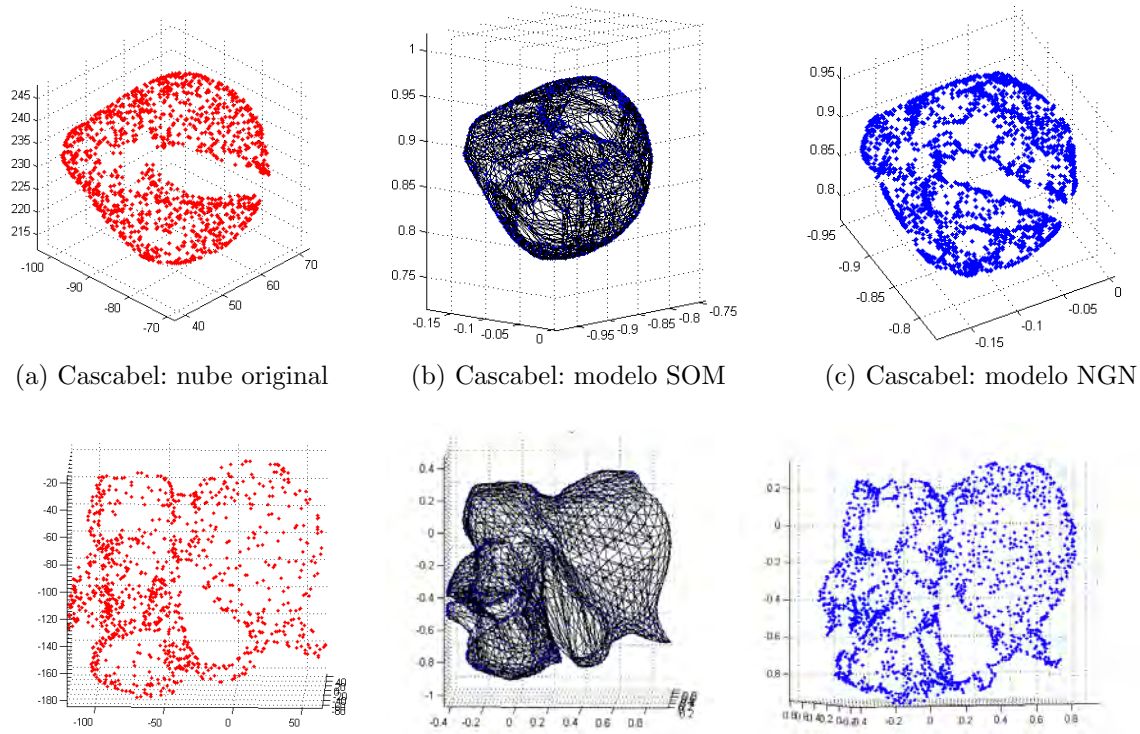
De la misma manera que los algoritmos SOM y "k–means", el algoritmo NGN utiliza una regla de adaptación "soft–max" [86]. Sin embargo, en lugar de utilizar la distancia d_{ci} definida sobre una malla de organización de neuronas, se utiliza un ranking de vecindad que tiene en cuenta todas las neuronas del espacio.
Cuando cada vector $\mathbf{x}$ se presenta, se calcula el ranking de vecindad $(\mathbf{w}_{i0}, \mathbf{w}_{i1}, ..., \mathbf{w}_{i(N-1)})$, donde $\mathbf{w}_{i0}$ y $\mathbf{w}_{i(N-1)}$ son la neurona más cercana y más lejana al vector de entrada $\mathbf{x}$, respectivamente.
La regla de adaptación de pesos sinápticos está dada por la Ecuación 4-9, donde ε es la razón de aprendizaje en el intervalo [0,1] y h_λ es la función de vecindad definida en la Ecuación 4-10, donde $k_i(x, w)$ denota la posición de la neurona i en el ranking del vecindario. Para $\lambda = 0$, este algoritmo es idéntico al algoritmo "k–means", mientras que para $\lambda \neq 0$ no sólo se actualiza la neurona ganadora, sino también las neuronas vecinas definidas en el ranking del vecindario.

$$\Delta \mathbf{w}_i = \varepsilon \cdot h_\lambda \left(k_i \left(\mathbf{x}, \mathbf{w}\right)\right) \cdot \left(\mathbf{x} - \mathbf{w}_i\right) \tag{4-9}$$

$$h_\lambda \left(k_i \left(x, w\right)\right) = \exp\left(-\frac{k_i \left(x, w\right)}{\lambda}\right) \tag{4-10}$$

Las Figuras **4-9**c y **4-9**f presenta dos ejemplos de modelado 3D usando NGN. Los objetos virtualizados correspondes a piezas arqueológicas [31]. En comparación con el modelado SOM

presentado en las Figuras **4-9**b y **4-9**e, la representación NGN sí permite el modelado de los agujeros de los objetos.



(a) Cascabel: nube original (b) Cascabel: modelo SOM (c) Cascabel: modelo NGN

(d) Alca: nube original (e) Alca: modelo SOM (f) Alca: modelo NGN

Figura 4-9.: Modelado 3D basado en Redes Neuronales Gas (NGN)

4.2.3. Comparación entre SOM y NGN

Para evaluar el desempeño de la red, se calculan los errores de cuantización como la distancia euclidiana absoluta media (EAM) desde cada punto de entrada hasta su correspondiente Neurona de Mejor Ajuste (BMU). El algoritmo SOM tiene una capacidad de adaptación inferior que la de NGN debido a las restricciones topológicas de la malla. Sin embargo, las NGN tiene el más alto costo computacional, ya que en cada época de entrenamiento se debe calcular el ranking de vecindario sobre todas las neuronas de la red $(\mathbf{w}_{i0}, \mathbf{w}_{i1}, ..., \mathbf{w}_{i(N-1)})$, lo que hace que el costo computacional aumente como $N \times log(N)$, donde N es el número de neuronas de la red [79]. En la siguiente sección se presentan los resultados experimentales que ilustran estas diferencias .

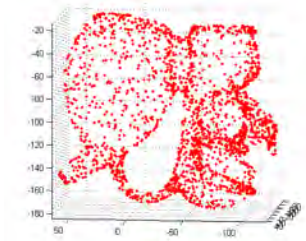
4.3. Resultados: modelado 3D empleando computación neuronal

Para probar el rendimiento de las tres arquitecturas neuronales en la tarea de modelado 3D, se utilizaron diferentes objetos del mundo real virtualizados mediante diferentes sistemas de adquisición: a) modelos fantasma de tumores cerebrales cuya información fue adquirida mediante endoneurosonografía (Ver Capítulo 3) [77, 101]. b) objetos triviales, como frutas y botellas, virtualizados mediante un sistema de visión estéreo StarCam FW–3R. c) piezas arqueológicas virtualizadas mediante un escáner de rango 3D Konica Minolta Vivid–9i [31, 65]. La Figura **4-10** presenta los objetos de realidad virtualizada utilizados en estos experimentos.
La Tabla **4-2** describe los parámetros de diseño de las tres arquitecturas neuronales empleadas. El modelo MLFFNN requiere el menor espacio de almacenamiento en disco debido a que su arquitectura sólo requiere 30 neuronas (2.3 Kbytes), mientras que los SOM y las NGN requieren 2000 neuronas (45 Kbytes) cada una. Respecto al número de puntos de entrenamiento, el modelado MLFFNN requiere 4 veces más datos que SOM y NGN debido a los puntos extra añadidos al interior y al exterior del objeto durante el entrenamiento. Gracias a estos puntos extras, el modelado MLFFNN ofrece una representación tanto volumétrica como superficial dependiendo de los datos que se deseen graficar, mientras que el modelado SOM y NGN es superficial.

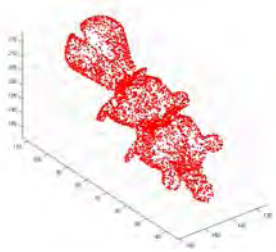
Tabla 4-2.: Comparación de los parámetros de diseño de cada arquitectura neuronal

	Arquitectura Neuronal		
	MLFFNN	**SOM**	**NGN**
Almacenamiento en disco	2.3 Kbytes	46 Kbytes	45 Kbytes
Número de neuronas	30 neuronas	2000 neuronas	
Método de entrenamiento	Levenberg–Marquardt, Filtro de Kalman	Aprendizaje competitivo	
Error	Error Cuadrático Medio (ECM), Error Absoluto Medio (EAM)	Error Absoluto Medio (EAM)	
Representación	Volumétrica Superficial	Superficial	
Puntos de Entrenamiento	$4 \times N_{train_points}$	$1 \times N_{train_points}$	

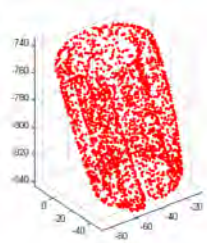
La Tabla **4-3** muestra la comparación de errores entre los modelos 3D generados mediante MLFFNN, SOM y NGN. El error se calcula como la distancia euclidiana absoluta promedio desde cada punto de entrada hasta la superficie del modelo generado. Es evidente que el modelado mediante MLFFNN tiene mayor error que los modelados basados en arquitecturas



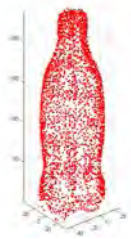
(a) Alca



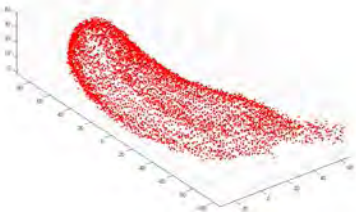
(b) Alfil



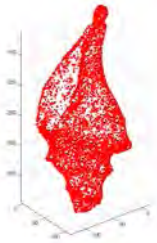
(c) Campana



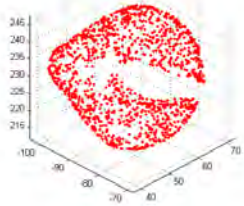
(d) Botella



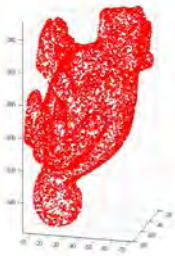
(e) Banana



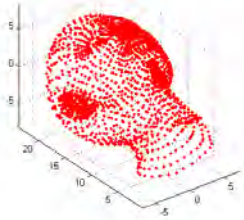
(f) Caracol



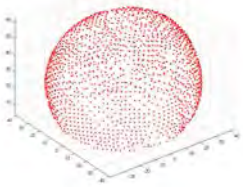
(g) Cascabel



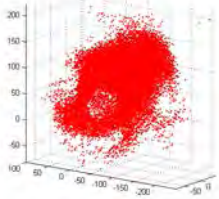
(h) Felino



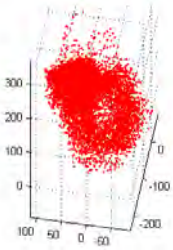
(i) Cabeza



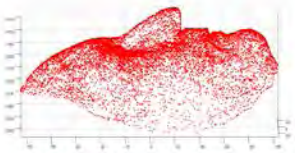
(j) Naranja



(k) Tumor 1



(l) Tumor 2



(m) Rostro

Figura 4-10.: Objetos de realidad virtualizada empleados en los experimentos

auto–organizadas. En general, para todas las arquitecturas los errores más altos se presentan para los objetos más elaborados, como los de las Figuras **4-10**a, **4-10**c, **4-10**f y **4-10**h, mientras que los objetos más simples presentan errores más bajos, como los mostrados en las Figuras **4-10**g, **4-10**i y **4-10**j.

Tabla 4-3.: Distancia desde cada punto hasta la superficie del modelo generado

	Distancia euclidiana absoluta promedio		
Modelo	**MLFFNN**	**SOM**	**NGN**
Alca	1.9715	0.0248	0.0149
Alfil	1.1195	0.0047	0.0024
Banana	1.8078	0.0145	0.0079
Botella	1.1334	0.0176	0.0079
Campana	1.6421	0.0065	0.0033
Caracol	1.3019	0.0152	0.0077
Cascabel	0.1404	0.0027	0.0019
Felino	1.3550	0.0111	0.0053
Cabeza	0.3446	0.0144	0.0082
Naranja	0.2773	0.0146	0.0101
Rostro	1.1493	0.0030	0.0018

La Figura **4-11** muestra una comparación entre el coste computacional de las diferentes arquitecturas medidas en tiempo de entrenamiento (segundos). Note que las NGN tiene el mayor tiempo de procesamiento ya que, como se explicó en las secciones anteriores, su tiempo de procesamiento aumenta como $N \times log(N)$ donde N es el número de neuronas [79]. Las pruebas de cómputo se realizaron usando MATLAB R2009a en un ordenador portátil con Windows Vista, procesador Intel Core 2 Duo T5670@1.80 GHz, 3 GB de memoria RAM, DirectX 11 y una tarjeta gráfica NVIDIA GeForce 8400M GS de 1523 MB.
La Figura **4-12** presenta el resultado del modelo 3D de dos piezas arqueológicas, las más complejas disponibles, utilizando cada una de las arquitecturas neuronales. La Figuras **4-12**b y **4-12**f exponen el resultado del modelado MLFFNN, con 3 capas ocultas y 10 neuronas en cada una, entrenada por el algoritmo de Levenberg–Marquardt. La Figuras **4-12**c y **4-12**g muestran el resultado del modelado SOM sin visualizar las conexiones de la malla. Por último, la Figuras **4-12**d y **4-12**h presentan el resultado del modelado NGN. Tanto los SOM como las NGN se entrenan competitivamente durante 15 épocas de ajuste grueso y 50 épocas de ajuste fino para una arquitectura de 2000 neuronas de representación. Estas arquitecturas auto–organizadas presentan mejor desempeño que las arquitecturas multicapa, lo que se explica por la poca capacidad de la MLFFNN de representar detalles finos.
En la Figura **4-13** se muestra la evolución de los entrenamientos grueso y fino en el modelado 3D de una pieza arqueológica mediante SOM. Durante el entrenamiento grueso, el radio del vecindario se inicializa en el valor $\sigma_t = 8$ y decrece hasta llegar al valor $\sigma_t = 3$. Durante el entrenamiento fino, el radio del vecindario se inicializa en el valor $\sigma_t = 3$ y decrece hasta

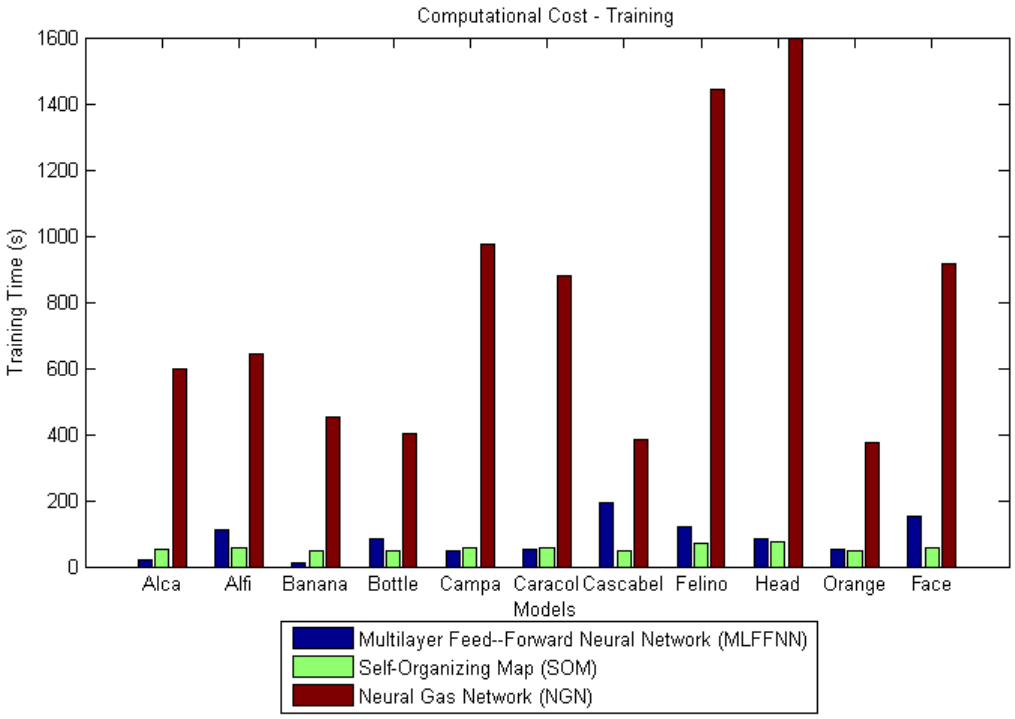
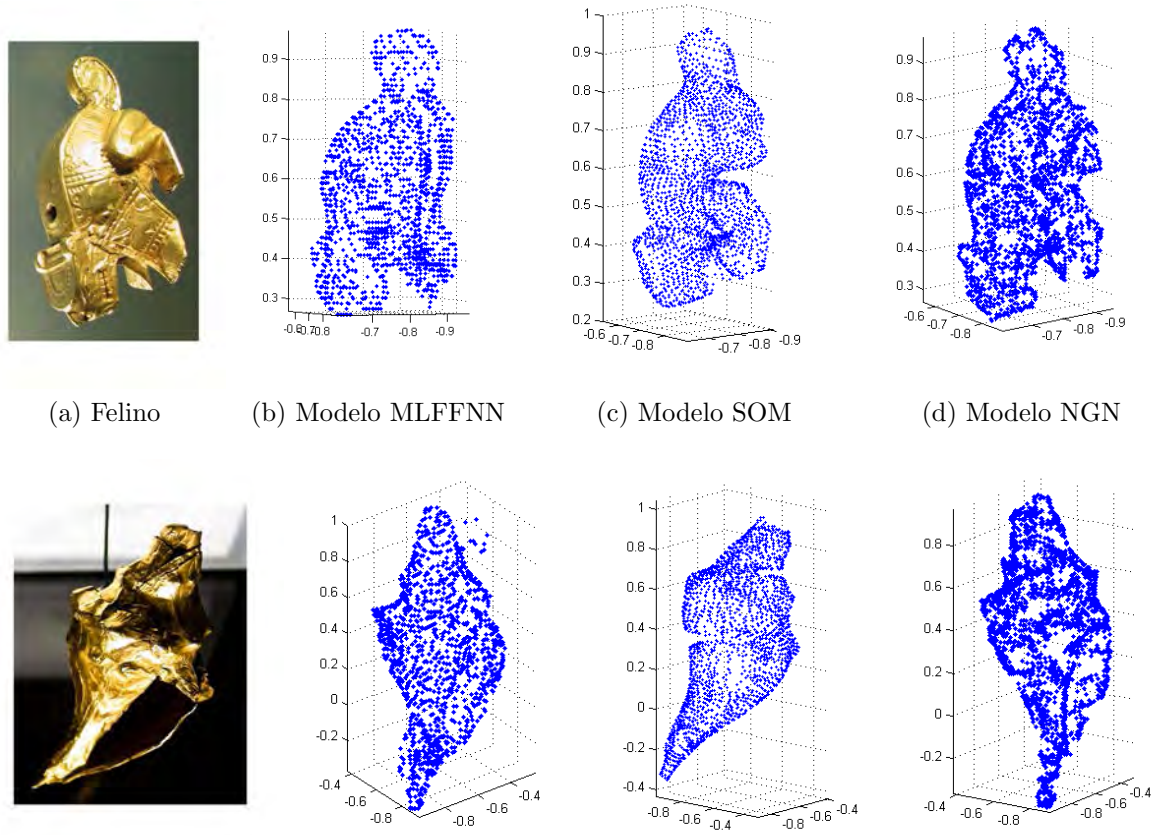


Figura 4-11.: Comparación del tiempo de entrenamiento

llegar al valor $\sigma_t = 1$, es decir, en las últimas etapas del entrenamiento fino, no se tiene vecindario y se ajusta solamente cada neurona ganadora.
La Figura **4-14** presenta una comparación entre los tiempos de visualización de cada modelo neuronal. Como se explica en las secciones anteriores, y se confirma con los resultados aquí presentados, el modelado MLFFNN es el que requiere mayor tiempo de visualización debido a que requiere un conjunto de datos de prueba como entrada de la red entrenada para poder visualizar el objeto. Por otro lado, el modelado NGN presenta el tiempo de visualización más corto debido a que sus pesos sinápticos son posiciones en el espacio de entrada que codifican directamente la morfología tridimensional del objeto. Finalmente, el modelado SOM tiene la misma ventaja que NGN de codificar directamente en sus pesos sinápticos la morfología del objeto, sin embargo, la visualización SOM es un poco más lenta que la NGN debido al tiempo de renderización de la malla de conexiones SOM.

4.4. Resumen y conclusiones

En este capítulo se describe la implementación de una metodología de modelado tridimensional basada en redes neuronales. Los patrones de entrada son nubes de puntos (x, y, z) que describen la superficie de los objetos a modelar y que fueron virtualizados por unos de los siguientes métodos de adquisición: un sistema ENS, un sistema de visión estéreo o un escáner de rango 3D.



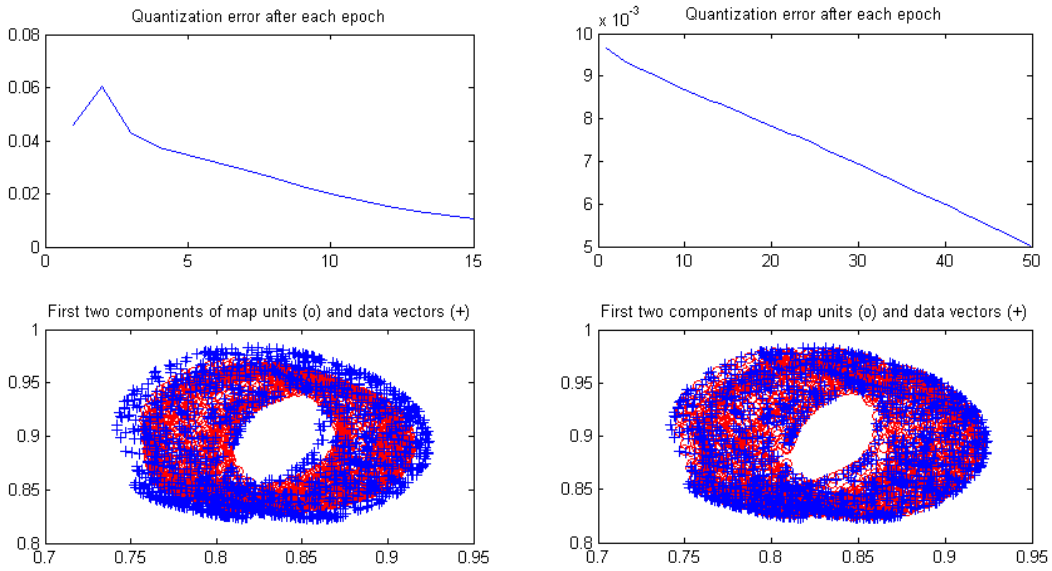
(a) Felino (b) Modelo MLFFNN (c) Modelo SOM (d) Modelo NGN

(e) Caracol (f) Modelo MLFFNN (g) Modelo SOM (h) Modelo NGN

Figura 4-12.: Modelado 3D de piezas arqueológicas empleando redes neuronales

Se explora el uso de tres tipos de arquitecturas neuronales para la tarea de modelado tridimensional: redes neuronales multicapa alimentadas hacia adelante (MLFFNN), mapas auto–organizados (SOM) y redes neuronales gas (NGN). Se presentan y discuten comparaciones experimentales entre estas arquitecturas en términos de desempeño, costo computacional, tiempo de entrenamiento, tiempo de visualización y complejidad de la arquitectura.

Los SOM y las NGN proveen una representación tridimensional de la superficie del objeto, mientras que MLFFNN provee una representación volumétrica en el sentido en que información al exterior y al interior del objeto se utiliza durante el entrenamiento. Las NGN presentan el mejor desempeño, seguidas por los SOM y las MLFFNN. En términos de costo computacional, las NGN son las que demandan más tiempo de entrenamiento, mientras que los SOM demandan el menor. Finalmente, las MLFFNN presentan los mayores errores de representación, tanto para entrenamiento con Levenberg–Marquardt (LM) como con Filtro de Kalman (KF), sin embargo, el entrenamiento con KF requiere menos épocas de entrenamiento que con LM para alcanzar el desempeño deseado, lo que indicaría su más rápida



(a) Entrenamiento grueso

(b) Entrenamiento fino

Figura 4-13.: Entrenamiento grueso y fino empleado para el modelado 3D

convergencia.

Con respecto al tiempo de visualización, las MLFFNN tienen el costo computacional más alto debido a que la red codifica la morfología del objeto en las conexiones internas y pesos sinápticos de las capas ocultas, por lo que se requiere un conjunto de datos de prueba para hacer que la red opere como clasificador y retorne los puntos en el espacio 3D que corresponden al interior, a la superficie o al exterior del objeto. En efecto, una MLFFNN es un clasificador entrenado para aprender el modelo volumétrico de un objeto de realidad virtualizada, por lo que el conjunto de datos de prueba es la entrada de la red, y la visualización consiste en graficar solo aquellos puntos cuya respuesta sea menor o igual que cero, es decir, los puntos que están sobre la superficie y al interior del objeto.

Por otra parte, los modelos neuronales basados en SOM y NGN codifican directamente la morfología del objeto en sus pesos sinápticos, los cuales son entendidos como posiciones en el espacio tridimensional. En este sentido, la visualización es inmediata, y solo consiste en graficar las neuronas de la red en el espacio 3D de referencia. Cabe aclarar que la visualización de los SOM es ligeramente más lenta que la de las NGN debido al tiempo de renderización que implica graficar la malla de conexiones de la red.

Una de las aplicaciones más interesantes del modelado 3D empleando computación neuronal es la posibilidad de modelar objetos variables en el tiempo, o cuya información no se conoce completamente al inicio del entrenamiento, sino que se va actualizando a medida que pasa el tiempo. Uno de los métodos de adquisición aquí nombrados, la ENS (explicada en detalle en el Capítulo 3), permite la adquisición en tiempo real y en escenarios intra–operativos de in-

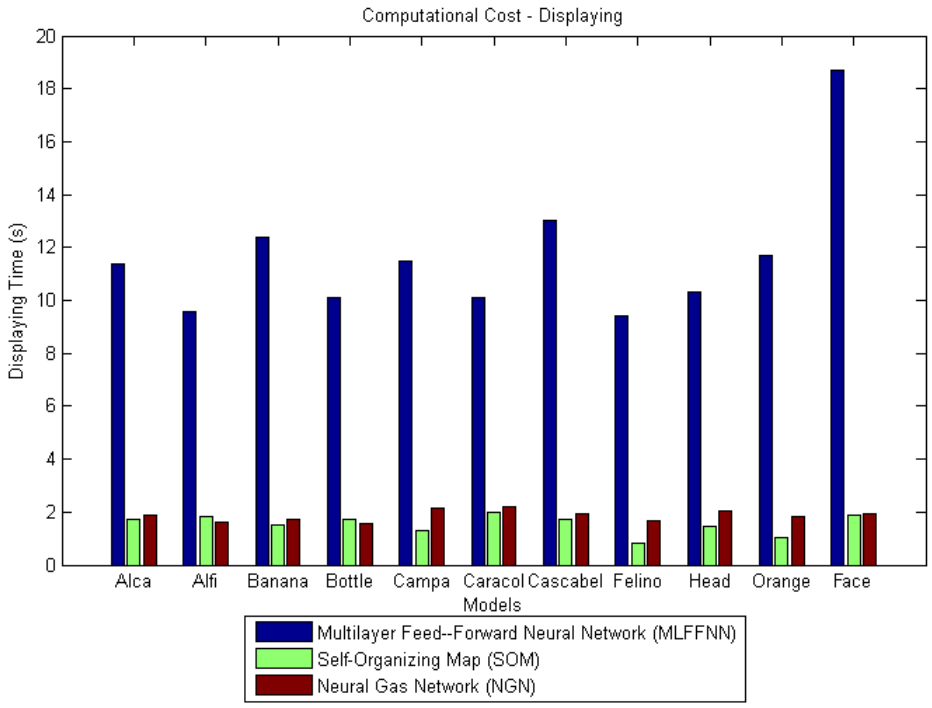


Figura 4-14.: Comparación del tiempo de visualización

formación 3D de estructuras cerebrales, tales como tumores, a partir de videos endoscópicos y de ultrasonido. Esto significa que mientras el neurocirujano manipula el equipo, durante la intervención quirúrgica, la información 3D que define la morfología del tumor se actualiza constantemente y debe emplearse para adaptar el modelo de representación. Las redes neuronales ofrecen una posibilidad interesante en este punto, ya que el modelado tridimensional puede desarrollarse en dos etapas: una etapa de entrenamiento fuera de línea, como la presentada en este capítulo, que permite generar un modelo inicial del tumor a partir de información ENS preliminar; y una etapa de entrenamiento en línea, que permite ajustar el modelo de acuerdo a la nueva información ENS recibida.

5. Modelado neuronal acelerado por hardware usando GPU

Las unidades de procesamiento gráfico (GPU - "Graphics Processing Unit") se han convertido en una solución informática de escritorio para la computación científica paralela a gran escala [124]. En este capítulo, se exploran sus posibilidades y limitaciones en la aceleración del algoritmo NGN para el modelado 3D de objetos de realidad virtualizada. Como se explica en la Sección 4.2, el algoritmo NGN es demandante en términos computacionales debido a que debe realizarse una búsqueda exhaustiva para encontrar y actualizar las neuronas ganadoras. Este procedimiento se puede implementar paralelamente en una GPU de forma que se reduzca drásticamente su costo computacional. En varios artículos, el potencial de la tecnología GPU ha sido demostrada en problemas de optimización no–lineal [125], búsqueda de patrones [124], problemas de asignación cuadrática [75] y registro de objetos 3D [122]. El objetivo es determinar cómo y cuánto las redes neuronales gas pueden ser potencialmente aceleradas en una plataforma GPU. Los algoritmos son implementados bajo la Arquitectura Unificada CUDA TM ("Compute Unified Device Architecture"), usando la librería "Thrust" [55], en una tarjeta NVIDIA GeForce GTX 460 SE [85].

5.1. Aceleración del algoritmo NGN

Los principales pasos del entrenamiento NGN se describen en el Algoritmo 2. La longitud de entrenamiento se mide en épocas de entrenamiento o en períodos de adaptación [79].

$$h_\lambda\left(k_i\left(x,w\right)\right) = \exp\left(-\frac{k_i\left(x,w\right)}{\lambda}\right) \tag{5-2}$$

Debido a que el entrenamiento NGN es un algoritmo iterativo, el proceso requiere que cada uno de los pasos se complete antes que el siguiente, por lo que no es un problema que permita paralelización completa. Sin embargo, es posible paralelizar individualmente cada uno de los pasos descritos anteriormente.

Debido a los altos tiempos de procesamiento, la gran cantidad de datos de entrenamiento y la larga longitud de los vectores de pesos sinápticos de la red, los pasos 2, 3 y 4 del Algoritmo 2 pueden ejecutarse de manera eficiente en una GPU. De esta manera, el comportamiento

Algoritmo 2 Algoritmo de entrenamiento de una red neuronal gas

[***Paso 1:***] Escoger aleatoriamente un vector de entrada $\mathbf{x}$.
[***Paso 2:***] Calcular la distancia euclidiana desde todas las neuronas de la red hasta el vector de entrada.
[***Paso 3:***] Ordenar ascendentemente estas distancias para obtener el ranking de vecindario.
[***Paso 4:***] Actualizar la neurona ganadora y las neuronas del ranking del vecindario de acuerdo a la regla de adaptación:

$$\Delta \mathbf{w}_i = \varepsilon \cdot h_\lambda \left(k_i \left(\mathbf{x}, \mathbf{w}\right)\right) \cdot \left(\mathbf{x} - \mathbf{w}_i\right) \tag{5-1}$$

[***Paso 5:***] Repetir el ***Paso 1*** hasta que se alcance la longitud de entrenamiento deseada.

general del proceso de aprendizaje sigue siendo secuencial, pero sus pasos individuales se paralelizan mediante un modelo de aceleración por hardware.
La determinación del ranking del vecindario y la actualización de los pesos sinápticos de la red mediante las Ecuaciones 5-1 y 5-2, se implementa en la GPU mediante la librería "Thrust" [55]. Adicionalmente, la librería "Templates" permite una rápida implementación de las transformaciones necesarias para el cálculo de la distancia euclidiana y la actualización de las neuronas usando las características de alto rendimiento de la GPU.
El ranking de vecindario se calcula como $(\mathbf{w}_{i0}, \mathbf{w}_{i1}, ..., \mathbf{w}_{i(N-1)})$, donde $\mathbf{w}_{i0}$ y $\mathbf{w}_{i(N-1)}$ son la neurona más cercana y más lejana al vector de entrada $\mathbf{x}$, respectivamente. Este cálculo requiere una implementación que funcione rápidamente con conjuntos de datos de más 10^6 puntos. Los algoritmos *back40Computing*, incluidos en la librería *Thrust*, son apropiados para esta tarea [82].
En este sentido, el tiempo global de entrenamiento NGN depende tanto del ordenamiento como de la longitud del entrenamiento. Con una implementación en GPU, la complejidad del ordenamiento tiende asintóticamente a $O(n)$, por lo que el tiempo de entrenamiento final puede escalarse linealmente de acuerdo a la cantidad de vectores de entrada, lo que no es posible en una implementación en una Unidad Central de Procesamiento (CPU - "Central Processing Unit"), donde la complejidad aumenta no-linealmente [82]. La longitud del entrenamiento depende del conjunto de datos de entrada y la calidad de representación esperada.

5.2. Pruebas experimentales

Para determinar el potencial de la aceleración por hardware, el algoritmo NGN se adapta a un esquema de programación paralela en GPU, y se compara con una implementación serial en una plataforma CPU.
Para la implementación en GPU, se utiliza una tarjeta gráfica NVIDIA GeForce GTX 460 SE

con un procesador AMD Phenon, bajo el sistema operativo Windows 7. Mientras que para la implementación en CPU, se utiliza una implementación en Matlab 10a en un procesador Intel Core i5 M450, bajo el sistema operativo Windows 7.
En la Figura **5-1** se muestra una comparación del tiempo de entrenamiento, en GPU y CPU, para un conjunto de entrenamiento fijo, variando el número de neuronas en la red. Esta comparación se presenta para dos objetos diferentes de realidad virtualizada. Para la implementación en GPU se observa como el tiempo de entrenamiento no aumenta considerablemente para redes con muchas neuronas, mientras que para CPU, este tiempo es altamente dependiente del número de neuronas. En los dos casos presentados, hay un punto de cruce de las curvas GPU y CPU cerca de 2×10^4 neuronas, lo que indica un umbral a partir del cual el tiempo de procesamiento en GPU es menor que el tiempo en CPU.

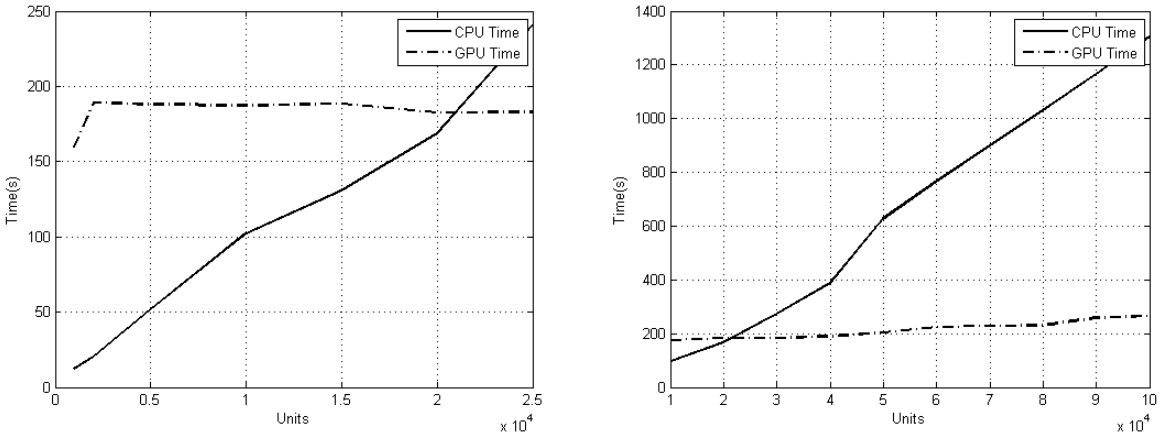
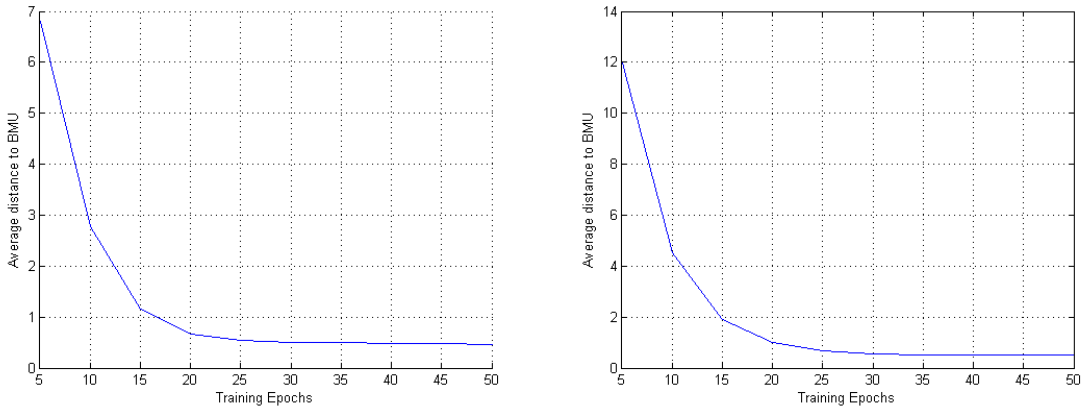


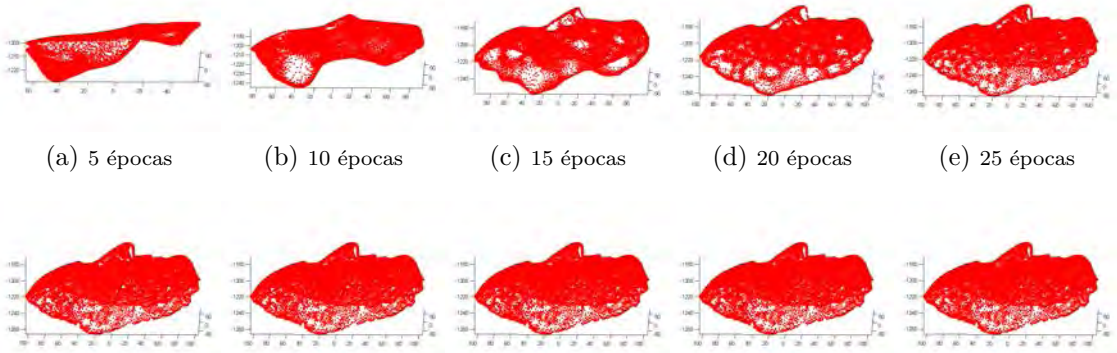
Figura 5-1.: Comparación del tiempo de entrenamiento en GPU y CPU

Para encontrar una longitud de entrenamiento óptima, se utilizan dos conjuntos de datos para entrenar redes NGN con diferente número de épocas de entrenamiento. Se utilizan intervalos de cinco épocas, y el error se mide como la distancia euclidiana promedio entre los puntos de entrada y la superficie del modelo. Las Figuras **5-2**a y **5-2**b muestran los resultados del modelado 3D de dos conjuntos de datos, un rostro humano y una escultura de un gato, con 3×10^4 y $1{,}1 \times 10^5$ puntos, respectivamente. En general, el desempeño de la representación converge a un valor constante después de la 30ª época de entrenamiento. Por este motivo, la longitud de entrenamiento óptima se establece en este valor para las siguientes pruebas. La Figura **5-3** ilustra la evolución del modelado 3D con una NGN con $1{,}5 \times 10^4$ neuronas para una nube de entrada con 3×10^4 puntos pertenecientes a un rostro humano.
Para determinar el rendimiento del algoritmo NGN acelerado, se usan varios conjuntos de entrada para entrenar redes de diferentes tamaños. Los objetos virtualizados de prueba se modelan mediante NGN cuyo número de neuronas varía desde el 10 % hasta el 300 % del



(a) Nube de puntos con 29421 datos (b) Nube de puntos con 112861 datos

Figura 5-2.: Error del modelado 3D para diferentes épocas de entrenamiento



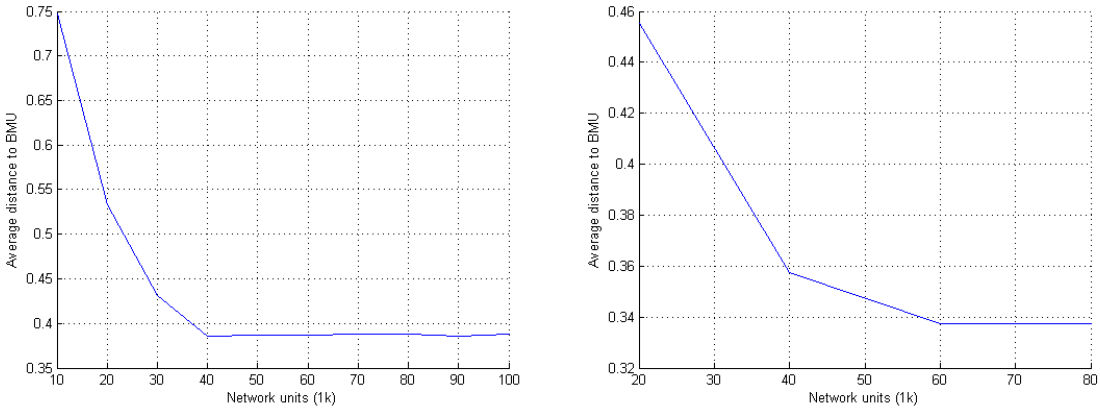
(a) 5 épocas (b) 10 épocas (c) 15 épocas (d) 20 épocas (e) 25 épocas

(f) 30 épocas (g) 35 épocas (h) 40 épocas (i) 45 épocas (j) 50 épocas

Figura 5-3.: Evolución del modelado NGN durante el entrenamiento

número de datos en el conjunto de entrada. La Figura **5-4** presenta el error para dos conjuntos de datos modelados con redes de diferentes tamaños. En estos resultados se observa que los mejores desempeños se alcanzan cuando el número de neuronas de la red es igual al número de datos en la nube de puntos de entrada, lo que implica que el sobremuestreo no genera ninguna ganancia significativa en la calidad del modelado con NGN.

En cuanto al tiempo de entrenamiento del algoritmo NGN, la paralelización mediante GPU disminuye el tiempo de procesamiento hasta la quinta parte del tiempo requerido en una implementación CPU cuando se tienen redes neuronales con más de 15 mil neuronas. Los resultados para diferentes tamaños de red se presentan en la Tabla **5-1**. El tiempo de entrenamiento en la implementación CPU se estima como $t_{CPU} = n \times a_{CPU}$, donde n es la longitud del entrenamiento y a_{CPU} es el tiempo de procesamiento de una sola etapa de adaptación en una implementación CPU. La estimación de a_{CPU} se realiza a partir de los datos usados



(a) Nube de puntos con 29421 datos (b) Nube de puntos con 112861 datos

Figura 5-4.: Desempeño del modelado para diferentes números de neuronas de la red

para los cálculos de la Figura **5-4**.

Tabla 5-1.: Aceleración del modelado 3D para diferentes objetos virtualizados

Neuronas	**Longitud de entrenamiento**	**Tiempo medido en GPU (segundos)**	**Tiempo estimado en CPU (segundos)**	**Aceleración**
10000	882630	3780.5	2013.5	0.5x
15000	882630	3811.3	2750.6	0.7x
15000	3385800	11035.4	10551.3	1.0x
20000	882630	3970.6	3547.4	0.9x
30000	882630	4054.7	5746.9	1.4x
40000	882630	4125.5	8181.0	2.0x
50000	882630	4348.7	13209.2	3.0x
60000	882630	4768.6	16134.1	3.4x
70000	882630	4812.1	18908.7	3.9x
80000	882630	4868.3	21679.7	4.5x
90000	882630	5444.8	24530.6	4.5x
100000	882630	5500.4	27444.6	5.0x

5.3. Resumen y conclusiones

Para determinar el potencial para aplicación en tiempo real del modelado 3D con redes neuronales, se implementó el algoritmo NGN de manera paralela en una GPU NVIDIA y se presentaron comparaciones experimentales contra una implementación serial en una

CPU usando Matlab. Las comparaciones son realizadas en términos de desempeño, costo computacional, tiempo de entrenamiento y calidad de la representación.
Respecto al tiempo de entrenamiento, en general, la implementación GPU es más veloz que la implementación CPU. Sin embargo, siempre hay un valor particular de número de neuronas de la red, para el cual la velocidad en GPU y CPU es igual. Este valor crítico depende del hardware utilizado, pero dada la complejidad del algoritmo de ordenamiento, la implementación GPU siempre encuentra la situación de mejor desempeño. Para el hardware utilizado en este trabajo, la implementación GPU es más rápida que la implementación CPU para redes neuronales con más de 2×10^4 neuronas.
Como investigación futura, se plantea la aceleración de otras arquitecturas neuronales, como MLFFNN y SOM, en la tarea de modelado 3D. Además de la aceleración de la etapa de entrenamiento, se trabajará en encontrar maneras más veloces y eficientes de visualización de los modelos neuronales generados.

6. Resultados: modelado 3D de tumores cerebrales

Una de las aplicaciones más interesantes del modelado 3D empleando computación neuronal es la posibilidad de modelar objetos variables en el tiempo, o cuya información no se conoce completamente al inicio del entrenamiento, sino que se actualiza a medida que pasa el tiempo. La Endoneurosonografía (ENS), permite la adquisición en tiempo real y en escenarios intra–operativos de información 3D de estructuras cerebrales, tales como tumores, a partir de videos endoscópicos y de ultrasonido. Esto significa que mientras el neurocirujano manipula el equipo, durante la intervención quirúrgica, la información 3D que define la morfología del tumor se actualiza constantemente y debe emplearse para adaptar el modelo de representación. Las redes neuronales ofrecen una posibilidad interesante en este punto, ya que el modelado tridimensional puede desarrollarse en dos etapas: una etapa de entrenamiento fuera de línea que permite generar un modelo inicial del tumor a partir de información ENS preliminar, y una etapa de entrenamiento en línea que permite ajustar el modelo de acuerdo a la nueva información ENS recibida. Dados los resultados de modelado obtenidos en el Capítulo 4, en los que se muestra la superioridad de las arquitecturas NGN y SOM sobre la MLFFNN en la tarea de modelado 3D, se utilizan redes neuronales auto–organizadas para el modelado en línea de tumores cerebrales. En la Figura **6-1** se muestra el esquema de modelado 3D de tumores cerebrales empleando redes neuronales artificiales. En este capítulo se presenta la aplicación de la metodología propuesta en los capítulos anteriores a la tarea específica del modelado 3D de tumores cerebrales. Los resultados experimentales corresponden a pruebas con modelos cerebrales fantasma de una base de datos ENS propia.

6.1. Adaptación o entrenamiento secuencial

Existen dos estilos de entrenamiento de redes neuronales: el primero, la adaptación o entrenamiento secuencial, en el que los pesos sinápticos se actualizan cada vez que se presenta un dato de entrada a la red. El segundo, el entrenamiento por lotes, en el que los pesos sinápticos se actualizan después de que se presentan todos los datos de entrada. Generalmente, el entrenamiento por lotes es más eficiente computacionalmente, sin embargo, en algunas aplicaciones no es posible disponer de todos los datos al inicio del entrenamiento, por lo que el paradigma de entrenamiento secuencial es más útil. Cuando se opera en un ambiente no estacionario, a mayor adaptación mayor desempeño [103].

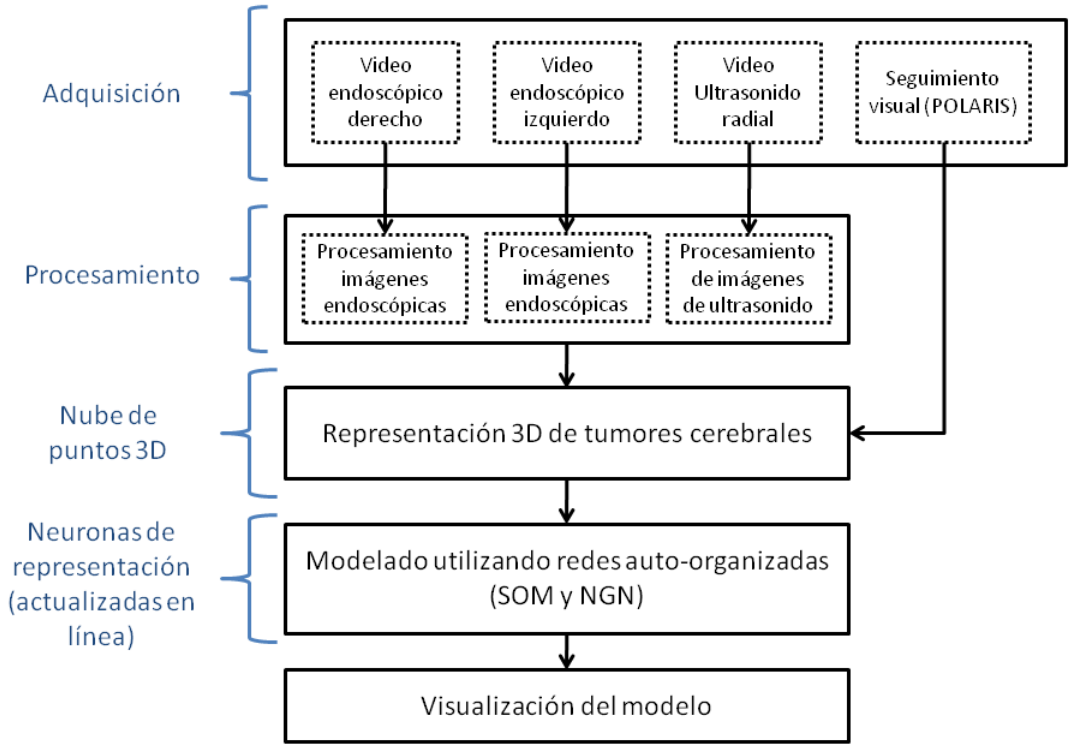


Figura 6-1.: Esquema de modelado 3D de tumores cerebrales

Cuando una red neuronal opera en un ambiente estacionario (en el cual las características estadísticas no cambian con el tiempo), una red neuronal puede aprender las características estadísticas del ambiente mediante aprendizaje fuera de línea. En particular, los pesos sinápticos de la red pueden ajustarse mediante el entrenamiento con un conjunto representativo de datos de entrada. Una vez el proceso de entrenamiento esté completado, la red debería haber capturado las características principales del espacio de entrada.

Sin embargo, el ambiente de interés puede ser no estacionario, lo que significa que los parámetros estadísticos del espacio de entrada pueden variar con el tiempo. En estas situaciones, los métodos tradicionales de entrenamiento no son adecuados para hacer seguimiento de estas variaciones estadísticas durante el entrenamiento. Para solucionar esto, es deseable que la red continuamente adapte sus parámetros de acuerdo a los datos de entrada. En ocasiones, este entrenamiento puede aplicarse en tiempo real, aunque depende de la aplicación y del tipo de datos de entrada. En un sistema adaptativo, el ajuste se realiza cada vez que se presenta un patrón de entrada, es decir, el entrenamiento nunca se detiene, por lo que generalmente se denomina aprendizaje continuo o secuencial [49].

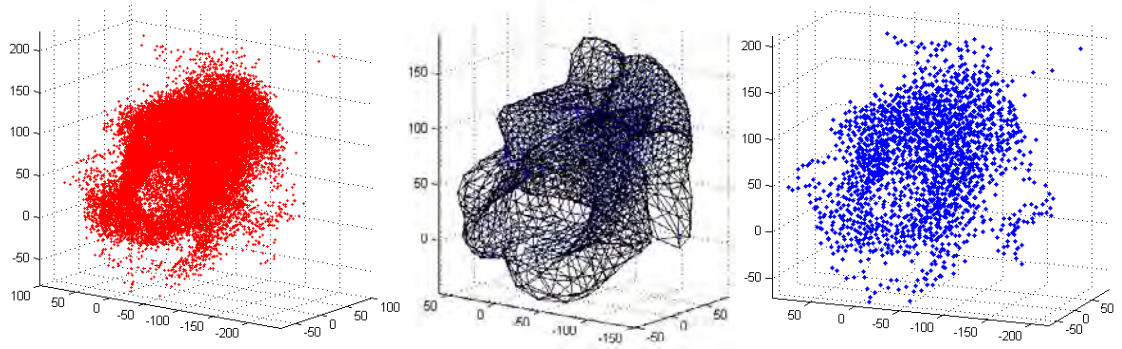
6.2. Resultados

Para la parte experimental, se dispone de una base de datos endoneurosonográfica de 7×10^4 imágenes de tumores fantasma. Esta base de datos corresponde a 4.86 horas de adquisición con el equipo ENS disponible en el laboratorio de Control Automático del CINVESTAV Guadalajara (México). Parte de la base de datos no se considera para las pruebas debido a varios problemas: a) falta de sincronización en la adquisición de las imágenes endoscópicas y de ultrasonido; b) pérdida de visibilidad de la sonda de ultrasonido por intervalos de tiempo demasiado largos, por lo que no fue posible computar un seguimiento correcto; c) errores y desajustes en el sistema de seguimiento visual (POLARIS). Todos estos problemas contribuyen a que no sea posible la reconstrucción 3D a partir de la información endoneurosonográfica.
La cantidad de información de cada cuadro ENS depende de las regiones segmentadas en cada una de las imágenes de ultrasonido. En la Tabla **6-1** se muestra la información de 6 secuencias ENS seleccionadas para las pruebas experimentales. Para cada una de estas secuencias, se utiliza el 10 % de los datos para realizar el modelo inicial del tumor usando un esquema de entrenamiento fuera de línea. A partir de allí, el 90 % restante de los datos se presenta secuencialmente para la adaptación en línea del modelo de representación.

Tabla 6-1.: Secuencias ENS empleadas para las pruebas experimentales

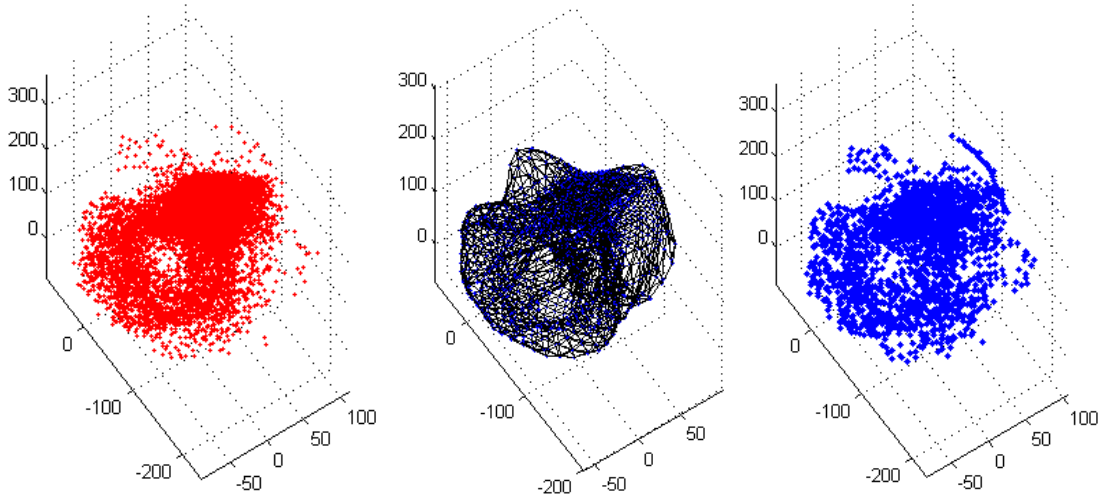
Secuencia	Número de Cuadros	Duración (segundos)	Número de Puntos 3D
1	20	5.0	$1{,}9 \times 10^4$
2	164	41.0	$1{,}1 \times 10^6$
3	254	63.5	$1{,}5 \times 10^6$
4	790	197.5	$2{,}3 \times 10^6$
5	580	145.0	$6{,}1 \times 10^5$
6	4582	1145.5	$1{,}9 \times 10^7$

Las Figuras **6-2** y **6-3** muestran ejemplos de modelado tridimensional de tumores cerebrales fantasma mediante información endoneurosonográfica y redes neuronales auto–organizadas. Los modelados presentados corresponden a la etapa fuera de línea del entrenamiento con nubes de puntos de 23421 y 9116 datos, respectivamente. A partir de este modelo, el proceso pasa a una etapa de entrenamiento en línea para adaptar la forma del modelo de acuerdo a la nueva información recibida. En ambos casos, se usan arquitecturas auto–organizadas con 2000 neuronas de representación.
La Figura **6-4** muestra el modelado 3D de un tumor cerebral empleando 20 cuadros ENS tomados a 4 cuadros por segundo (Secuencia 1). Este video corresponde a una adquisición de 5 segundos de duración. El modelado se lleva a cabo mediante una red neuronal SOM de 2000 neuronas organizadas en una malla cilíndrica con vecindario hexagonal. Para el entrenamiento fuera de línea se emplean 5 épocas de entrenamiento grueso y 10 épocas de



(a) Nube original: 23421 puntos (b) SOM. Error = 6.0414 (c) NGN. Error = 3.8116

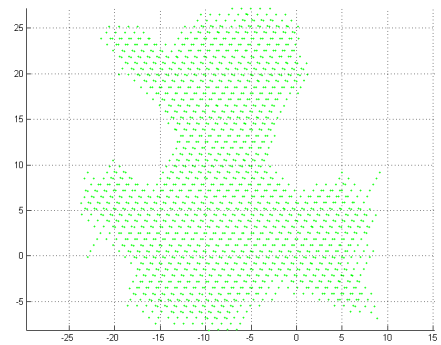
Figura 6-2.: Modelado 3D de tumores cerebrales usando ENS y redes neuronales



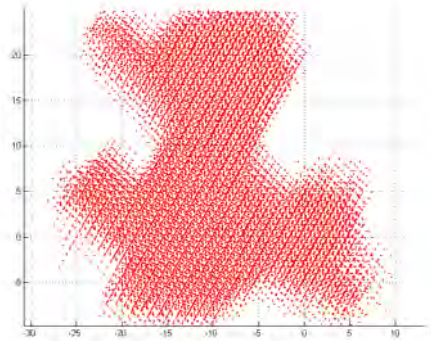
(a) Nube original: 9116 puntos (b) SOM. Error = 6.48 (c) NGN. Error = 4.14

Figura 6-3.: Modelado 3D de tumores cerebrales usando ENS y redes neuronales

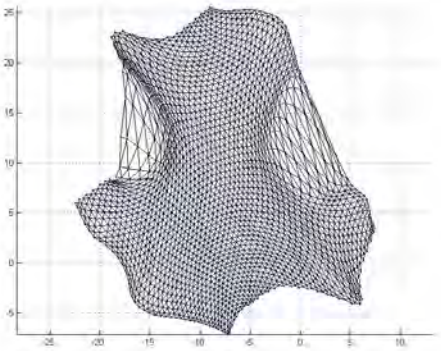
entrenamiento fino, con $1{,}9 \times 10^3$ datos (Figura **6-4**a). Para el entrenamiento en línea, se utilizan $1{,}7 \times 10^4$ puntos 3D tomados del resto de la secuencia ENS. Esta información se presenta secuencialmente para la adaptación de la red (Figura **6-4**b). La Figura **6-4**c muestra el resultado del modelado SOM, mientras que la Figura **6-4**d muestra el mismo resultado pero ocultando las conexiones de la malla. Los errores topográficos y de cuantización son 0.0557 y 13.5061, respectivamente. En la Figura **6-4**c se observa un error en el modelado de las fronteras laterales del tumor, esto se debe a que las conexiones de la malla no permiten que las neuronas del mapa modelen adecuadamente los agujeros y las fronteras de los objetos. La Figura **6-5** muestra el modelado 3D de un tumor cerebral empleando 164 cuadros ENS tomados a 4 cuadros por segundo (Secuencia 2). Este video corresponde a una adquisición de 41 segundos de duración. El modelado se lleva a cabo mediante una red neuronal SOM de 2000 neuronas organizadas en una malla cilíndrica con vecindario hexagonal. Para el



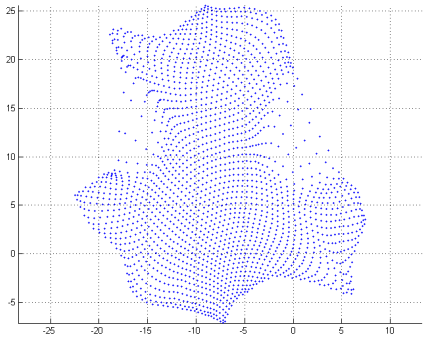
(a) Datos para etapa fuera de línea



(b) Datos para etapa en línea



(c) SOM con conexiones

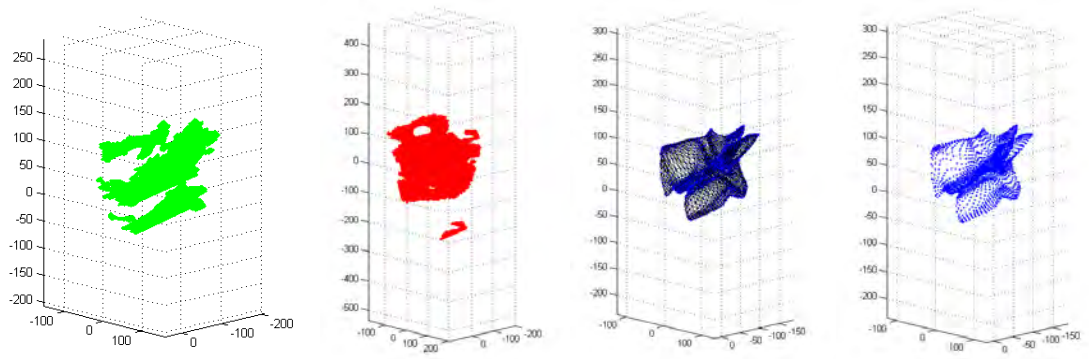


(d) SOM sin conexiones

Figura 6-4.: Modelado 3D del tumor de la secuencia 1 usando SOM

entrenamiento fuera de línea se emplean 5 épocas de entrenamiento grueso y 10 épocas de entrenamiento fino, con $1{,}1 \times 10^5$ puntos 3D adquiridos de los primeros 4.1 segundos de la secuencia ENS (Figura **6-5**a). Para el entrenamiento en línea, se utilizan $9{,}9 \times 10^5$ puntos 3D tomados del resto de la secuencia ENS. Esta información se presenta secuencialmente para la adaptación de la red (Figura **6-5**b). Las Figuras **6-5**d y **6-5**c muestran los resultados del modelado SOM mostrando y ocultando las conexiones de la malla, respectivamente. Los errores topográficos y de cuantización son 0.1149 y 36.86, respectivamente.

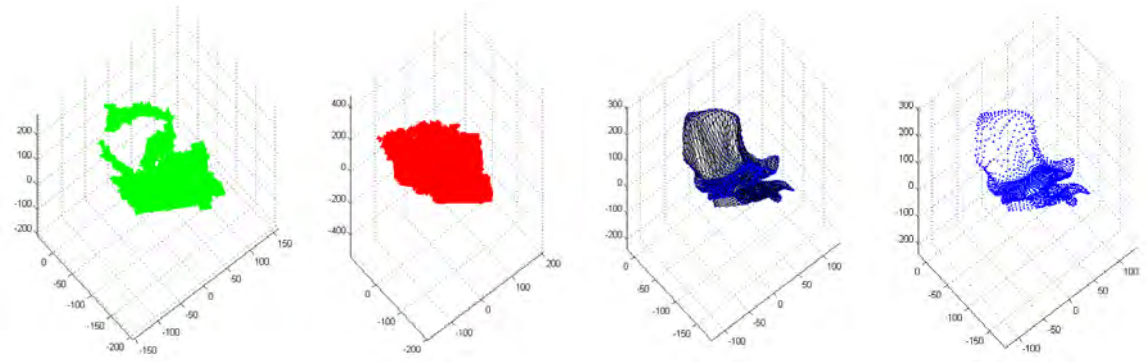
La Figura **6-6** muestra el modelado 3D de un tumor cerebral empleando 254 cuadros ENS tomados a 4 cuadros por segundo (Secuencia 3). Este video corresponde a una adquisición de 63.5 segundos de duración. El modelado se lleva a cabo mediante una red neuronal SOM de 2000 neuronas organizadas en una malla cilíndrica con vecindario hexagonal. Para el entrenamiento fuera de línea se emplean 5 épocas de entrenamiento grueso y 10 épocas de entrenamiento fino, con $1{,}5 \times 10^5$ puntos 3D adquiridos de los primeros 6.35 segundos de la secuencia ENS (Figura **6-6**a). Para el entrenamiento en línea, se utilizan $1{,}3 \times 10^6$ puntos 3D tomados del resto de la secuencia ENS. Esta información se presenta secuencialmente para la adaptación de la red (Figura **6-6**b). Las Figuras **6-6**d y **6-6**c muestran los resultados



(a) Datos etapa fuera de línea (b) Datos etapa en línea (c) SOM con conexiones (d) SOM sin conexiones

Figura 6-5.: Modelado 3D del tumor de la secuencia 2 usando SOM

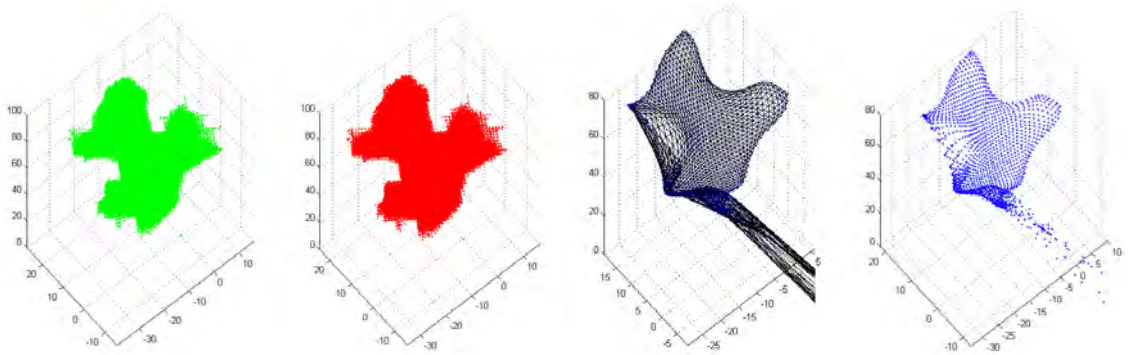
del modelado SOM con y sin visualización de las conexiones, respectivamente. Los errores topográficos y de cuantización son 0.1258 y 33.75, respectivamente.



(a) Datos etapa fuera de línea (b) Datos etapa en línea (c) SOM con conexiones (d) SOM sin conexiones

Figura 6-6.: Modelado 3D del tumor de la secuencia 3 usando SOM

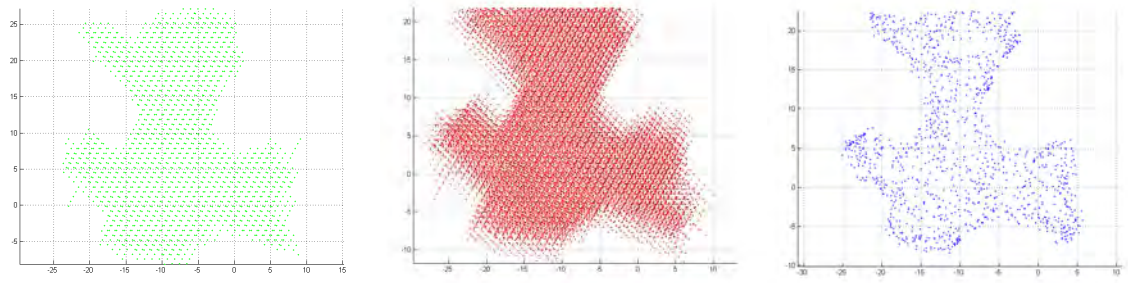
La Figura **6-7** muestra el modelado 3D de un tumor cerebral empleando 790 cuadros ENS tomados a 4 cuadros por segundo (Secuencia 4). Este video corresponde a una adquisición de 197.5 segundos de duración. El modelado se lleva a cabo mediante una red neuronal SOM de 2000 neuronas organizadas en una malla cilíndrica con vecindario hexagonal. Para el entrenamiento fuera de línea se emplean 5 épocas de entrenamiento grueso y 10 épocas de entrenamiento fino, con $2{,}3 \times 10^5$ puntos 3D adquiridos de los primeros 19.75 segundos de la secuencia ENS (Figura **6-7**a). Para el entrenamiento en línea, se utilizan $2{,}1 \times 10^6$ puntos 3D tomados del resto de la secuencia ENS. Esta información se presenta secuencialmente para la adaptación de la red (Figura **6-7**b). Las Figuras **6-7**d y **6-7**c muestran los resultados del modelado SOM. Los errores topográficos y de cuantización son 0.9758 y 0.0562, respectivamente.



(a) Datos fuera de línea (b) Datos etapa en línea (c) SOM con conexiones (d) SOM sin conexiones

Figura 6-7.: Modelado 3D del tumor de la secuencia 4 usando SOM

La Figura **6-8** muestra el modelado 3D de un tumor cerebral empleando 20 cuadros ENS tomados a 4 cuadros por segundo (Secuencia 1). Este video corresponde a una adquisición de 5 segundos de duración. El modelado se lleva a cabo mediante una red NGN de 2000 neuronas. Para el entrenamiento fuera de línea se emplean 5 épocas de entrenamiento grueso y 10 épocas de entrenamiento fino, con $1{,}9 \times 10^3$ puntos 3D adquiridos en el primer 10 % de la secuencia ENS (Figura **6-8**a). Para el entrenamiento en línea, se utilizan $1{,}7 \times 10^4$ puntos 3D tomados del resto de la secuencia. Esta información se presenta secuencialmente para la adaptación de la red (Figura **6-8**b). El error de cuantización después del entrenamiento es 1.4112.

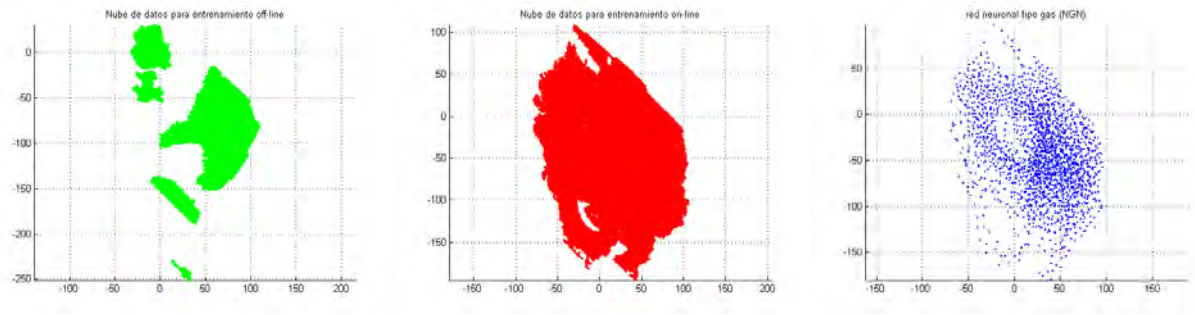


(a) Datos etapa fuera de línea (b) Datos etapa en línea (c) Modelo NGN

Figura 6-8.: Modelado 3D del tumor de la secuencia 1 usando NGN

La Figura **6-9** muestra el modelado 3D de un tumor cerebral empleando 790 cuadros ENS tomados a 4 cuadros por segundo (Secuencia 4). Este video corresponde a una adquisición de 197.5 segundos de duración. El modelado se lleva a cabo mediante una red NGN de 2000 neuronas. Para el entrenamiento fuera de línea se emplean 5 épocas de entrenamiento grueso y 10 épocas de entrenamiento fino, con $2{,}3 \times 10^5$ puntos 3D adquiridos de los primeros 19.75 segundos de la secuencia ENS (Figura **6-9**a). Para el entrenamiento en línea, se utilizan

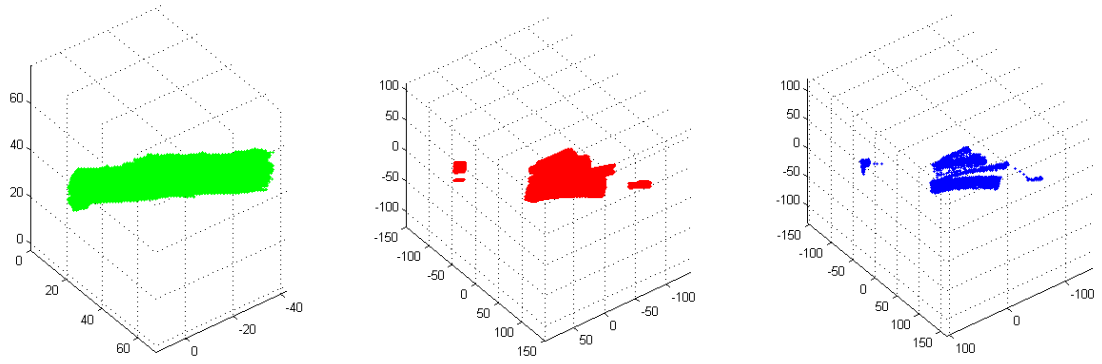
$2{,}1 \times 10^6$ puntos 3D tomados del resto de la secuencia. Esta información se presenta secuencialmente para la adaptación de la red (Figura **6-9**b). El error de cuantización después del entrenamiento es 4.1560.



(a) Datos etapa fuera de línea (b) Datos etapa en línea (c) Modelo NGN

Figura 6-9.: Modelado 3D del tumor de la secuencia 4 usando NGN

La Figura **6-10** muestra el modelado 3D de un tumor cerebral empleando 580 cuadros ENS tomados a 4 cuadros por segundo (Secuencia 5). Este video corresponde a una adquisición de 145 segundos de duración. El modelado se lleva a cabo mediante una red NGN de 2000 neuronas. Para el entrenamiento fuera de línea se emplean 5 épocas de entrenamiento grueso y 10 épocas de entrenamiento fino, con $6{,}1 \times 10^4$ puntos 3D adquiridos de los primeros 14.5 segundos de la secuencia ENS (Figura **6-9**a). Para el entrenamiento en línea, se utilizan $5{,}5 \times 10^5$ puntos 3D tomados del resto de la secuencia. Esta información se presenta secuencialmente para la adaptación de la red (Figura **6-9**b). El error de cuantización después del entrenamiento es 1.4640. Para efectos de interpretación, nótese que los ejes de la Figura **6-10**a están en un rango diferente.



(a) Datos etapa fuera de línea (b) Datos etapa en línea (c) Modelo NGN

Figura 6-10.: Modelado 3D del tumor de la secuencia 5 usando NGN

En la Tabla **6-2** se muestran los errores del modelado tridimensional de tumores cerebrales

usando SOM y NGN. El error de cuantización se calcula como la distancia euclidiana promedio desde los puntos del tumor hasta la neurona más cercana en la red (BMU). El error topográfico es una medida cuantitativa de la preservación topológica del SOM después del entrenamiento, y se calcula como la proporción de todos los datos de entrada para los cuales la neurona más cercana y la segunda neurona más cercana no son adyacentes dentro de la malla de la red. El error topográfico de una red NGN es cero [114].
Note que el modelo SOM presenta un error topográfico de más del 50 % para la Secuencia 5. Generalmente, esto se debe a una inicialización inadecuada de la red, la cual se realiza aleatoriamente sobre el eje principal de la nube de datos de entrenamiento fuera de línea. Dado que en el inicio del modelado no se conoce la morfología del tumor, no es posible garantizar la representatividad de este conjunto de entrenamiento respecto a los datos futuros. Adicionalmente, esta mala inicialización se refleja en un alto error de cuantización.

Tabla 6-2.: Error topográfico y de cuantización del modelado 3D

	SOM		**NGN**
Secuencia	**Error Topográfico**	**Error de Cuantización**	**Error de Cuantización**
1	0.0557	13.51	1.4112
2	0.1149	36.86	3.3886
3	0.1258	33.75	2.7592
4	0.0562	9.76	4.1560
5	0.5585	68.09	1.4640

De acuerdo a la correspondencia entre las unidades de medida en el equipo ENS y las unidades de medida en el mundo real (Capítulo 3), cada unidad de medida en los datos de entrada corresponde a 0.1875 mm en el mundo real. En la Tabla **6-3** se presentan los errores de cuantización de los modelos 3D de tumores cerebrales medidos en milímetros.
En el trabajo de Letteboer y otros [67], se presenta un estudio clínico con 12 pacientes a los cuales se les estima el movimiento y la deformación de sus estructuras cerebrales durante una neurocirugía. El método de adquisición es ultrasonido 3D, y se reportan deformaciones verticales promedio de 3.0 mm con un valor máximo de 7.5 mm, y deformaciones horizontales promedio de 3.9 mm con un valor máximo de 8.2 mm. De acuerdo a estos resultados, el modelado NGN propuesto en esta tesis tiene la precisión suficiente para detectar estos cambios en las estructuras cerebrales. Como se mostró en el Capítulo 5, es posible aumentar el número de neuronas de la red con el fin de mejorar la calidad de representación, sin incurrir en un aumento considerable del costo computacional, lo que puede ser la solución en caso de que el modelo requiera mayor resolución. Debido a los errores reportados, el modelado SOM no garantiza, al menos con este número de neuronas, este nivel de detalle en la representación, por lo que no sería adecuado para este tipo de casos clínicos.
En otros trabajos [80, 54, 81], que utilizan resonancia magnética como técnica de adquisición, se han reportado pacientes con deformaciones promedio entre 5 y 10mm, con valores máximos

de hasta 20mm. En estos casos, las redes SOM podrían ser útiles como herramientas de modelado, dado que sus errores de cuantización están por debajo de estos rangos.

Tabla 6-3.: Errores de cuantización del modelado 3D (milímetros)

	Error de cuantización (milímetros)	
Secuencia	**SOM**	**NGN**
1	2.5331	0.2646
2	6.9113	0.6354
3	6.3281	0.5174
4	1.8300	0.7793
5	12.7661	1.4640

6.3. Conclusiones

De acuerdo a los estudios clínicos reportados en la literatura y a los resultados obtenidos en esta tesis, las redes NGN presentan el mayor potencial para su aplicación en el modelado 3D de tumores cerebrales a partir de endoneurosonografía. Sin embargo, no se descarta la aplicación de redes SOM como herramienta de modelado, pues su precisión podría ser útil en algunas aplicaciones específicas.

Aunque el modelado 3D de tumores cerebrales se basa en un esquema de entrenamiento en línea, aún no es posible su aplicación en tiempo real debido a los altos tiempos de entrenamiento. Una de las soluciones en las que se trabaja actualmente, es la aceleración mediante hardware gráfico de las etapas de adaptación y visualización de la red. Adicionalmente, dado el relativo bajo costo del entrenamiento SOM con respecto al entrenamiento NGN, se planea desarrollar un modelo neuronal híbrido capaz de combinar la velocidad de las SOM con la capacidad de adaptación de las NGN. Como propuesta inicial, se está trabajando en un modelado NGN inicializado mediante una red SOM.

En el modelado con redes neuronales auto–organizadas, la morfología de los tumores se codifica directamente en los pesos sinápticos de la red, los cuales pueden ser entendidos como posiciones en el espacio de entrada. Por este motivo, la visualización es inmediata, pues consiste en graficar directamente las neuronas de la red sobre el espacio de entrada tridimensional. Con el modelado neuronal existe una ganancia evidente en el tiempo de visualización y espacio de almacenamiento en disco, pues en lugar de manipular la nube de puntos que representa el tumor, se trabaja únicamente con las neuronas de representación. Como regla general, el número de neuronas siempre es menor o igual que el número de puntos en la nube de entrada. Como se presenta en los resultados del Capítulo 5, cuando el número de neuronas de la red es mayor que el número de puntos en la nube de entrada, no hay ninguna mejora considerable en el desempeño, por lo que el sobremuestreo no es justificado.

El modelado con redes neuronales tiene la ventaja de que no requiere ningún conocimiento "a–priori" de la estructura que se quiere modelar, lo que es bastante adecuado en este tipo de aplicaciones, pues desde el inicio de la adquisición no se conoce la forma ni el tamaño del tumor que se desea modelar. Adicionalmente, gracias a la capacidad de adaptación de las redes, el modelo puede ajustarse dinámicamente de acuerdo a la información que proviene del equipo ENS.

7. Conclusiones y trabajo futuro

En los capítulos anteriores se han expuesto conclusiones y consideraciones específicas de acuerdo a cada unos de los aportes presentados. A continuación, se presentan conclusiones generales del trabajo, se detallan los principales aportes de la investigación y se direcciona el trabajo futuro.

En el Capítulo 3, se presenta una solución sencilla y eficaz al problema de adquisición de información tridimensional de estructuras cerebrales a partir de endoneurosonografía (ENS). El método es simple, eficiente, y razonablemente robusto a la oclusión, a los desplazamientos de la sonda y a la salida de ésta del rango de visión de las cámaras endoscópicas.

En el Capítulo 4, se describe la implementación de una metodología de modelado tridimensional utilizando computación neuronal. Se estudia el uso de tres tipos de arquitecturas neuronales para la tarea de modelado 3D: redes neuronales multicapa alimentadas hacia adelante (MLFFNN), mapas auto–organizados (SOM) y redes neuronales gas (NGN). Se presentan y discuten comparaciones experimentales entre estas arquitecturas en términos de desempeño, costo computacional, tiempo de entrenamiento, tiempo de visualización y complejidad de la arquitectura.

La arquitectura NGN presenta el menor error de cuantización y la mejor capacidad de adaptación durante las pruebas de modelado 3D, lo que indica que es la arquitectura más adecuada en términos de calidad de representación. Sin embargo, también presenta el más elevado costo computacional, lo que limita su posible aplicación en tiempo real. Por esta razón, en el Capítulo 5 se estudia el potencial, para aplicación en tiempo real, del modelado 3D con NGN usando una implementación acelerada por hardware. La implementación acelerada utiliza la arquitectura unificada CUDA TM sobre una unidad de procesamiento gráfico (GPU). Se presentan comparaciones experimentales de la implementación paralela en una GPU contra una implementación serial en una CPU. Las comparaciones se realizan en términos de desempeño, costo computacional, tiempo de entrenamiento y calidad de la representación. Las pruebas demuestran que pueden alcanzarse aceleraciones de hasta 5x.

Como investigación futura, se plantea la aceleración de otras arquitecturas neuronales, como MLFFNN y SOM, en la tarea de modelado 3D. Además de la aceleración de la etapa de entrenamiento, se trabajará en encontrar maneras más veloces y eficientes de visualización de los modelos neuronales generados. Adicionalmente, se está trabajando en el desarrollo de un modelo neuronal híbrido capaz de combinar la velocidad de las SOM con la capacidad de adaptación de las NGN.

Una de las aplicaciones más interesantes del modelado 3D empleando computación neuronal

es la posibilidad de modelar objetos variables en el tiempo, o cuya información no se conoce completamente al inicio del entrenamiento, sino que se va actualizando a medida que pasa el tiempo. En este sentido, la ENS permite la adquisición en tiempo real y en escenarios intra–operativos de información 3D de estructuras cerebrales, tales como tumores. En el Capítulo 6 se presentan los resultados del modelado 3D de estos tumores empleando redes neuronales auto–organizadas. El modelado se desarrolla en dos etapas: una etapa de entrenamiento fuera de línea, que permite generar un modelo inicial del tumor a partir de información ENS preliminar, y una etapa de entrenamiento en línea, que permite ajustar el modelo de acuerdo a la nueva información ENS recibida. Se presentan pruebas del modelado en una base de datos ENS de 7×10^4 imágenes de tumores fantasma correspondientes a 4.86 horas de adquisición. De esta base de datos, se descartan muchas de las secuencias debido a problemas en la adquisición, en la sincronización o en el sistema de seguimiento visual, por lo que las pruebas se hacen con algunas secuencias ENS seleccionadas.
Con el fin de evitar implicaciones éticas y morales, esta investigación se desarrolla en su totalidad sobre modelos fantasma del cerebro humano. Una vez que se compruebe, en el laboratorio, la precisión y la conveniencia de la técnica, podría plantearse su aplicación en escenarios intra–operativos reales. De acuerdo a los estudios clínicos reportados en la literatura y a los resultados obtenidos en esta tesis, las redes NGN presentan el mayor potencial para su aplicación en el modelado 3D de tumores cerebrales a partir de endoneurosonografía, aunque no se descarta el uso de redes SOM. Sin embargo, presentan la limitación del costo computacional, cuya solución se plantea como investigación futura.

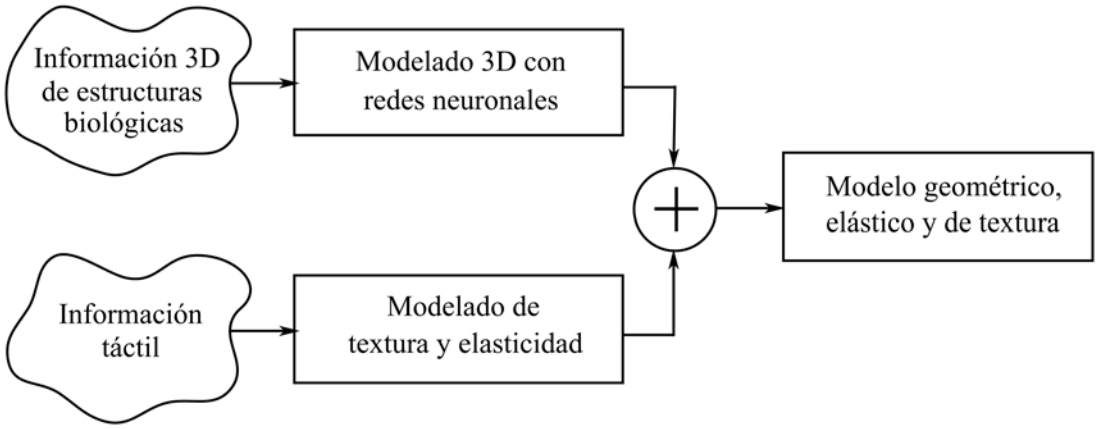


Figura 7-1.: Propuesta de trabajo futuro

Actualmente, se está mejorando el equipo quirúrgico y de adquisición disponible en el Laboratorio de Control del CINVESTAV Guadalajara (México), con el fin de agregar sensores táctiles que entreguen información de textura y elasticidad. Con esta información, se plantea desarrollar un modelo neuronal como el que se muestra en la Figura **7-1**, en el que no sólo se tenga en cuenta la geometría de la estructura biológica, sino también su textura y su elasticidad. Adicionalmente, se está instalando un brazo robótico para la manipulación

del endoscopio y de los instrumentos quirúrgicos. De esta manera, se eliminan los errores introducidos por el sistema de seguimiento visual (POLARIS) y por la manipulación incorrecta de los equipos por parte de manos inexpertas. Como complemento, se están instalando sensores de color y profundidad [84], con el fin de reconstruir, identificar y seguir objetos y personas presentes en el escenario 3D. El uso de estas tecnologías puede extenderse a otros tipos de aplicaciones médicas como descripción de órganos, detección de patologías, apoyo a intervenciones quirúrgicas, trasplantes, entrenamiento de nuevos médicos y cirugía a distancia.

A. Apéndice: Filtro de Kalman

En 1960, Rudolph Emil Kalman publicó su famoso artículo en el que describe una solución recursiva al problema de filtrado lineal de datos discretos [61]. El Filtro de Kalman (KF) es un conjunto de ecuaciones matemáticas que proveen una solución recursiva eficiente computacionalmente al problema de estimación de estados de un proceso mediante la minimización del Error Cuadrático Medio (ECM).
Dos de las principales ventajas del filtro con respecto a otros métodos de estimación son: soporta estimaciones del pasado, del presente, e incluso del futuro; funciona bien incluso cuando el modelo del sistema es desconocido [118].

A.1. Proceso de estimación

El filtrado de Kalman trata de resolver el problema general de estimar, en el instante k, el estado $w_k \in \Re^n$ de un proceso discreto gobernado por la ecuación estocástica en diferencias mostrada por la Ecuación A-1, con la medida $z_k \in \Re^m$ de la Ecuación A-2. Inicialmente, la matriz de transición A y la matriz de control B se asumen constantes en el tiempo.

$$w_k = Aw_{k-1} + Bu_{k-1} + \omega_{k-1} \tag{A-1}$$

$$z_k = Hw_k + v_k \tag{A-2}$$

Las variables aleatorias ω_k y v_k representan el ruido del proceso y el ruido de medición, respectivamente, donde las matrices de covarianza $Q \in \Re^{n\times n}$ y $R \in \Re^{m\times m}$ definen sus funciones de probabilidad normal, como se muestra en la Ecuación A-3

$$\begin{aligned} p(\omega) &\sim N(0, Q) \\ p(v) &\sim N(0, R) \end{aligned} \tag{A-3}$$

La primera ecuación para definir el filtro consiste en encontrar una forma de computar la estimación *a posteriori* del estado $\widehat{w}_k$ como una combinación lineal de la estimación *a priori*

$\hat{w}_k^-$ y la diferencia con la medición z_k, como se describe en la Ecuación A-4. La estimación de los errores *a priori* y *a posteriori* se definen en las Ecuaciones A-5 y A-6, respectivamente.

$$\hat{w}_k = \hat{w}_k^- + K_k \left(z_k - H\hat{w}_k^- \right) \tag{A-4}$$

$$e_k^- \equiv w_k - \hat{w}_k^- \tag{A-5}$$

$$e_k \equiv w_k - \hat{w}_k \tag{A-6}$$

La matriz $K_k \in \Re^{n \times m}$ se denomina Ganancia de Kalman y se elige de tal forma que minimice la covarianza del error *a posteriori*, tal y como se muestra en la Ecuación A-7, donde P_k^- es la covarianza del error de estimación *a priori* como se muestra en la Ecuación A-8.

$$K_k = P_k^- H^T \left(H P_k^- H^T + R \right)^{-1} \tag{A-7}$$

$$P_k^- = E\left[e_k^- \left(e_k^- \right)^T \right] \tag{A-8}$$

Note que si la matriz de covarianza del error de medición R es cero, entonces la matriz de Ganancia de Kalman es la inversa de la matriz H; mientras que cuando la matriz de covarianza del error de estimación *a priori* es cero, entonces la matriz de Ganancia de Kalman es cero, como se muestra en la Ecuación A-9.

$$\begin{aligned} \lim_{R \to 0} K_k &= H^{-1} \\ \lim_{P_k^- \to 0} K_k &= 0 \end{aligned} \tag{A-9}$$

A.2. Filtro discreto de Kalman

El filtro discreto de Kalman estima el estado de un proceso en algún instante de tiempo y obtiene una retroalimentación en forma de mediciones con ruido. El filtro se compone de dos tipos de ecuaciones que se explican a continuación.

A.2.1. Ecuaciones de actualización de tiempo

Son las encargadas de proyectar hacia adelante el estado actual y la covarianza del error de estimación. Estas ecuaciones son también llamadas **Ecuaciones de Predicción**, y se presentan a continuación:

$$\begin{aligned} \hat{w}_k^- &= A\hat{w}_{k-1}^- + Bu_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned} \tag{A-10}$$

A.2.2. Ecuaciones de actualización de medición

Son responsables de la retroalimentación, la cual es realiza mediante la incorporación de una nueva medición en la estimación *a priori* con el fin de obtener una estimación *a posteriori* mejorada. Estas ecuaciones son también llamadas **Ecuaciones de Corrección**, y se presentan a continuación:

$$\begin{aligned} K_k &= P_k^- H^T \left(HP_k^- H^T + R\right)^{-1} \\ \hat{w}_k &= \hat{w}_k^- + K_k \left(z_k - H\hat{w}_k^-\right) \\ P_k &= \left(I - K_k H\right) P_k^- \end{aligned} \tag{A-11}$$

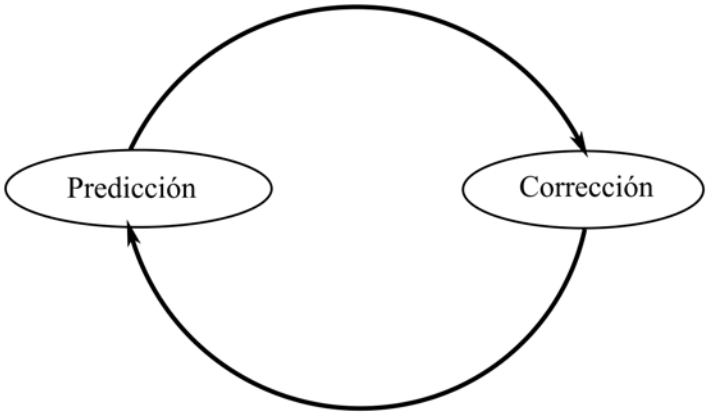


Figura A-1.: Ecuaciones de predicción y corrección del filtro de Kalman

A.2.3. Parámetros del filtro

Los parámetros libres del filtro de Kalman son las matrices de covarianza del error P, Q y R. La matriz de covarianza del error de estimación P generalmente se inicializa como una matriz diagonal y a partir de allí se utilizan las ecuaciones de tiempo para su actualización. Habitualmente, es posible estimar la matriz de covarianza del error de medición R mediante la medición fuera de línea del proceso. La determinación de la covarianza del error del proceso

Q generalmente es más complicada y requiere una etapa previa de identificación del sistema. En general, las covarianzas del error R y Q son decrecientes en el tiempo y se deben ajustar dinámicamente.
En el trabajo de Song y Grizzle [104], se discuten aspectos de la convergencia del filtro de Kalman cuando es usado como observador determinístico de un sistema no lineal en tiempo discreto. Esta convergencia no requiere ningún conocimiento *a priori* del modelo a la hora de asumir los errores de covarianza, siempre y cuando los estados estén dentro de un dominio compacto convexo. Aunque la sintonización de los parámetros P, Q y R es una ardua tarea, generalmente resuelta mediante heurísticas, suele cumplirse la condición de inicialización que se muestra en la Ecuación A-12.

$$P \geq R \geq Q \tag{A-12}$$

A.3. Filtro extendido de Kalman

Algunas de las aplicaciones más interesantes del Filtro de Kalman surgen cuando el proceso a ser estimado $w_k \in \Re^n$ y/o la medición $z_k \in \Re^m$ son no lineales. Un KF que linealice con respecto a la media y a la varianza actual del proceso se denomina Filtro extendido de Kalman (EKF) [118].
En este caso, el proceso es gobernado por la ecuación estocástica no–lineal en diferencias y la medición igualmente no lineal de la Ecuación A-13.

$$\begin{aligned} w_k &= f\left(w_{k-1}, u_{k-1}, \omega_{k-1}\right) \\ z_k &= h\left(w_k, v_k\right) \end{aligned} \tag{A-13}$$

Para facilitar el problema, las variables de ruido ω_k y v_k pueden hacerse cero, por lo que se pueden reescribir las ecuaciones anteriores como se muestra en la Ecuación A-14, donde $\tilde{w}_k$ es una estimación *a posteriori* del estado a partir del estado anterior.

$$\begin{aligned} \tilde{w}_k &= f\left(\hat{w}_{k-1}, u_{k-1}\right) \\ \tilde{z}_k &= h\left(\tilde{w}_k\right) \end{aligned} \tag{A-14}$$

La ecuaciones que linealizan la predicción se muestran en la Ecuación A-15, donde A y Ω son las matrices jacobianas de f con respecto a w y ω, respectivamente; mientras que H y V son las matrices jacobianas de h con respecto a w y v, respectivamente.

$$\begin{aligned} w_k &\approx \tilde{w}_k + A\left(w_{k-1} - \hat{w}_{k-1}\right) + \Omega\omega_{k-1} \\ z_k &\approx \tilde{z}_k + H\left(w_k - \hat{w}_k\right) + Vv_k \end{aligned} \tag{A-15}$$

A.3.1. Ecuaciones del filtro extendido de Kalman

Las Ecuaciones A-16 y A-17 muestran las ecuaciones EKF de actualización de tiempo y actualización de medición, respectivamente.

$$\begin{aligned} \hat{w}_k^- &= f\left(\hat{w}_{k-1}, \hat{u}_{k-1}, 0\right) \\ P_k^- &= A_k P_{k-1} A_k^T + \Omega_k Q_{k-1} \Omega_k^T \end{aligned} \tag{A-16}$$

$$\begin{aligned} K_k &= P_k^- H_k^T \left(H_k P_k^- H_k^T + V_k R_k V_k^T\right)^{-1} \\ \hat{w}_k &= \hat{w}_k^- + K_k \left(z_k - h\left(\hat{w}_k^-, 0\right)\right) \\ P_k &= \left(I - K_k H_k\right) P_k^- \end{aligned} \tag{A-17}$$

A.4. Entrenamiento de redes neuronales con el filtro de Kalman

El uso del Filtro de Kalman como algoritmo de entrenamiento para redes neuronales se basa en el hecho de que los algoritmos de gradiente descendente, mínimos cuadrados recursivos y retropropagación, son casos específicos del Filtro de Kalman [98]. Durante el entrenamiento mediante filtro de Kalman, los pesos sinápticos son los estados a estimar y la salida de la red es la medida utilizada para la corrección. En este sentido, el problema de entrenamiento de una red neuronal puede considerarse como un problema de filtrado óptimo [103].

A.4.1. Entrenamiento MLFFNN

Como una MLFFNN es un sistema no lineal, es necesario emplear el filtro extendido de Kalman (EKF). El entrenamiento de esta red se resuelve como un problema de filtrado óptimo cuya solución es recursiva, utiliza información actual y no necesita almacenar toda la información de la evolución de los pesos sinápticos. El objetivo es encontrar los pesos sinápticos que minimicen el error de predicción [50].
Considere una MLFFNN con n pesos sinápticos y m nodos de salida. Sea w_k el vector de pesos de la red en la k-ésima iteración. La evolución de los pesos sinápticos de la red se indica en la Ecuación A-18, donde χ_k es el vector de entrada de la red neuronal, $\hat{z}_k$ es el vector de salida definido por la función no lineal $h\left(\bullet\right)$.

$$\begin{aligned} w_{k+1} &= w_k \\ \hat{z}_k &= h\left(w_k, \chi_k\right) \end{aligned} \tag{A-18}$$

Las ecuaciones de Kalman necesarias para el entrenamiento de la red se muestran en la

Ecuación A-19, donde z_k es la salida deseada de la red.

$$\begin{aligned} K_k &= P_k H_k^T \left(H_k P_k H_k^T + R_k\right)^{-1} \\ w_{k+1} &= w_k + K_k \left(z_k - h\left(w_k, \chi_k\right)\right) \\ P_{k+1} &= \left(I - K_k H_k\right) P_k + Q_k \end{aligned} \tag{A-19}$$

Considere la MLFFNN mostrado en la Figura **A-2** con p entradas y noc neuronas en la capa oculta. La organización del vector de pesos de la red se muestra en la Ecuación A-20. Adicionalmente, se define la matriz $H_{ij} = \partial\hat{z}/\partial w$ que contiene las derivadas de cada salida de la red con respecto cada peso sináptico, como se muestra en la Ecuación A-21, donde σ_i es la salida de la neurona i en la capa oculta.

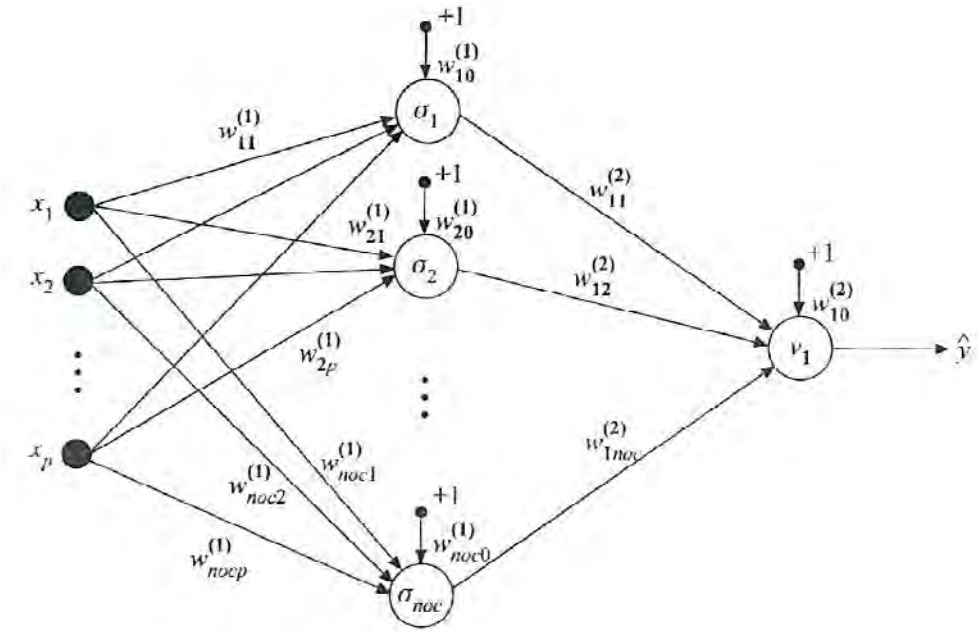


Figura A-2.: MLFFNN entrenado con filtro extendido de Kalman. Tomado de [103]

$$w = \left[\begin{array}{cccccccccccc} w_{10}^{(1)} & w_{11}^{(1)} & \cdots & w_{1p}^{(1)} & w_{20}^{(1)} & w_{21}^{(1)} & \cdots & w_{noc\,p}^{(1)} & w_{10}^{(2)} & w_{11}^{(2)} & \cdots & w_{1noc}^{(2)} \end{array}\right]^T \tag{A-20}$$

$$H = \begin{bmatrix} \gamma\left(n_1\right)\chi_o & \cdots & \gamma\left(n_1\right)\chi_p & \gamma\left(n_2\right)\chi_o & \cdots & \gamma\left(n_{noc}\right)\chi_p & \sigma_0 & \sigma_1 & \cdots & \sigma_{noc} \end{bmatrix}$$

$$\gamma\left(n_i\right) = \frac{w_{1i}^{(2)}\, e^{-n_i}}{\left(1+e^{-n_i}\right)^2}\ ; \quad \forall i = 1, ..., noc \tag{A-21}$$

$$n_i = \sum_{j=0}^{p} w_{ij}^{(1)} \chi_j\ ; \quad \chi_0 = \pm 1$$

B. Apéndice: Procesamiento de imágenes de ultrasonido

Las imágenes de ultrasonido son una herramienta esencial en el diagnóstico médico debido a su naturaleza no invasiva, bajo costo y adquisición en tiempo real. Su uso extendido la ha convertido en una importante área de investigación en los últimos años, siendo el estudio del ruido "speckle" uno de los problemas más analizados. La presencia de este ruido afecta la interpretación humana de las imágenes y la precisión de las técnicas de diagnóstico asistidas por computador [35].

El ruido "speckle" es un ruido típico de los sistemas con iluminación estructurada, tales como los escáneres ultrasónicos [8]. Es un ruido multiplicativo que reduce la resolución y el contraste de la imagen, haciendo difíciles la segmentación y la extracción de características. En la Ecuación B-1 se muestra la imagen $I(x, y)$ contaminada por el ruido $\eta_m(x, y)$.

$$f(x, y) = I(x, y)\ \eta_m(x, y) \tag{B-1}$$

Debido a la compresión logarítmica que utilizan los sistemas de visualización, el ruido "speckle" puede procesarse como si se tratara de ruido aditivo [94], como se muestra en la Ecuación B-2. En la Figura **B-1** se muestran ejemplos de ruido multiplicativo en imágenes de ultrasonido.

$$\log\left[f(x, y)\right] = \log\left[I(x, y)\right] + \log\left[\ \eta_m(x, y)\right] \tag{B-2}$$

En la literatura se reportan métodos de reducción efectiva de este tipo de ruido, entre los que se encuentran análisis estocásticos [83], filtros de difusión anisotrópica [121] y filtros basados en wavelets [63]. La metodología explicada en este apéndice corresponde a la reducción de ruido "speckle" empleando filtrado wavelet.

El objetivo es aplicar una técnica de reducción de ruido "speckle" y realce basada en filtrado wavelet, para lo cual se aplica la metodología presentada en el Algoritmo 3.

(a) Vista lateral del rostro (b) Vista superior del cráneo

(c) Vista de un tumor cerebral (d) Vista de un tumor cerebral

Figura B-1.: Ejemplos de ruido "speckle" en imágenes de ultrasonido

B.1. Caracterización de la señal

B.1.1. Transformada rápida de Fourier (FFT)

En la Figura **B-2** se observa la Transformada Rápida de Fourier de una imagen de ultrasonido de prueba. Nótese que la máxima componente del espectro se ubica en el origen (centro de la imagen), cuya principal componente corresponde a frecuencia cero. Por la distribución del espectro se observa que la imagen es una señal de banda limitada con concentración en bajas frecuencias. Esta información es importante porque permite identificar las componentes frecuenciales de ruido que se quieren reducir.

B.1.2. Extracción de bordes

En la mayoría de los problemas de visión por computador, la extracción de bordes es una de las tareas más importantes, pues permite la ubicación y segmentación de las regiones de interés.

En la Figura **B-2**c se muestran los bordes obtenidos mediante el operador de Sobel [11, 60] a

Algoritmo 3 Reducción de ruido y realce en imágenes de ultrasonido

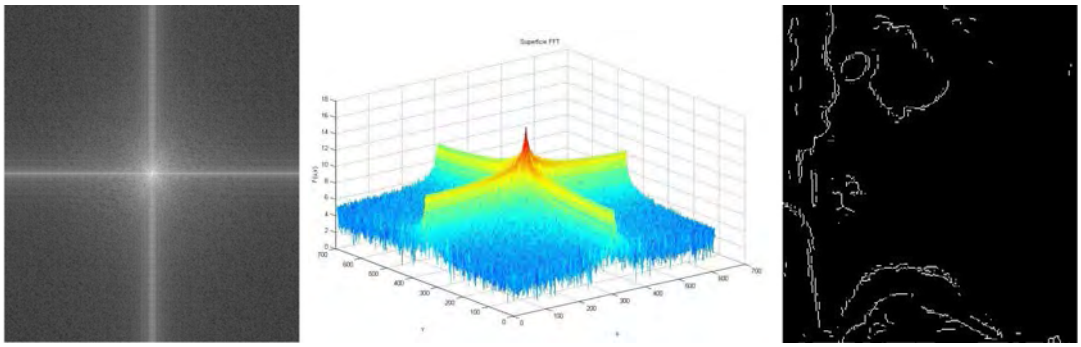
[***Paso A:***] Caracterización de la señal.

1. Análisis Frecuencial (Transformada Rápida de Fourier)
2. Análisis Espacial (Extracción de bordes)
3. Determinación del orden de la señal

[***Paso B:***] Filtrado Wavelet.

1. Selección de la familia Wavelet
2. Reducción de ruido y realce en el dominio Wavelet

una imagen de ultrasonido. Este operador es una de las máscaras de gradiente más empleadas en el procesamiento digital de imágenes, y aunque existen técnicas más elaboradas para la extracción de contornos como las "snakes" [62, 108], sigue siendo muy empleado debido a su simplicidad y velocidad de cómputo.



(a) Imagen FFT (b) Superficie FFT (c) Bordes

Figura B-2.: Transformada rápida de Fourier y extracción de bordes

B.1.3. Orden de la señal

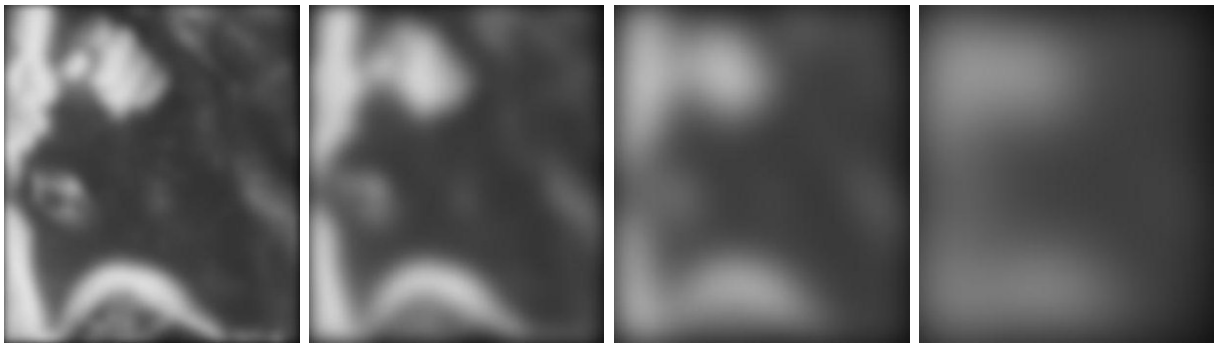
En el caso unidimensional, el orden se calcula ajustando la señal a un polinomio de grado n para el cual el error cuadrático medio de la representación es menor a un umbral ϵ. Para el caso bidimensional (imagen), esta representación polinomial suele ser complicada, por lo que se emplea una aproximación del orden de la señal mediante la aplicación sucesiva de filtros gaussianos, como se explica en el Algoritmo 4.

En la Figura **B-3** se muestran los niveles de detalle obtenidos para cuatro diferentes filtros gaussianos. Debe notarse la pérdida progresiva de detalle desde la Figura **B-3**a hasta la

Algoritmo 4 Determinación del orden para una señal bidimensional

[***Paso 1:***] Definir un kernel gaussiano con varianza σ^2 para filtrar la imagen objetivo [116].
[***Paso 2:***] Aumentar el valor de σ^2 y realizar cada vez la convolución con la imagen objetivo.
[***Paso 3:***] A medida que σ^2 aumenta, la imagen resultante tiene menor nivel de detalle, hasta cierto valor σ^2_{max} para el cual el resultado es una imagen homogénea en la que los detalles son irreconocibles.
[***Paso 4:***] El valor de σ^2_{max} es la estimación del orden de la señal bidimensional.

Figura **B-3**d a medida que la varianza σ^2 del filtro aumenta. Subjetivamente, el orden de la señal se ha estimado en $\sigma^2_{max} \approx 4$, que corresponde al valor de varianza para el cual los detalles importantes son totalmente irreconocibles.



(a) $\sigma^2 = 2.2361$ (b) $\sigma^2 = 3.1623$ (c) $\sigma^2 = 4.4721$ (d) $\sigma^2 = 5.9161$

Figura B-3.: Filtrado gaussiano para hallar el orden de la señal

B.1.4. Transformada wavelet

El filtrado frecuencial en el dominio de Fourier permite la eliminación de ruido en frecuencias específicas. Sin embargo, cuando no se utiliza información espacial, existen zonas de la imagen (como los bordes) que pierden nitidez y contraste.
Gracias a la transformada wavelet, que ofrece tanto información de escala (frecuencia) como de desplazamiento (posición), es posible realizar filtrados selectivos reduciendo ruido en algunas zonas y realzando bordes en otras.
Debido a que las imágenes de ultrasonido contienen bordes de diferentes tamaños, se adopta una aproximación multi–resolución basada en la transformada wavelet. De esta forma, se obtiene información frecuencial y espacial mediante la descomposición de la imagen en versiones escaladas y desplazadas de una función wavelet madre.
La familia wavelet utilizada es una *Daubechies 4*, mostrada en la Figura **B-4**. Esta función es simple computacionalmente y se han reportado buenos resultados en la literatura para

este tipo de aplicaciones [63]. En el algoritmo presentado se utiliza una descomposición nivel 4, que corresponde con el orden estimado de la señal en la Sección B.1.3.

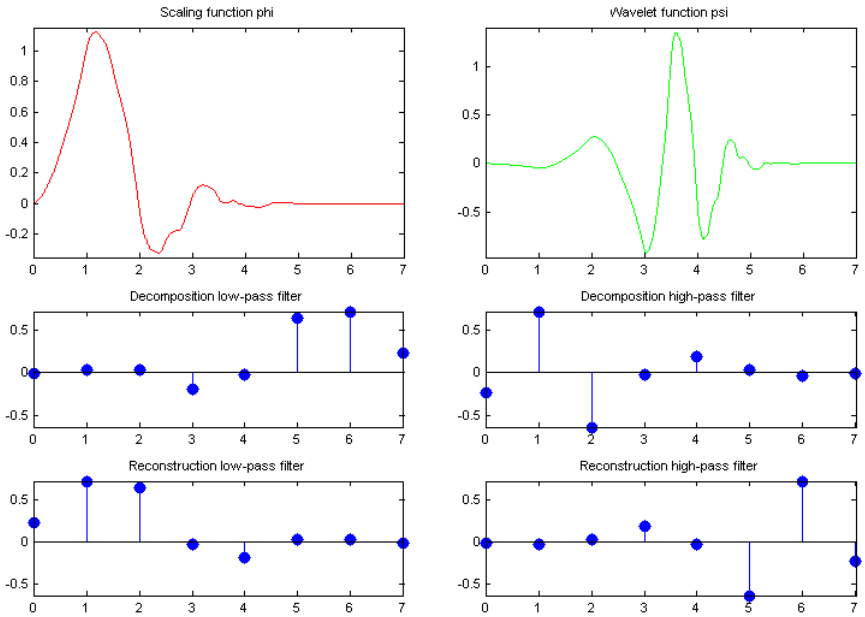


Figura B-4.: Función wavelet *daubechies 4*

B.2. Reducción de ruido y realce

En este trabajo, se utiliza el algoritmo propuesto por Kim y Ra [63] para reducción de ruido "speckle" y realce de bordes, como se muestra en la Figura **B-5**. Este procedimiento se descompone en dos etapas que se explican a continuación.

B.2.1. Etapa 1: extracción de bordes

La descomposición wavelet de una imagen consiste en el cálculo de los coeficientes de aproximación y de detalle, como se muestra en la Figura **B-6**. Los coeficientes de aproximación corresponden a escalas bajas, mientras que los coeficientes de detalle corresponden a escalas altas. Los coeficientes de aproximación son los más inmunes al ruido "speckle", por lo que se realiza extracción de bordes sobre estas componentes para cada uno de los niveles de descomposición. En este caso se tienen cuatro imágenes binarias $E_l(x, y)$ con los bordes extraídos.

B.2.2. Etapa 2: reducción de ruido

En cada uno de los niveles, el ruido "speckle" se puede descomponer de acuerdo a las bandas de escala (frecuencia) de la transformada wavelet, teniendo sus principales componentes en

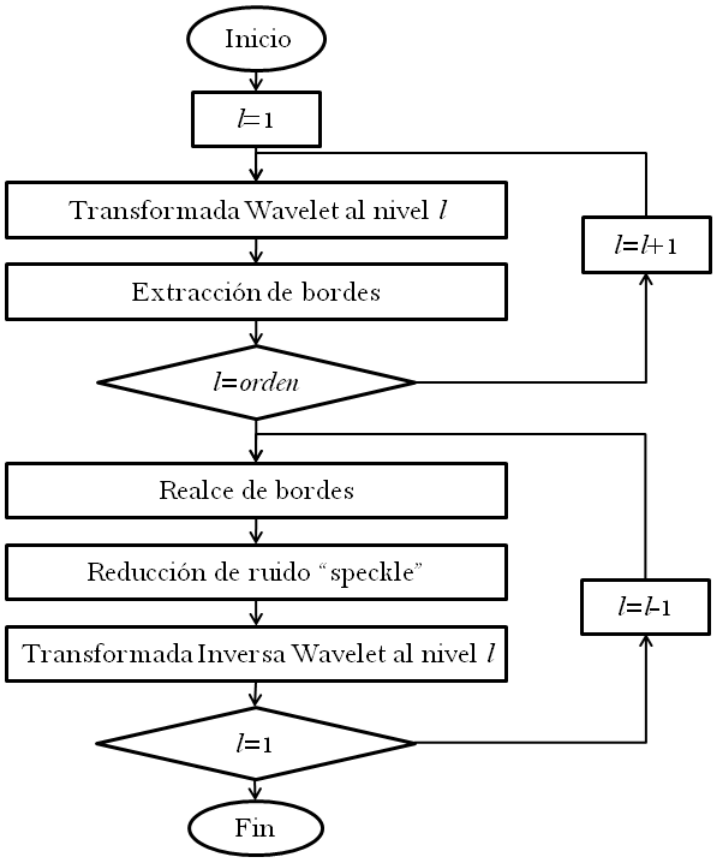


Figura B-5.: Diagrama de flujo de la metodología de procesamiento de imágenes US

las bandas más altas.
Aunque el ruido "speckle" es multiplicativo, en la práctica puede considerarse como ruido aditivo debido a la compresión logarítmica que utilizan los sistemas de visualización ecográficos [94]. En este sentido, la aplicación de un filtro pasa–bajas a los coeficientes de detalle de cada nivel de descomposición, excluyendo los bordes, reduce eficazmente el ruido.
Esta reducción puede realizarse mediante la multiplicación de los coeficientes wavelet por la matriz de reducción R_l de la Ecuación B-3, donde k_l es una constante positiva menor que 1.

$$R_l\left(x,y\right)=\left\{\begin{array}{ll}1\ , & si\ E_l\left(x,y\right)=1\\ k_l, & en\ otro\ caso\end{array}\right. \tag{B-3}$$

B.3. Resultados

En las Figuras **B-8** y **B-7** se muestran las descomposiciones wavelet de dos imágenes de ultrasonido utilizando *Daubechies 4* como función madre. El resultado de la transformada al nivel 1 de descomposición (mostrado en la Figura **B-8**a) consta de 4 secciones: coeficientes de

(a) Coeficientes de aproximación (b) Coeficientes de detalle horizontal

(c) Coeficientes de detalle vertical (d) Coeficientes de detalle diagonal

Figura B-6.: Coeficientes de aproximación y detalle de la descomposición wavelet

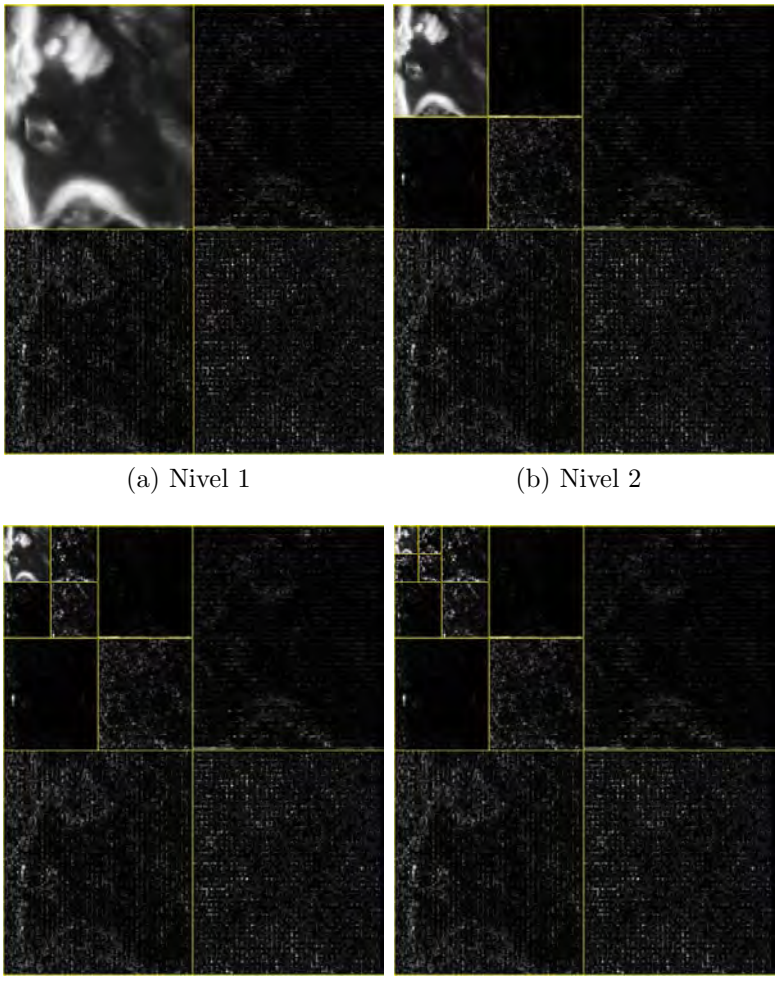
aproximación (izq. arr.), coeficientes de detalle horizontal (der. arr.), coeficientes de detalle vertical (izq. ab.), y coeficientes de detalle diagonal (der. ab.).

Las bandas bajas (izquierda y arriba) de la transformada son más inmunes al ruido, mientras que las bandas altas son más afectadas.

En la Figura **B-8**b se muestra la transformada wavelet de una imagen de ultrasonido radial de prueba utilizando *Daubechies 4* como función madre al nivel 2 de descomposición. La diferencia con respecto a la Figura **B-8**b, es que la subimagen superior izquierda está dividida en cuatro secciones: coeficientes de aproximación (izq. arr.), coeficientes de detalle horizontal (der. arr.), coeficientes de detalle vertical (izq. ab.), y coeficientes de detalle diagonal (der. ab.).

De manera similar, las Figuras **B-8**c y **B-8**d contienen los coeficientes de detalle a los niveles descomposición 3 y 4, respectivamente.

En la Figura **B-9** se muestran los resultados del procesamiento de imágenes de ultrasonido empleando filtrado wavelet. Las Figuras **B-9**a y **B-9**c corresponden a ecografías prenatales de un bebé, y las Figuras **B-9**e y **B-9**g corresponden a ultrasonidos radiales de un tumor cerebral fantasma.



(a) Nivel 1 (b) Nivel 2

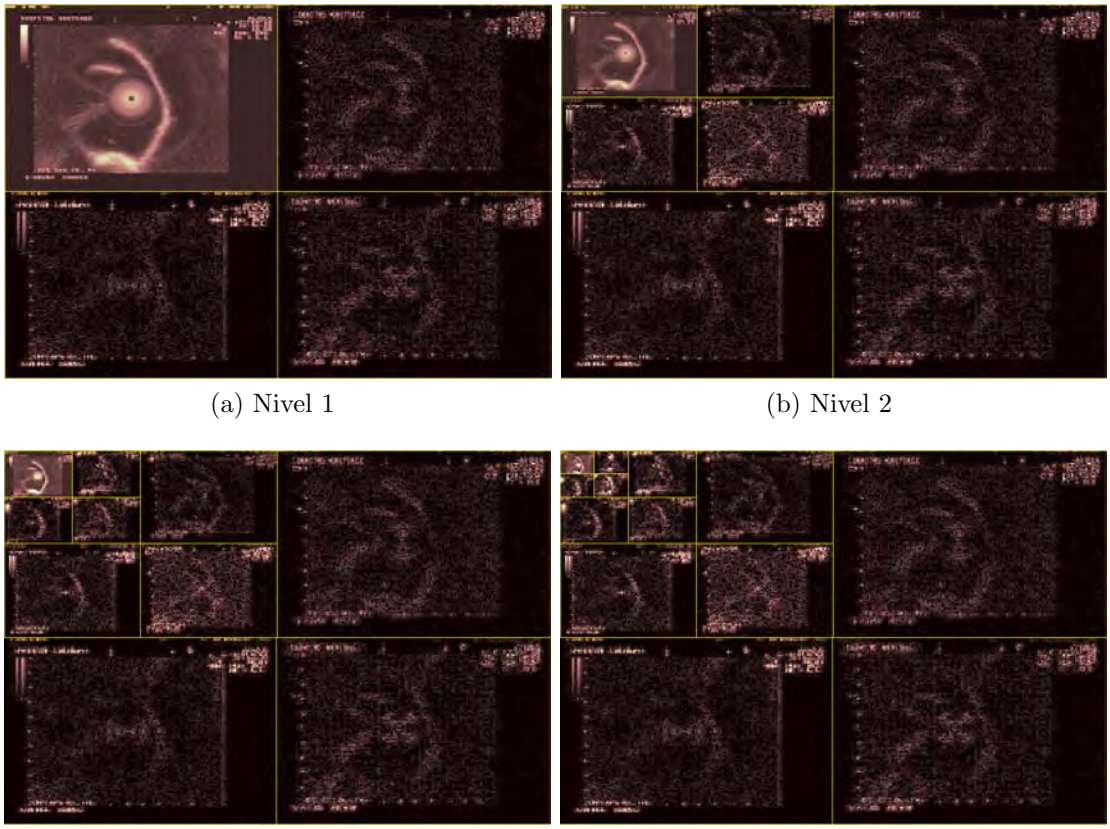
(c) Nivel 3 (d) Nivel 4

Figura B-7.: Ecografía: Descomposición wavelet utilizando *daubechies 4*

En [63], se encuentra una comparación del método aquí implementado contra dos métodos establecidos basados en filtrado no lineal de difusión anisotrópica (NCD - "Nonlinear Coherent Diffusion") [1] y el adelgazamiento wavelet con ajuste de contraste (WSCE - "wavelet shrinkage and contrast enhancement") [126]. El desempeño fue evaluado subjetivamente con la ayuda expertos médicos y expertos en procesamiento digital de imágenes. Los resultados presentados sugieren la superioridad del método basado en filtrado wavelet sobre los otros dos.

B.4. Resumen y conclusiones

Se implementó un algoritmo alternativo al presentado en la Sección 3.5 para el procesamiento de imágenes médicas de ultrasonido. El algoritmo presentado está basado en el trabajo de



(a) Nivel 1 (b) Nivel 2

(c) Nivel 3 (d) Nivel 4

Figura B-8.: Ultrasonido radial: Descomposición wavelet utilizando *daubechies 4*

Kim y Ra [63] y utiliza filtrado wavelet y realce de bordes para llevar a cabo el procesamiento. En las imágenes de ultrasonido, la energía del ruido "speckle" está distribuida en un amplio rango de componentes frecuenciales sobre diferentes posiciones de la imagen, por lo que no es suficiente realizar una reducción de ruido empleando únicamente información frecuencial o espacial, sino que se requieren ambas.

Adicionalmente, se realiza un análisis multi–resolución mediante el uso de cuatro niveles de descomposición de la transformada wavelet, lo que permite la reducción de ruido a diferentes escalas y el realce de bordes de diferente tamaño.

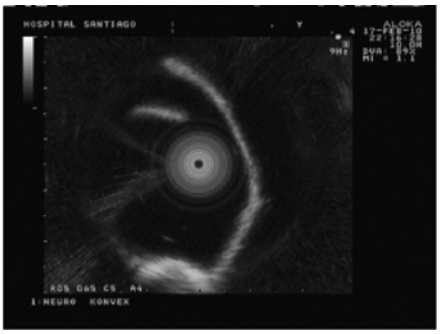
Los resultados experimentales muestran una mejora subjetiva en la calidad visual de la imagen, y de acuerdo a los trabajos reportados en la literatura, ofrece mejor desempeño que otros métodos existentes.

(a) Ecografía prenatal

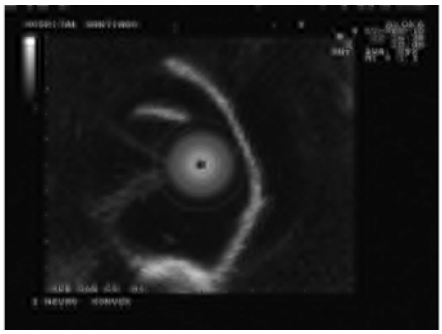
(b) Imagen procesada

(c) Ecografía prenatal

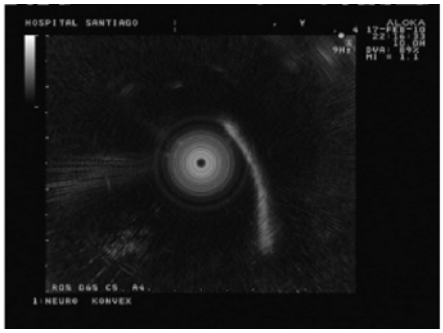
(d) Imagen procesada



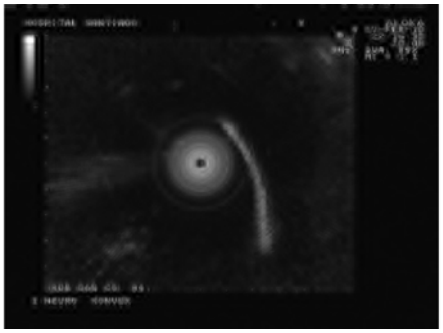
(e) Ultrasonido radial de un tumor



(f) Imagen procesada



(g) Ultrasonido radial de un tumor



(h) Imagen procesada

Figura B-9.: Resultado del procesamiento empleando filtrado wavelet

C. Apéndice: Publicaciones

Parte de esta tesis y otros trabajos realizados durante la maestría han sido publicado como se relaciona a continuación:

- **A. F. Serna–Morales**, F. Prieto, E. Bayro–Corrochano. "Spatio–Temporal Image Tracking Based on Optical Flow and Clustering: An Endoneurosonographic Application". In proceedings of MICAI 2010: Lecture Notes on Artificial Intelligence (LNAI) 6437, pp. 290-300, 2010. G. Sidorov et al. (Eds.). c Springer–Verlag. Berlin Heidelberg 2010. ISBN: 978–3–642–16760–7.

- Alexander Ceballos, **Andrés F. Serna–Morales**, Flavio Prieto, Juan B. Gómez, Tanneguy Redarce. "Sistema Audiovisual para reconocimiento de comandos". En Ingeniare. Revista Chilena de Ingeniería. Vol. 19 N° 2, Mayo–Agosto 2011. ISSN 0718–3291.

- **A. F. Serna–Morales**, F. Prieto, E. Bayro–Corrochano, E. Sánchez. "3D Modeling of Virtualized Reality Objects Using Neural Computing". In IEEE International Joint Conference on Neural Networks, IJCNN 2011. San Jose, California. July 31 - August 5, 2011. **(Aceptado)**

- **A. F. Serna–Morales**, F. Prieto, E. Bayro–Corrochano. "Acquisition of Three–Dimensional Information of Brain Structures Using Endoneurosonography". In International Journal ESWA: Expert Systems With Applications. Elsevier 2011. **(En tercera etapa de revisión)**

- Luis Felipe Montoya–Franco, **A. F. Serna–Morales**, F. Prieto. "3D object modeling with Graphics Hardware Acceleration and Neural Gas Networks". In 5th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications. MIRAGE 2011. October 10–12, 2011. INRIA Rocquencourt, France. **(Sometido a revisión)**

- **A. F. Serna–Morales**, F. Prieto, E. Bayro–Corrochano, E. Sánchez. "Modelado tridimensional de tumores cerebrales empleando endoneurosonografía y redes neuronales auto–organizadas". **(En preparación)**

Bibliografía

[1] Abd-Elmoniem, K.Z. ; Youssef, A. B. M. ; Kadah, Y. M.: Real–time speckle reduction and coherence enhancement in ultrasound imaging via nonlinear anisotropic diffusion. En: *IEEE Transactions on Biomedical Engineering* 49 (2002), Nr. 9, p. 997–1014

[2] ALOKA CO., LTD.: *Ultrasound Diagnostic Equipment ALOKA SSD-5000*, 2002. – MN1–1102 Rev. 9

[3] Ardon, R. ; Cohen, L.D. ; Yezzi, A.: Implicit surface segmentation by minimal paths, applications in 3D medical images. En: *IEEE International Conference on Image Processing. ICIP* Vol. 1, 2005, p. 309–12

[4] Baraff, D. ; Witkin, A.: Dynamic Simulation of Non-penetrating Flexible Bodies. En: *Annual Conference Series in Computer Graphics*, ACM, 1992, p. 303–308

[5] Bartels, R. ; Beatty, J. ; Barsky, B.: *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Los Altos, California, USA : Morgan Kaufmann, 1987

[6] Bayro-Corrochano, E.: Neural Computing for 2D Contour and 3D Surface Reconstruction. En: *Geometric Computing*. Springer London, 2010, p. 443–455

[7] Betancur, J. ; Prieto, F.: Active contour–based segmentation of coffee cherries images. En: 9th *International Conference on Signal Processing, ICSP* (2008), Oktober, p. 2738–2741

[8] Bhuiyan, M. I. H. ; Omair Ahmad, M. ; Swamy, M. N. S.: New Spatially Adaptive Wavelet-based Method for the Despeckling of Medical Ultrasound Images. En: *IEEE International Symposium on Circuits and Systems, ISCAS*, 2007, p. 2347–2350

[9] Bloomenthal, J.: Polygonisation of implicit surfaces. En: *Computer Aided Geometric Design* Vol. 5, 1988, p. 341–355

[10] Bouguet, J. Y.: Camera Calibration Toolbox for MATLAB / Computer Vision Research Group, Department of Electrical Engineering. California Institute of Technology. Mail Code 136–93. Pasadena, CA 91125, Juli 2010. – Informe de Investigación.

Disponible en: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html Ultimo acceso: Octubre 5, 2010

[11] Bovik, Al: *Handbook of Image and Video Processing.* Academic Press (Communications, Networking, and Multimedia). – 3–5 p.

[12] Bradski, Gary ; Kaehler, Adrian: *Learning OpenCV. Computer Vision with the OpenCV Library.* O'Reilly Media, 2008

[13] Breen, D. E. ; Mauch, S. ; Whitaker, R.T.: 3D scan conversion of CSG models into distance volumes. En: *IEEE Symposium on Volume Visualization*, 1998, p. 7–14

[14] Brown, D.C.: Close–Range Camera Calibration. En: *Photogrammetric Engineering* 37 (1971), Nr. 8, p. 855–866

[15] Cameron, S.: Efficient bounds in constructive solid geometry. En: *IEEE Computer Graphics and Applications* 11 (1991), Mai, Nr. 3, p. 68–74

[16] CGAL: *Delaunay Triangulation.* The Computational Geometry Algorithms Library. – Disponible en: http://www.cgal.org Ultimo acceso: Junio 29, 2010

[17] Cohen, F.S. ; Ibrahim, W. ; Pintavirooj, C.: Ordering and parameterizing scattered 3D data for B-spline surface approximation. En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), Juni, Nr. 6, p. 642–648

[18] Collier, J. ; Collier, B. ; O'Toole, G. ; Sargand, S.: Drape prediction by means of finite-element analysis. En: *Journal of the Textile Institute* 82 (1991), Nr. 1, p. 96–107

[19] Cretu, A. M.: 3D Object Modeling: Issues and Techniques / University of Ottawa. 2003. – Preliminary Report

[20] Cretu, A. M. ; Payeur, P. ; Petriu, E.M.: Neural Network Mapping and Clustering of Elastic Behavior From Tactile and Range Imaging for Virtualized Reality Applications. En: *IEEE Transactions on Instrumentation and Measurement* 57 (2008), September, Nr. 9, p. 1918–1928

[21] Cretu, A. M. ; Payeur, P. ; Petriu, E.M.: Selective elasticity data acquisition on 3D deformable objects for virtualized reality applications. En: *IEEE Workshop on Computational Intelligence in Virtual Environments, CIVE*, 2009, p. 14–19

[22] Cretu, A. M. ; Petriu, E. M.: Neural–network–based adaptive sampling of three–dimensional–object surface elastic properties. En: *IEEE Transactions on Instrumentation and Measurement* 55 (2006), apr, Nr. 2, p. 483–492. – ISSN 0018–9456

[23] Cretu, A. M. ; Petriu, E.M. ; Patry, G.G.: Neural–network–based models of 3D objects for virtualized reality: a comparative study. En: *IEEE Transactions on Instrumentation and Measurement* 55 (2006), Februar, Nr. 1, p. 99–111

[24] Cretu, A. M. ; Petriu, E.M. ; Payeur, P.: Data acquisition and modeling of 3D deformable objects using neural networks. En: *IEEE International Conference on Systems, Man and Cybernetics, SMC*. San Antonio, TX, USA, Oktober 2009, p. 3383–3388

[25] Daniilidis, K. ; Bayro-Corrochano, E.: The dual quaternion approach to hand–eye calibration. En: *Proceedings of the* 13th *International Conference on Pattern Recognition* Vol. 1, 1996, p. 318–322

[26] Daniilidis, Konstantinos: Hand–Eye Calibration Using Dual Quaternions. En: *International Journal of Robotics Research* (1999), p. 286–298

[27] Darabi, K. ; Resch, K. ; Weinert, J. ; Jendrysiak, U. ; Perneczky, A.: Real and simulated endoscopy of neurosurgical approaches in an anatomical model. En: Troccaz, Jocelyne (Ed.) ; Grimson, Eric (Ed.) ; Mösges, Ralph (Ed.): *CVRMed-MRCAS* Vol. 1205. 55101 Mainz, Germany : Springer Berlin / Heidelberg, 1997, p. 323–326

[28] Duda, R. O. ; Hart, P. E. ; Stork, D. G.: *Pattern Classification*. Second. New York, USA : John Wiley & Sons, Inc., 2001

[29] Farin, G.: *Curves and Surfaces for Computer Aided Geometric Design*. Second. San Diego : Academic Press, Inc., 1990

[30] Feuerstein, Marco: *The Hand-Eye Calibration Problem*. Munchen, German: Chair for Computer Aided Medical Procedures & Augmented Reality. Technische Universität München, November 2009. – Disponible en: `http://campar.in.tum.de/view/Chair/HandEyeCalibration` Ultimo acceso: Marzo 30, 2011

[31] Figueroa, P. ; Londoño, E. ; Prieto, F. ; Boulanger, P. ; Borda, J. ; Restrepo, D.: Experiencias Virtuales con piezas del Museo del Oro de Colombia / RENATA: Red Nacional Académica de Tecnología Avanzada. 2006. – Informe de Investigación. Disponible en: `http://www.renata.edu.co/index.php/ciencias-sociales/` Ultimo acceso: Diciembre 20, 2010

[32] Foley, S. E. J. ; Dam, A. V. ; Hughes., J.: *Computer Graphics: Principles and Practice*. Second. Reading, MA: Addison - Wesley, 1990

[33] Fritzke, Bernd: A Growing Neural Gas Network Learns Topologies. En: *Advances in Neural Information Processing Systems* Vol. 7, MIT Press, 1995, p. 625–632

[34] Fu, K. S. ; Gonzalez, R. C. ; Lee, C. S. G. ; Sánchez, A. V. (Ed.): *Robótica: Control, detección, visión e Inteligencia.* Madrid : McGraw–Hill, 1988

[35] F.Wagner, Robert ; W.Smith, Stephen ; M.Sandrik, John ; Lopez, Hector: Statistic of Speckle in Ultrasound B–scans. En: *IEEE Transactions on Sonics and Ultrasonics* 30 (1983), Mai, Nr. 3, p. 156–163

[36] Galic, S. ; Loncaric, S.: Spatio-temporal image segmentation using optical flow and clustering algorithm. En: *Proceedings of the* 1st *International Workshop on Image and Signal Processing and Analysis, IWISPA*, 2000, p. 63–68

[37] Gibson, S. F. F. ; Mirtich, B.: A survey of deformable modeling in computer graphics / MERL–Mitsubishi Electric Research Laboratory. 1997. – Informe de Investigación

[38] Gillams, A.: 3D imaging - A clinical perspective. En: *IEE Colloquium on 3D Imaging Techniques for Medicine*, 1991, p. 111–112

[39] Gobbi, D.G. ; Comeau, R. M. ; Lee, B. K. H. ; Peters, T. M.: Integration of intra-operative 3D ultrasound with pre-operative MRI for neurosurgical guidance. En: *Proceedings of the* 22nd *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* Vol. 3, 2000, p. 1738–1740

[40] Goldfeather, J. ; Monar, S. ; Turk, G. ; Fuchs, H.: Near real–time CSG rendering using tree normalization and geometric pruning. En: *IEEE Computer Graphics and Applications* 9 (1989), Mai, Nr. 3, p. 20–28

[41] Gono, K.: Multifunctional Endoscopic Imaging System for Support of Early Cancer Diagnosis. En: *IEEE Journal of Selected Topics in Quantum Electronics* 14 (2008), Januar, Nr. 1, p. 62–69

[42] Gonzalez, R. C. ; Woods, R. E. ; Eddins, S. L.: *Digital Image Processing using MATLAB.* 2. Gatesmark Publishing, 2009

[43] Gourret, J.P. ; Magnenat-Thalmann, N. ; Thalmann, D.: Simulation of object and human skin deformations in a grasping task. En: *Proceedings of Annual Conference Series In Computer Graphics*, ACM SIGGRAPH, 1989, p. 21–30

[44] Guthkelch, A. N. ; Carter, L. P. ; Cassady, J. R. [u. a.]: Treatment of malignant brain tumors with focused ultrasound hyperthermia and radiation: Results of a phase 1 trial. En: *Neurooncol* 10 (1991), p. 271–284

[45] Hagan, M.T. ; Menhaj, M.B.: Training feedforward networks with the Marquardt algorithm. En: *IEEE Transactions on Neural Networks* 5 (1994), November, Nr. 6, p. 989–993

[46] Haibo, G. ; Wenxue, H. ; Jianxin, C. ; Yonghong, X.: Optimization of Principal Component Analysis in Feature Extraction. En: *International Conference on Mechatronics and Automation, ICMA*, 2007, p. 3128–3132

[47] Hall, R.: Supporting Complexity and Conceptual Design in Modeling Tools. En: Rogers, D.F. (Ed.) ; Earnshaw, R.A. (Ed.): *State of the Art in Computer Graphics: Visualization and Modeling*, SIGGRAPH, 1991

[48] Hayashida, Takeo: Present status of gastro camera. En: *Journal of Gastroenterology* 1 (1966), Nr. 2, p. 6–7

[49] Haykin, Simon: *Neural Networks - A Comprehensive Foundation*. 2. Communications Research Laboratory, McMaster University, Hamilton, Ontario, Canada : Prentice Hall, Juli 1998

[50] Haykin, Simon: *Kalman Filtering and Neural Networks*. Communications Research Laboratory, McMaster University, Hamilton, Ontario, Canada : John Wiley & Sons, Inc., 2001

[51] He, Ying ; Qin, Hong: Surface reconstruction with triangular B–splines. En: *Geometric Modeling and Processing*, 2004, p. 279–287

[52] Heckscher, Moritz ; Reeff, Mireille ; Wengert, Christian: Endoscope Calibration Setup / Computer Vision Lab, ETH Zuerich. 2005. – Informe de Investigación

[53] Heikkila, J. ; Silven, O.: A four-step camera calibration procedure with implicit image correction. En: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, p. 1106–1112

[54] Hill, Derek ; Maurer, Calvin ; Wang, Matthew ; Maciunas, Robert ; Barwise, John ; Fitzpatrick, J.: Estimation of intraoperative brain surface movement. En: *CVRMed–MRCAS'97* Vol. 1205. Springer Berlin/Heidelberg, 1997, p. 449–458

[55] Hoberock, Jared ; Bell, Nathan: *Thrust: A Parallel Template Library*. 2010. – Disponible en: `http://www.meganewtons.com/` Ultimo acceso: Marzo 30, 2011

[56] Horn, B. K. P. ; Schunck, B. G.: Determining Optical Flow. En: *Artificial Intelligence* 17 (1981), p. 185–203

[57] Hynynen, K. ; Clement, G.: Clinical applications of focused ultrasound: The brain. En: *International Journal on Hyperthermia* 23 (2007), Nr. 2, p. 193–202

[58] Iseki, Kazushige ; Tatsuta, Masaharu ; Iishi, Hiroyasu ; Sakai, Noriko ; Yano, Hiroyuki ; Ishiguro, Shingo: Effectiveness of the near–infrared electronic endoscope for diagnosis of the depth of involvement of gastric cancers. En: *Gastrointestinal Endoscopy* 52 (2000), Nr. 6, p. 755–762

[59] Jaramillo, A. E. ; Boulanger, P. ; Prieto, F.: On–line 3–D inspection of deformable parts using FEM trained radial basis functions. En: *IEEE* 12th *International Conference on Computer Vision Workshops, ICCV.*, 2009, p. 1733–1739

[60] Jähne, B.: *Digital Image Processing.* 5. Berlin Heidelberg New York : Springer–Verlag, 2002

[61] Kalman, Rudolph E.: A New Approach to Linear Filtering and Prediction Problems. En: *Transactions of the ASME–Journal of Basic Engineering* 82 (1960), Nr. Series D, p. 35–45

[62] Kass, M. ; Witkin, A. ; Terzopoulos, D.: Snakes: Active contour models. En: *International Journal of Computer Vision* 1 (1988), Nr. 4, p. 321–331

[63] Kim, Yong S. ; Ra, Jong B.: Improvement of ultrasound image based on wavelet transform: speckle reduction and edge enhancement. En: *SPIE Medical Imaging*, 2005, p. 1085–1092

[64] Kohonen, T. ; Nieminen, I. ; Honkela, T.: On the Quantization Error in SOM vs. VQ: A Critical and Systematic Study. En: Príncipe, J. (Ed.) ; Miikkulainen, R. (Ed.): *Advances in Self–Organizing Maps* Vol. 5629. Springer Berlin/Heidelberg, German, 2009, p. 133–144

[65] Konica Minolta Sensing, Inc.: *Non-contact 3D Digitizer Vivid 9i Instruction Manual.* 2004

[66] Lefèvre, S. ; Fluck, C. ; Maillard, B. ; Vincent, N.: A Fast Snake–based Method to Track Football Players. En: *International Workshop on Machine Vision Applications, IAPR.* Tokyo, Japan : RFAI publication, November 2000, p. 501–504

[67] Letteboer, M.M.J. ; Willems, P.W.A. ; Viergever, M.A. ; Niessen, W.J.: Brain shift estimation in image–guided neurosurgery using 3–D ultrasound. En: *IEEE Transactions on Biomedical Engineering* 52 (2005), Februar, Nr. 2, p. 268–276

[68] Levenberg, K.: A method for the solution of certain problems in least squares. En: *Quart. Applied Math.* 2 (1944), p. 164–168

[69] Li, Bing ; Acton, S.T.: Active Contour External Force Using Vector Field Convolution for Image Segmentation. En: *IEEE Transactions on Image Processing* 16 (2007), August, Nr. 8, p. 2096–2106

[70] Lin, Chin-Feng ; Yang, Don-Lin ; Chung, Yeh-Ching: A marching voxels method for surface rendering of volume data. En: *Computer Graphics International*, 2001, p. 306–313

[71] Linh, T.K.: Reconstruction of Euclidean planes from voxels. En: 2nd *International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings.*, 2004, p. 781–788

[72] Liu, H. ; Wang, X. ; Qiang, W.: Implicit surface reconstruction from 3D scattered points based on variational level set method. En: 2nd *International Symposium on Systems and Control in Aerospace and Astronautics. ISSCAA.*, 2008, p. 1–5

[73] min Liu, Xu ; kuan Huang, Hou ; xiang Xu, Wei ; Chen, Jing: Research on the reconstruction method of B-spline surface based on radius basis function neural networks. En: *IEEE Conference on Cybernetics and Intelligent Systems* Vol. 2, 2004, p. 1123–1127

[74] Luong, Q. T. ; Faugeras, O. D.: Self-Calibration of a Moving Camera from Point Correspondences and Fundamental Matrices. En: *International Journal of Computer Vision* 22 (1997), Nr. 3, p. 261–289. – ISSN 0920–5691

[75] Luong, ThÃ© V. ; Loukil, L. ; Melab, N. ; Talbi, E.-G.: A GPU-based iterated tabu search for solving the quadratic 3-dimensional assignment problem. En: *Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on*, 2010, p. 1–8

[76] Machucho-Cadena, R. ; de la Cruz-Rodríguez, S. ; Bayro-Corrochano, E.: Rendering of brain tumors using endoneurosonography. En: 19th *International Conference on Pattern Recognition, ICPR*, 2008, p. 1–4

[77] Machucho-Cadena, Ruben ; de la Cruz-Rodríguez, Sergio ; Bayro-Corrochano, Eduardo: Joint Freehand Ultrasound and Endoscopic Reconstruction of Brain Tumors. En: *Progress in Pattern Recognition, Image Analysis and Applications* Vol. 5197. Springer Berlin/Heidelberg, 2008, p. 691–698

[78] Marquardt, D.: An algorithm for least squares estimation of non-linear parameters. En: *Soc. Ind. Appl. Math* (1963), p. 431–441

[79] Martinetz, T.M. ; Berkovich, S.G. ; Schulten, K.J.: 'Neural-gas' network for vector quantization and its application to time-series prediction. En: *IEEE Transactions on Neural Networks* 4 (1993), Juli, Nr. 4, p. 558–569

[80] Maurer, Calvin ; Hill, Derek ; Maciunas, Robert ; Barwise, John ; Fitzpatrick, J. ; Wang, Matthew: Measurement of intraoperative brain surface deformation under a craniotomy. En: Wells, William (Ed.) ; Colchester, Alan (Ed.) ; Delp, Scott (Ed.): *Medical Image Computing and Computer-Assisted Interventation, MICCAI98* Vol. 1496. Springer Berlin / Heidelberg, 1998, p. 51–62

[81] Maurer, Jr. ; Hill, D.L.G. ; Martin, A.J. ; Liu, Haiying ; McCue, M. ; Rueckert, D. ; Lloret, D. ; Hall, W.A. ; Maxwell, R.E. ; Hawkes, D.J. ; Truwit, C.L.: Investigation of intraoperative brain deformation using a 1.5–T interventional MR system: preliminary results. En: *IEEE Transactions on Medical Imaging* 17 (1998), Oktober, Nr. 5, p. 817–825

[82] Merrill, Duane ; Grimshaw, Andrew: Revisiting Sorting for GPGPU Stream Architectures / University of Virginia, Department of Computer Science. Charlottesville, VA, USA, 2010 (CS2010-03). – Informe de Investigación

[83] Michailovich, O. ; Tannenbaum, A.: Despeckling of medical ultrasound images. En: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 53 (2006), p. 64–78

[84] Microsoft, Corporation: *Xbox 360 – Creada para la diversión.* 2010. – Disponible en: `http://www.xbox.com/es-MX/kinect` Ultimo acceso: Abril 13, 2011

[85] Montoya-Franco, Luis F. ; Serna-Morales, A. F. ; Prieto., Flavio: 3D object modeling with Graphics Hardware Acceleration and Neural Gas Networks / Grupo de Automática de la Universidad Nacional de Colombia. Bogotá, Colombia, April 2011. – Informe de Investigación

[86] Na, S. ; Xumin, L. ; Yong, G.: Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm. En: 3rd *International Symposium on Intelligent Information Technology and Security Informatics, IITSI*, 2010, p. 63–67

[87] NDI: *Polaris Vicra Measurement Volume.* Northern Digital Inc, Measurement Solutions in Medical, 2010. – Disponible en: `http://www.ndigital.com/medical/polarisfamily-volumes-vicra.php`. Ultimo acceso: Diciembre 12, 2010

[88] Nummiaro, K. ; Koller-Meier, E. ; Gool, L. V.: An Adaptive Color-Based Particle Filter. En: *Image Vision Computing* 21 (2003), Nr. 1, p. 99–110

[89] Ortegon-Aguilar, J. ; Bayro-Corrochano, E.: Omnidirectional Vision Tracking with Particle Filter. En: 18th *International Conference on Pattern Recognition, ICPR 2006* Vol. 3, 2006, p. 1115–1118

[90] Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. En: *IEEE Transactions on Systems, Man and Cybernetics* 9 (1979), Nr. 1, p. 62–66

[91] Patil, S. ; Ravi, B.: Voxel–based representation, display and thickness analysis of intricate shapes, 2005, p. 6

[92] Piastra, M.: A growing self-organizing network for reconstructing curves and surfaces. En: *International Joint Conference on Neural Networks, IJCNN*, 2009, p. 2533–2540

[93] Piperakis, E. ; Kumazawa, I.: Affine transformations of 3D objects represented with neural networks Department of Computer Science, Tokyo Institute of Technology

[94] Quistgaard, J. U.: Signal Acquisition and Processing in Medical Diagnostic Ultrasound. En: *IEEE Signal Processing Magazine* 14 (1997), Januar, p. 67–74

[95] Resch, Klaus Dieter M.: *Transendoscopic Ultrasound for Neurosurgery.* Greifswald Germany : Springer Berlin Heidelberg, 2006 (Medicine)

[96] Rivera-Rovelo, J. ; Bayro-Corrochano, E.: Surface Approximation and Morphing using Self-Organized Networks Improved with Gradient Information in the Geometric Algebra Framework / CINVESTAV, Unidad Guadalajara. 2009. – Informe de Investigación

[97] Roberts, D. W. ; Hartov, A. ; Kennedy, F. E. ; Hartov, E. ; Miga, M. I. ; Paulsen, K. D. *Intra–operative Brain Shift and Deformation: A Quantitative Analysis of Cortical Displacement in 28 Cases.* 1998

[98] Ruck, D.W. ; Rogers, S.K. ; Kabrisky, M. ; Maybeck, P.S. ; Oxley, M.E.: Comparative analysis of backpropagation and the extended Kalman filter for training multilayer perceptrons. En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), Juni, Nr. 6, p. 686–691

[99] Schmidt, Jochen ; Vogt, Florian ; Niemann, Heinrich: Calibration Free Hand–Eye Calibration: A Structure from Motion Approach. En: Kropatsch, Walter G. (Ed.) ; Sablatnig, Robert (Ed.) ; Hanbury, Allan (Ed.): *Pattern Recognition* Vol. 3663. Springer Berlin/Heidelberg, 2005, p. 67–74

[100] Seber, G. A. F.: *Multivariate Observations.* Hoboken, NJ : John Wiley & Sons, Inc., 1984

[101] Serna-Morales, A. F. ; Prieto, F. ; Bayro-Corrochano, E.: Spatio–temporal Image Tracking Based on Optical Flow and Clustering: An Endoneurosonographic Application. En: Sidorov, G. (Ed.) ; Hernández-Aguirre, A. (Ed.) ; Reyes-García, C. (Ed.): *Advances in Artificial Intelligence* Vol. 6437. Springer Berlin/Heidelberg, German, Oktober 2010, p. 290–300

[102] Shiu, Y.C. ; Ahmad, S.: Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB. En: *IEEE Transactions on Robotics and Automation* 5 (1989), Februar, Nr. 1, p. 16–29

[103] Sánchez-Camperos, E. N. ; Alanís-García, A. Y.: *Redes Neuronales: Conceptos fundamentales y aplicaciones a control automático.* CINVESTAV, Unidad Guadalajara : Pearson Prentice Hall, 2006

[104] Song, Yongkyu ; Grizzle, Jessy W.: The Extended Kalman Filter as a Local Asymptotic Observer for Nonlinear Discrete-Time Systems. En: *American Control Conference, AMC*, 1992, p. 3365–3369

[105] Steinbach, E. ; Girod, B. ; Eisert, P. ; Betz, A.: 3D reconstruction of real–world objects using extended voxels. En: *International Conference on Image Processing* Vol. 1, 2000, p. 569–572

[106] Takehana, S. ; Kaneko, M. ; Mizuno, H.: Endoscopic Diagnostic System Using Autofluorescence. En: *Diagnostic and Therapeutic Endoscopy* 5 (1999), Nr. 2, p. 59–63

[107] Tatar, F. ; Mollinger, J. R. ; Dulk, R. C. D. ; van Duyl, W. A. ; Goosen, J. F. L. ; Bossche, A.: Ultrasonic sensor system for measuring position and orientation of laparoscopic instruments in minimal invasive surgery. En: 2nd *Annual International IEEE–EMB Special Topic Conference on Microtechnologies in Medicine Biology*, 2002, p. 301–304

[108] Tauber, C. ; Batatia, H. ; Morin, G. ; Ayache, A.: Robust B-spline snakes for ultrasound image segmentation. En: *Computers in Cardiology.* Toulouse, France, 2004

[109] Tiller, W.: Rational B–Splines for Curve and Surface Representation. En: *IEEE Computer Graphics and Applications* 3 (1983), September, Nr. 6, p. 61–69

[110] Tsai, R.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off–the–shelf TV cameras and lenses. En: *IEEE Journal of Robotics and Automation* 3 (1987), August, Nr. 4, p. 323–344

[111] Tsai, R.Y. ; Lenz, R.K.: Real time versatile robotics hand/eye calibration using 3D machine vision. En: *IEEE International Conference on Robotics and Automation* Vol. 1, 1988, p. 554–561

[112] Tsai, R.Y. ; Lenz, R.K.: A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. En: *IEEE Transactions on Robotics and Automation* 5 (1989), Juni, Nr. 3, p. 345–358

[113] Varshney, S. S. ; Rajpal, N. ; Purwar, R.: Comparative study of image segmentation techniques and object matching using segmentation. En: *International Conference on Methods and Models in Computer Science, ICM2CS*, 2009, p. 1–6

[114] Vesanto, J. ; Himberg, J. ; Alhoniemi, E. ; Parhankangas, J.: *SOM Toolbox for MATLAB.* P. O. Box 5400, FIN–02015 HUT, Finland: Helsinki University of Technology, April 2000. – Disponible en: `http://www.cis.hut.fi/projects/somtoolbox/` Ultimo acceso: Octubre 5, 2010

[115] Wei, G. Q. ; Ma, S. D.: A complete two–plane camera calibration method and experimental comparisons. En: *Proceedings of the* 4th *International Conference on Computer Vision*, 1993, p. 439–446

[116] Weickert, J. ; Ishikawa, S. ; Imiya, A.: Gaussian Scale–Space Theory. En: *On the history of Gaussian scale-space axiomatics*. 1997, p. 45–58

[117] Weisstein, E. W.: *Plane*. Wolfram Research, Inc, 2010. – Disponible en: `http://mathworld.wolfram.com/Plane.html` Ultimo acceso: Septiembre 12, 2010

[118] Welch, Greg ; Bishop, Gary: An Introduction to the Kalman Filter / University of North Carolina at Chapel Hill, Department of Computer Science. Chapel Hill, NC 27599–3175, Juli 2006. – TR 95–041. Disponible en: `http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf` Ultimo acceso: Febrero 22, 2010

[119] Wengert, C. ; Reeff, M. ; Cattin, C. ; Székely, G.: Fully Automatic Endoscope Calibration for Intraoperative Use. En: *Bildverarbeitung für die Medizin*, Springer–Verlag, März 2006, p. 419–23

[120] Yang, Fang-Chung ; Kuo, Chung-Hsien ; Wing, Jein-Jong ; Yang, Ching-Kun: Reconstructing the 3D solder paste surface model using image processing and artificial neural network. En: *IEEE International Conference on Systems, Man and Cybernetics* Vol. 3, 2004, p. 3051–3056

[121] Yu, Yongjian ; Acton, Scott T.: Speckle Reducing Anisotropic Difusion. En: *IEEE Transactions on Medical Imaging* Vol. 11, 2002, p. 1261–1270

[122] Zhang, Qian ; Jia, Jinyuan ; Li, Hongyu: A GPU Based 3D Object Retrieval Approach Using Spatial Shape Information. En: *IEEE International Symposium on Multimedia, ISM.*, 2010, p. 212–219

[123] Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. En: *Proceedings of the* 7th *IEEE International Conference on Computer Vision* Vol. 1, 1999, p. 666–673

[124] Zhu, Weihang ; Curry, J.: Parallel ant colony for nonlinear function optimization with graphics hardware acceleration. En: *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 2009, p. 1803–1808

[125] Zhu, Weihang ; Curry, J.: Particle Swarm with graphics hardware acceleration and local pattern search on bound constrained problems. En: *IEEE Swarm Intelligence Symposium, SIS*, 2009, p. 1–8

[126] Zong, Xuli ; Laine, A.F. ; Geiser, E.A.: Speckle reduction and contrast enhancement of echocardiograms via multiscale nonlinear processing. En: *IEEE Transactions on Medical Imaging* 17 (1998), August, Nr. 4, p. 532–540