

**DESARROLLO DE UN ENTORNO VIRTUAL PARA LA SIMULACIÓN DE
INTERVENCIONES QUIRÚRGICAS EN NEUROCIRUGIA**

Trabajo de investigación

Maestría en Ingeniería

Área: Electrónica

Línea: Procesamiento de imágenes médicas

Grupo de Investigación en Bioinstrumentación e Ingeniería Clínica: GIBIC.

Presentado por:

Sara Isabel Duque Vallejo

sara.duquevallejo@gmail.com

Tutor:

John Fredy Ochoa Gómez

john.ochoa@udea.edu.co



**UNIVERSIDAD
DE ANTIOQUIA**

2016

RESUMEN

Los sistemas de neurocirugía guiada por imágenes han surgido como una herramienta útil en la formación médica para enfrentar las deficiencias de los métodos tradicionales de enseñanza. El hecho de que cada año muchas personas mueran en los hospitales por errores en los procedimientos quirúrgicos ayuda a entender el posible impacto del entrenamiento médico y revela que la insuficiencia de éste podría costar vidas humanas [1].

Aunque en el medio existen diferentes tipos de entornos desarrollados para la aplicación de sistemas de neurocirugía guiada por imágenes, con una alta calidad en la visualización y la interacción con los modelos, en general estos sistemas presentan un costo muy elevado, por lo que su aplicación con fines educativos se ve disminuida, además, tampoco ofrecen todas las opciones de visualización existentes ya que las opciones más avanzadas requieren conocimientos mayores de procesamiento de imágenes. Por esta razón se desarrolló un entorno para el acceso a operaciones avanzadas de procesamiento de imágenes y la manipulación intuitiva de imágenes volumétricas, el cual puede manipularse por medio de un periférico de entrada diferente al mouse, mejorando y facilitando la forma de interacción usuario-sistema. El entorno fue desarrollado en 3D Slicer, que es un software de uso libre, y facilitará el acceso de los neurocirujanos en formación de la ciudad, a las técnicas de craneotomía virtual y manipulación de representaciones tridimensionales.

Palabras clave: Procesamiento de imágenes médicas, Simulación quirúrgica, Sistema de Neurocirugía Guiado por Imágenes.

ABSTRACT

Image-guided neurosurgery systems has emerged as a useful tool in medical training to address the shortcomings of traditional methods of teaching, because many people die each year in hospitals as result of surgical procedures errors. This information helps to understand the possible impact of medical training and reveals that its failure could cost human lives [1].

Although there are different types of developed environments for the implementation of image guided neurosurgery systems that have high visualization quality and image interaction, they generally have a high cost, so their application in education may be diminished, also, these systems do not offer all existing visualization options because the more advanced options require greater knowledge of image processing. For this reason the development of an environment with the appropriate characteristics is proposed, which could be manipulated by an input device instead of the mouse, allowing more intuitive interaction. The environment will be developed in 3D Slicer that is free software and will provide access to training neurosurgeons to virtual craniotomy techniques.

Keywords: IGNS, medical imaging, Surgical Simulation

CONTENIDO

INTRODUCCIÓN	8
1 PRELIMINARES	10
1.1 PLANTEAMIENTO DEL PROBLEMA	10
1.2 OBJETIVOS DEL TRABAJO DE INVESTIGACIÓN	12
1.2.1 General.....	12
1.2.2 Específicos	12
1.3 PUBLICACIONES DERIVADAS.....	12
1.4 ESTADO DEL ARTE.....	13
1.4.1 Sistemas de Neurocirugía Guiada por Imágenes.....	14
1.4.2 Estado del Arte en Colombia	16
1.4.3 Limitaciones.....	19
2 VISUALIZACIÓN PARA LA REPRESENTACIÓN TRIDIMENSIONAL DE LOS DATOS DE IMÁGENES MÉDICAS.....	20
2.1 IMÁGENES MÉDICAS.....	20
2.1.1 Selección de imágenes.....	20
2.2 TÉCNICAS UTILIZADAS EN EL PROCESAMIENTO DIGITAL DE IMÁGENES.....	21
2.2.1 Eliminación de ruido.	21
2.2.2 Segmentación.....	22
2.2.3 Registro	24
2.2.4 Reconstrucción de volúmenes.....	25
2.3 TÉCNICAS DE VISUALIZACIÓN	26
2.3.1 Visualización 3D individual.	26
2.3.2 Visualización basada en multiventanas gráficas.	27
2.3.3 Visualización en línea.	27
2.3.4 Visualización en línea aumentada.	27
2.4 SOFTWARE UTILIZADO Y LIBRERIAS	27
2.4.1 3D Slicer.....	27
2.4.2 Python	28
2.4.3 Visualization Tool Kit (VTK)	28
2.4.4 Qt	28
2.4.5 FSL.....	28
2.5 IMPLEMENTACION DE UN ALGORITMO DE PARA LA REPRESENTACIÓN TRIDIMENSIONAL DE LOS DATOS.....	29

2.5.1	Recolección de las imágenes	29
2.5.2	Segmentación.....	30
2.5.3	Reconstrucción tridimensional de las estructuras	33
2.6	DISCUSION	37
3	SIMULACIÓN DE CRANEOTOMÍAS EN REPRESENTACIONES TRIDIMENSIONALES Y LA SIMULACIÓN DE LAS VISTAS QUIRÚRGICAS QUE SE OBTIENEN LUEGO DE LA CRANEOTOMÍA VIRTUAL.	38
3.1	CLIPPING	38
3.2	RESECCIÓN VIRTUAL.....	39
3.3	CORTE DE DATOS VOLUMÉTRICOS	40
3.4	CRANEOTOMIA	41
3.5	ALGORITMO DESARROLLADO PARA LA SIMULACIÓN DE CRANEOTOMIAS.....	43
3.6	DISCUSIÓN	47
4	IMPLEMENTACION DE UN PROTOCOLO PARA LA MANIPULACIÓN DEL ENTORNO POR MEDIO DE UN PERIFÉRICO DE ENTRADA QUE PERMITE RELACIONAR COORDENADAS EN EL ESPACIO FÍSICO CON EL ESPACIO VIRTUAL.....	48
4.1	PERIFÉRICOS DE ENTRADA UTILIZADOS EN SIMULADORES DE CIRUGÍA.....	48
4.2	PERIFÉRICO DE ENTRADA ELEGIDO.....	49
4.3	REDISEÑO DEL PERIFÉRICO DE ENTRADA	50
4.4	DISCUSION	56
5	IMPLEMENTACIÓN DE LOS ALGORITMOS DESARROLLADOS EN UNA APLICACIÓN DE USO LIBRE.	58
5.1	DESCRIPCIÓN DEL SIMULADOR DESARROLLADO	58
5.2	DISCUSION	66
	CONCLUSIONES Y TRABAJO FUTURO	67
	BIBLIOGRAFIA	69
	ANEXOS.....	74
	ANEXO 1: CÓDIGO FUENTE CRANEOTOMÍA VIRTUAL	74
	ANEXO 2: CÓDIGO FUENTE PARA LA RECEPCIÓN DE DATOS VIA SERIAL	77

TABLA DE FIGURAS

Figura 1. Proyección AP y lateral en la cual se pueden observar los electrodos subdurales. Dado que la imagen de rayos X es una proyección, no es posible conocer exactamente la relación espacial de los electrodos	17
Figura 2. Representación 3D final de la invasión con 2 rejillas de 32 electrodos y una tirilla de 8 electrodos	17
Figura 3. Visualización de un tumor, a partir de una resección virtual sobre la superficie craneal superior derecha, mediante el uso de una geometría regular [28].	18
Figura 4. Implementación de la técnica de planos oblicuos y volume rendering, para mejorar la visualización de un modelo tumoral 3D, donde el plano es perpendicular a la dirección de acceso al tumor	18
Figura 5. Imagen de difusión del tracto corticoespinal	21
Figura 6. Múltiples vistas en la interfaz de neuroendoscopia	26
Figura 7. Metodología seguida para la representación tridimensional de los datos, se incluye los algoritmos o herramientas utilizadas en cada paso.	29
Figura 8. Segmentación del tumor con crecimiento de regiones.	31
Figura 9. Proceso para la segmentación del cerebro. a) Datos originales $H(x,y)$. b) Extracción del Cerebro $B(x,y)$	32
Figura 10. Segmentación del Pericráneo.	32
Figura 11. Segmentación del tracto corticoespinal. a) Imagen DWI. b) Imagen DTI. c) Caracterización del tamaño del tensor. d) Caracterización de la forma del elipsoide de difusión. e) Región de interés.	33
Figura 12. Reconstrucción tridimensional del tumor con cubos marchantes.	34
Figura 13. Representación tridimensional del cerebro.	35
Figura 14. Representación tridimensional del pericráneo.	35
Figura 15. Reconstrucción tridimensional del tracto corticoespinal. a) Imagen DTI. b) Selección de la región de interés. c) Tracto corticoespinal.	36
Figura 16 . Volume y surface rendering con un plano de clipping para explorar las relaciones espaciales en una tomografía computarizada de cabeza	39
Figura 17. Resección de un área específica de tejido hepático con una esfera por "borrado"(izquierda). Se utiliza clipping selectivo para mostrar los vasos sanguíneos que podrían ser afectados en una resección real. El resultado se muestra también en 2D (derecha)	40
Figura 18. Algoritmo de craneotomía virtual.	43
Figura 19. Reconstrucción volumétrica del pericráneo y representación de la herramienta de craneotomía simulada con un modelo de cono.	44
Figura 20. Vista obtenida después de realizar la craneotomía virtual.	45
Figura 21. Craneotomía virtual en paciente con lesión tumoral hemisferio derecho.	46
Figura 22. Metodología seguida para la manipulación del entorno por medio de un periférico de entrada.	48
Figura 23. Transformaciones aplicadas al brazo mecánico.	50
Figura 24. Esquema del circuito desarrollado.	51
Figura 25. Flujo de información para la conexión del dispositivo mecatrónico con 3D Slicer.	52

Figura 26. Paquete de información enviada al computador. Los números 0,1,2,3,4,5 y 6 son bits de arranque.....	53
Figura 27. Diagrama de flujo que resume los pasos a seguir para la adquisición de la información desde el periférico de entrada hacia el software de visualización..	54
Figura 28. Operación de los potenciómetros independientes y su relación con la cámara de la escena.....	55
Figura 29. Periférico de entrada.....	56
Figura 30. Flujo de información en el simulador desarrollado.....	58
Figura 31. Simulador: Paso 1, Selección del caso.	59
Figura 32. Simulador: Paso 1, Caso cargado en la escena.....	60
Figura 33. Simulador: Paso 2, Información del caso seleccionado.....	60
Figura 34. Simulador: Paso 2, Opción de visualización del pericráneo.	61
Figura 35. Simulador: Paso 2, opción de visualización del cerebro.....	61
Figura 36. Simulador: Paso 2, Lista de estructuras disponibles.	62
Figura 37. Simulador: Paso 3, Iniciar craneotomía, periférico no conectado.	63
Figura 38. Simulador: Paso 3, Iniciar craneotomía, periférico conectado.	63
Figura 39. Simulador: Paso 3, Activar control de la cámara desde el periférico. ..	64
Figura 40. Simulador: Paso 3, Detener craneotomía.....	65
Figura 41. Simulador y conexión con el periférico de entrada.	65

INTRODUCCIÓN

En procedimientos de neurocirugía es necesario una alta precisión para localizar y distinguir la profundidad y superficie de las estructuras que son críticas a nivel neurológico [2]. Con el advenimiento de las imágenes neurológicas volumétricas y funcionales, muchos de los procesos pueden ser planeados previamente por el neurocirujano en diferentes software que permiten la visualización tridimensional del cerebro, tumores, árbol venoso, la corteza elocuente y fibras de materia blanca [3].

Aunque el objetivo de los software de planificación neuroquirúrgica es aumentar las posibilidades de visualización para el cirujano, lo que implica procedimientos más seguros [4]–[6], la manipulación de los volúmenes 3D no siempre es fácil para los médicos que generalmente usan imágenes 2D.

Hay diferentes clases de simuladores, como los entrenadores Part-task, que se centran en la repetición de un aspecto particular de un procedimiento [7]. El propósito de estos simuladores es ayudar al cirujano con la adquisición de conocimientos técnicos, psicomotrices, o habilidades de los diferentes procedimientos.

Este trabajo de investigación, presenta el diseño y desarrollo de un entorno virtual en 3D Slicer [8] para la simulación de intervenciones quirúrgicas en neurocirugía, enfocado en la visualización de craneotomías para la resección de tumores cerebrales. El entorno contiene escenarios con representaciones tridimensionales de imágenes médicas de diferentes pacientes que incluyen el pericráneo, el cerebro y el tracto cortico espinal. El entorno incluye un algoritmo de craneotomía virtual que permite eliminar segmentos específicos del pericráneo. La aplicación incluye un dispositivo electromecánico que ayuda a manipular el volumen 3D y la herramienta de craneotomía virtual.

El primer capítulo del trabajo de investigación habla de la importancia de disponer de simuladores para el entrenamiento de neurocirujanos en formación, los objetivos del trabajo, y el estado del arte asociado a esta investigación, con un resumen de los proyectos relacionados que están siendo desarrollados en Colombia y en el mundo.

En el segundo capítulo, se describen los tipos de imágenes utilizadas en este trabajo de investigación y la metodología que se siguió para la implementación de los algoritmos para el procesamiento, visualización y representación tridimensional de imágenes de resonancia y tomografía, además se muestran los resultados obtenidos al aplicar dichos algoritmos, y se incluye la información sobre los software utilizados en el desarrollo del trabajo de investigación.

En el tercer capítulo, se describen las técnicas existentes para la exploración de datos volumétricos y se detalla el desarrollo de un algoritmo que permite la simulación de craneotomías en representaciones tridimensionales, además se presentan los resultados obtenidos al aplicar el algoritmo desarrollado tal como la simulación de las vistas quirúrgicas que se obtienen luego de la craneotomía virtual.

El cuarto capítulo, describe algunos de los dispositivos existentes para la interacción con representaciones tridimensionales de imágenes médicas y además contiene la información sobre la implementación de un protocolo para la manipulación del entorno por medio de un periférico de entrada que permite relacionar coordenadas en el espacio físico con el espacio virtual.

El quinto capítulo, incluye información sobre el desarrollo de la aplicación final, la cual es de libre distribución y se desarrolló para su posterior uso por los estudiantes de la especialización de neurocirugía y cirujanos en ejercicio.

En el apartado de conclusiones, se presenta un análisis del entorno que se obtuvo, además se plantean estudios adicionales a realizar y trabajos futuros.

1 PRELIMINARES

1.1 PLANTEAMIENTO DEL PROBLEMA

El entrenamiento médico es una de las partes más importantes de la medicina. Tradicionalmente, la formación de los neurocirujanos para la realización de cirugías se lleva a cabo con pacientes reales, situación en la que pueden ocurrir complicaciones como hemorragias internas graves o incluso la muerte. Por esta razón es importante resaltar el impacto del entrenamiento médico y entender que la insuficiencia en la formación de éste podría costar vidas humanas. Con el fin de evitar complicaciones graves y consecuencias irreversibles, se debe tener como objetivo prioritario que los médicos obtengan las habilidades necesarias para realizar determinados procedimientos y puedan planificar cuidadosamente cada una de las cirugías.

En la planificación tradicional de neurocirugías, los médicos basan la estrategia que llevarán a cabo utilizando un conjunto de cortes de estudios de imágenes médicas, lo que en algunos casos, como en cirugías mínimamente invasivas, podría no proporcionar la información suficiente para realizar la cirugía exitosamente, y generar complicaciones graves, ya que se carece de acceso rápido y de control sobre el problema debido al limitado campo de visión.

Para enfrentar las deficiencias de los métodos tradicionales de enseñanza y realizar una planeación adecuada de las neurocirugías, surgen los sistemas de neurocirugía guiada por imágenes (IGNS), los cuales se han convertido en un apoyo para los neurocirujanos en ejercicio y en formación, esto gracias a que los avances en las imágenes digitales han cambiado los paradigmas en la práctica de la medicina proporcionando datos precisos de detalles relevantes de cada paciente. Una aplicación directa de los sistemas IGNS se da en la planeación de las cirugías, porque facilita la localización de las estructuras sobre las que se desea intervenir, la definición de la relación anatómica con otras estructuras y la especificación en una etapa pre-operatoria de la trayectoria de mínimo riesgo, también permite representar el procedimiento en un modelo virtual de un paciente específico garantizando la viabilidad técnica o prever posibles dificultades; además estos sistemas también permiten el aprendizaje de técnicas específicas como la neuronavegación con marco esterotáxico o neurocirugía endoscópica en un entorno sin riesgos antes de llevar a cabo la experiencia en vivo.

Algunos de los sistemas de neurocirugía guiada por imágenes encontrados en el medio, como "The virtual Brain Project", están dirigidos a desarrollar simulaciones para procedimientos neuroquirúrgicos a partir de modelos deformables [9]. El "Dextroscope" integra las imágenes de tomografía computarizada (CT) e imágenes de resonancia magnética (MRI) en objetos 3D volumétricos que se pueden ver estereoscópicamente [10]. El "ROBO-SIM" es otro simulador existente en el medio y se utiliza para la planificación y simulación para neurocirugía mínimamente invasiva [11].

Aunque los simuladores mencionados presentan muy buenas características, los costos involucrados en su adquisición (aproximadamente 300.000 dólares) y su posterior soporte son altos, lo que los hace poco accesibles a entornos educativos, además, tampoco ofrecen todas las opciones de visualización existentes ya que las opciones más avanzadas requieren conocimientos mayores de procesamiento de imágenes [12].

En Colombia no se utiliza ningún tipo de software para la planeación de neurocirugías, ésta generalmente se realiza sobre imágenes 2D que no entregan suficiente información sobre la relación entre las estructuras anatómicas del paciente a tratar, por esta razón se observa la necesidad de realizar una apropiación de las tecnologías existentes, tanto en el área de la electrónica y la informática, desarrollando entornos visuales realistas a partir de imágenes obtenidas de estudios de tractografía [13] y datos volumétricos [14], puesto que se ha mostrado que integrar este tipo de imágenes de pacientes con tumores cerebrales puede proveer información adicional sobre el curso de los tractos de materia blanca y su relación con el tumor [15], mejorando las opciones del especialista a la hora de tomar decisiones.

La aplicación desarrollada no involucra interfaces hápticas; se busca mejorar la apariencia de los datos médicos volumétricos y definir los métodos más adecuados de interacción, utilizando un periférico de entrada que reemplace el mouse y permita relacionar coordenadas en el espacio físico con el espacio virtual, con el objetivo de lograr una interacción más intuitiva y una mejor aproximación a la forma como el médico visualiza el interior del paciente y lo relaciona con las imágenes, todo esto con el fin lograr un entorno en el cual, los residentes de neurocirugía puedan practicar la técnica de craneotomía virtual, además se puedan familiarizar con este tipo de herramientas y en un futuro utilizarlas para la planeación de cirugías, garantizando una intervención con menor riesgo sobre el paciente.

1.2 OBJETIVOS DEL TRABAJO DE INVESTIGACIÓN

1.2.1 General

Desarrollar una aplicación para la definición de abordajes neuroquirúrgicos a partir de estudios volumétricos de resonancia y tomografía.

1.2.2 Específicos

- Implementar un algoritmo de visualización para la representación tridimensional de los datos en resonancia y tomografía.
- Desarrollar un algoritmo que permita la simulación de craneotomías en representaciones tridimensionales y la simulación de las vistas quirúrgicas que se obtienen luego de la craneotomía virtual.
- Implementar un protocolo para la manipulación del entorno por medio de un periférico de entrada que permita relacionar coordenadas en el espacio físico con el espacio virtual.
- Implementar los algoritmos desarrollados en una aplicación de uso libre para su posterior uso por los estudiantes de la especialización de neurocirugía y cirujanos en ejercicio.

1.3 PUBLICACIONES DERIVADAS

Duque, S. I., Ochoa, J. F., Botero, A. F., & Ramírez, M. (2015, August). Algorithm for simulation of craniotomies assisted by peripheral for 3D virtual navigation. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 7043-7046). IEEE.

Botero, A. F., **Duque, S. I.**, Ochoa, J. F., & Hernandez, A. M. (2014). Diagnostic Imaging Viewer using the Leap Motion Positioning System and 3D Slicer Software. *Revista Facultad de Ingeniería* (Aceptado)

Botero, A. F., Higueta, D. E., Ramírez, M., Velásquez, J. J., Rincón, M. A., **Duque, S. I.**, & Ochoa, J. F. (2013). Técnicas de Visualización para el Entrenamiento Neuroquirúrgico en Software Libre. *Pan American Health Care Exchanges (PAHCE). Conference, workshops, and exhibits.*

1.4 ESTADO DEL ARTE

La Tabla 1 presenta un resumen del estado del arte presentado en esta sección.

Tabla 1. Lista de entornos virtuales discutidos en este documento. Cada sistema se identifica por el autor o grupo que lo desarrolló

<i>Autor</i>	<i>Año</i>	<i>Componentes</i>	<i>Módulos Quirúrgicos</i>
<i>Phillips</i>	1999	Modelos 3D	Canulación del sistema ventricular intracraneal.
<i>Kockro</i>	2000	Estereoscópico, 3D	Planeación pre quirúrgica.
<i>Radetzky</i>	2000	Táctil, Háptico	Planeación de cirugía mínimamente invasiva.
<i>Larsen</i>	2001	Modelos 3D	Planeación pre quirúrgica, simulación en modelos 3D estándar del cerebro.
<i>Brown</i>	2006	Háptico	Ventriculostomía endoscópica del tercer ventrículo.
<i>Wong</i>	2006	Imágenes digitales específicas del paciente 3D	Craneotomía y clipado de aneurisma intracraneal.
<i>Joshi</i>	2008	Imágenes tridimensionales	Visualización de regiones de interés utilizando herramientas de recorte que permiten el recorte de la anatomía del cerebro.
<i>Díaz</i>	2008	Detección de colisiones, interfaz física.	Laparoscopia.
<i>Pérez</i>	2008	Detección de colisiones	Cirugía endoscópica transesfenoidal.
<i>Vivas</i>	2011	Háptico	Laparoscopia.
<i>Delorme</i>	2011	Sistema de visión estéreo, herramientas hápticas bimanuales.	Simulación de extirpación de tumores.

Ochoa	2013	Visualización de electrodos subdurales	Planeación de cirugía de epilepsia.
Alaraj	2015	Háptico	Clipado de aneurismas.
Duque	2016	Herramienta manual, modelos 3D	Simulación de craneotomías.

1.4.1 **Sistemas de Neurocirugía Guiada por Imágenes**

El desarrollo de los sistemas de neurocirugía guiada por imágenes (IGNS) han marcado una pauta en el entrenamiento y planeación en neurocirugía, facilitando la tarea de los cirujanos, puesto que permite una visión más amplia de las relaciones anatómicas del paciente y de esta forma se puede planear la ruta más adecuada para llegar a la lesión.

The virtual Brain Project es uno de los desarrollos actuales de navegación neuroquirúrgica y está dirigido a desarrollar simulaciones de procedimientos neuroquirúrgicos. Este simulador está enfocado en modelos 3D específicos para cada paciente, los modelos son deformables y tienen sistemas de demostración visual y táctil [9].

El Dextroscope, es un sistema de neuronavegación que integra imágenes de tomografía computarizada (CT) e imágenes de resonancia magnética (MRI) en objetos 3D volumétricos que pueden verse estereoscópicamente. Con ésta tecnología, un cirujano puede realizar una simulación virtual con imágenes propias del paciente, esto permite a su vez evaluar diferentes métodos quirúrgicos desde un sinnúmero de ángulos, antes de realizar cualquier procedimiento. El software proporciona un conocimiento profundo de las relaciones anatómicas y las patologías, para la evaluación y planificación quirúrgica, también utiliza dos controladores posicionales con seis grados de libertad, los cuales funcionan como extensiones de las manos del cirujano, proporcionando la capacidad para interactuar con las imágenes de los pacientes [10].

También se encuentra en este grupo de simuladores el software ROBO-SIM, que es una herramienta de planificación y simulación para neurocirugía mínimamente invasiva. Utiliza imágenes de pacientes reales e incluye todos los pasos de planificación necesarios, tales como: definición del punto de trepanación para la entrada en el cráneo y el punto de destino dentro de la profundidad del cerebro, control de la ruta quirúrgica, realización de trepanaciones virtuales (craneotomía virtual), y definición de los volúmenes elegidos para su uso con un manipulador activo intraoperatorio. Con este sistema los neurocirujanos son capaces de simular intervenciones quirúrgicas enteras directamente en la anatomía del paciente utilizando los mismos instrumentos que se utilizan en la cirugía real [11].

Se desarrolló también el sistema de visualización y navegación VizDexter [16], que utiliza conjuntos de datos fusionados y coregistrados específicos de pacientes, que

se transforman en un objeto virtual tridimensional y puede ser visto estereoscópicamente con gafas tridimensionales y manipulado con instrumentos especiales en un entorno de realidad virtual. Este sistema permite cierta medida de planificación prequirúrgica y de ensayos perioperatorios, pero carece de retroalimentación sensorial (tecnología háptica) para el usuario y tiene algunas limitaciones debido a la representación gráfica de los tejidos. Sin embargo, se ha utilizado como complemento en cursos de formación sobre cirugía ósea temporal, cirugía de tumores y planificación de cirugías complejas.

Hay diferentes clases de simuladores, como los entrenadores Part-task, que se centran en la repetición de un aspecto particular de un procedimiento [7]. El propósito de estos simuladores es ayudar al cirujano con la adquisición de conocimientos técnicos, psicomotrices, o habilidades de los diferentes procedimientos. Ejemplos de estos entrenadores son los diseñados para la colocación del catéter de derivación ventricular, que proveen retroalimentación háptica y simula la resistencia y relajación del paso de catéter virtual 3D de ventriculostomía a través del parénquima cerebral. También contiene una interfaz gráfica tridimensional dinámica que cambia de perspectiva visual cuando la cabeza del usuario se mueve [17]. Otro ejemplo de este tipo de simuladores es el desarrollado por Phillips et al [18], disponible en internet, que permite a los usuarios practicar la canulación del sistema ventricular intracraneal. En esta simulación se usan capas anatómicas genéricas y translúcidas (cuero cabelludo, cráneo y ventrículos) para permitir al usuario desarrollar una apreciación de puntos de referencia de la superficie anatómica en relación con los ventrículos.

Alaraj et al. desarrollaron un simulador de realidad virtual con retroalimentación háptica para el grapado de aneurismas utilizando la plataforma Immersive Touch. Se creó un modelo de un aneurisma de la arteria cerebral media a partir de una angiografía por tomografía computarizada, el sistema proporciona un entorno de realidad virtual de inmersión 3D que incluye la deformación del aneurisma, del vaso y retroalimentación háptica, y se puede simular la rotura intraoperatoria del aneurisma. Diecisiete residentes de neurocirugía probaron el simulador e hicieron comentarios sobre su utilidad y la semejanza con la cirugía de grapado real de un aneurisma [19].

Así mismo, existen entornos diseñados para la simulación de procedimientos basados en craneotomías, un ejemplo de éstos es el NeuroTouch que permite la simulación de la extirpación de tumores usando un enfoque de craneotomía en tres habilidades de entrenamiento, uso del aspirador quirúrgico y ultrasónico, uso del electrobisturí bipolar y uso de las microtijeras. Sus principales componentes son un sistema de visión estéreo, herramientas hápticas bimanuales y un computador de gama alta [20].

Algunos de estos sistemas incluyen herramientas para visualizar las regiones de interés utilizando herramientas de recorte que permiten el corte regular (esférico, cúbico, cilíndrico y elipsoidal) de la anatomía del cerebro [21], [22].

Echegaray et al. desarrollaron un simulador de perforación craneal e interacción quirúrgica con el cerebro, en el cual se utilizan modelos obtenidos de tomografía computarizada (TC) y datos de resonancia magnética (MRI) de los pacientes. Un dispositivo háptico controla la herramienta virtual quirúrgica del simulador; la retroalimentación de fuerza aumenta el realismo de la simulación y la inmersión del usuario. Para simular la deformación del cerebro debido a la interacción con la herramienta, el simulador emplea una versión no lineal del método del elemento finito (FEM), de esta forma se reproduce realísticamente el comportamiento físico del tejido. Además de utilizar el simulador para el entrenamiento, los cirujanos pueden usarlo para planear y ensayar procedimientos neuroquirúrgicos complejos [23].

1.4.2 Estado del Arte en Colombia

En Colombia se han desarrollado aplicaciones para la simulación de diferentes tipos de cirugía que están orientadas al entrenamiento de cirujanos y al aprendizaje de habilidades específicas.

En la Universidad EAFIT se desarrolló un simulador quirúrgico para entrenamiento de procedimientos de laparoscopia que incluye una interfaz física y un entorno virtual correspondiente para la simulación quirúrgica. El entorno se basa en tres escenarios de formación, dos escenarios entrenan en tareas simples de laparoscopia tales como transporte de objetos con los instrumentos y el tercer escenario entrena en un procedimiento complejo, es decir, un procedimiento real [24].

En la Universidad Nacional de Colombia se desarrolló un módulo de navegación para un sistema de entrenamiento virtual aplicado a cirugía de la base del cráneo, el cual consiste en modelo de navegación de un endoscopio rígido empleado en un sistema de entrenamiento virtual aplicado a cirugía endoscópica transesfenoidal. La interacción usuario-endoscopio-tejido se modeló en dos partes. La primera es un sistema de navegación sobre voxels para detectar las colisiones considerando la geometría del endoscopio y la segunda es un modelo dinámico que relaciona la fuerza ejercida por el cirujano en un extremo del endoscopio con la respuesta de la colisión en el tejido [25].

También se construyó un simulador quirúrgico para operaciones de laparoscopia utilizando interfaces hápticas en la Universidad del Cauca. En el marco de este proyecto han sido diseñados dos robots: el robot quirúrgico Lapbot que es una estructura de nueve grados de libertad, con seis articulaciones activas y tres pasiva; también se diseñó el robot porta endoscopio Hibou, el cual es un robot asistente que sostiene la cámara al interior del abdomen del paciente y es controlado directamente por el cirujano. Consta de siete articulaciones, de las cuales cinco son activas y dos son pasivas. También se desarrolló una interfaz háptica propia. La interfaz de usuario del simulador fue desarrollada en Ogre3D. Además el sistema

desarrollado permite realizar el seguimiento de las trayectorias necesarias para una colecistectomía laparoscópica e incluye algoritmos de detección de colisiones [26].

En la Universidad de Antioquia se han adelantado varios proyectos que dan cuenta de la utilidad de desarrollar aplicaciones para la planeación de cirugías, en el trabajo "Implementación de un método para la visualización de electrodos subdurales en cirugía de epilepsia", Ochoa et al introducen un método en el cual se contrasta la implementación de los electrodos utilizando las placas convencionales (Figura 1) y la implementación utilizando técnicas de visualización volumétrica que muestran la a relación de la lesión con los electrodos y la anatomía tridimensional del paciente (Figura 2) [27].

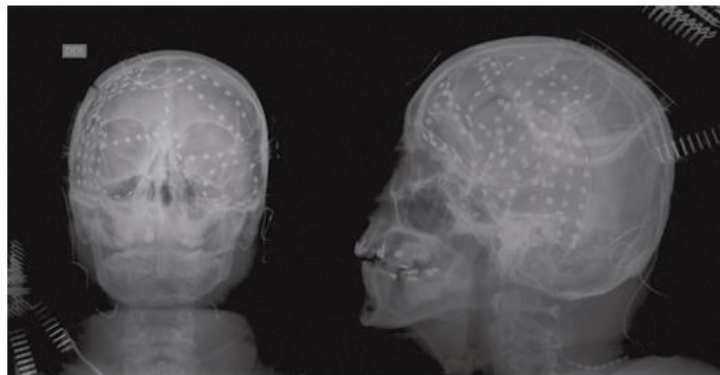


Figura 1. Proyección AP y lateral en la cual se pueden observar los electrodos subdurales. Dado que la imagen de rayos X es una proyección, no es posible conocer exactamente la relación espacial de los electrodos [27].

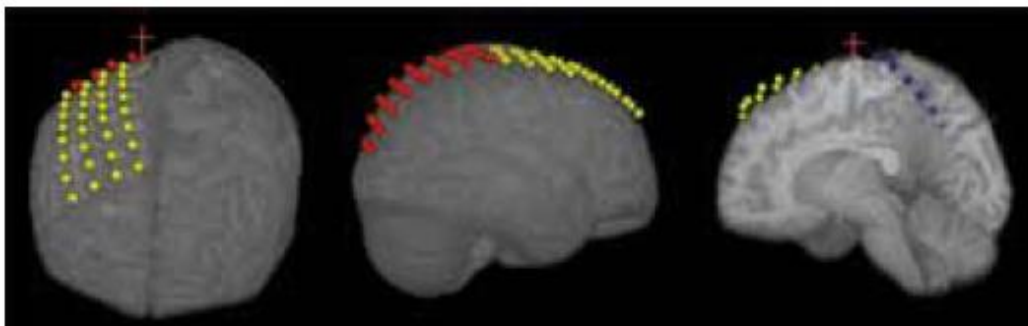


Figura 2. Representación 3D final de la invasión con 2 rejillas de 32 electrodos y una tirilla de 8 electrodos [27].

También se han llevado a cabo algunos proyectos por parte de los estudiantes del programa de bioingeniería, en los cuales se han desarrollado algunos algoritmos para visualización como un algoritmo de craneotomía virtual a partir de clipping, el cual permite la visualización de zonas cerebrales específicas, con la ayuda de figuras geométricas como esferas o cilindros (Figura 3) [28].

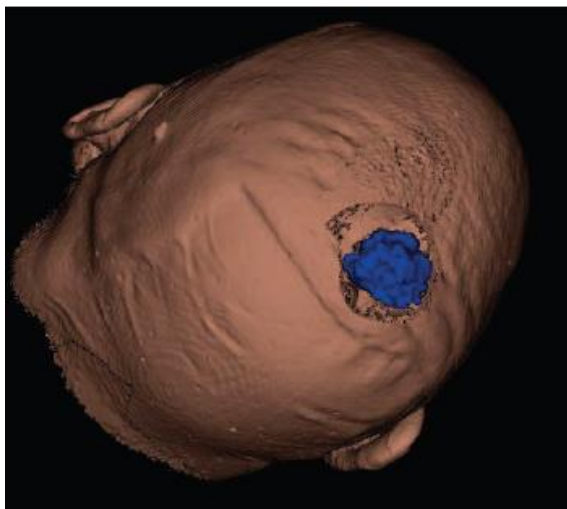


Figura 3. Visualización de un tumor, a partir de una resección virtual sobre la superficie craneal superior derecha, mediante el uso de una geometría regular [28].

Adicionalmente, se desarrolló un algoritmo de navegación mediante planos oblicuos, ya que los planos anatómicos tradicionalmente usados en las imágenes de resonancia magnética y las tomografías computarizadas (axial, sagital y coronal) no brindan al cirujano la información de forma intuitiva (Figura 4) [28].

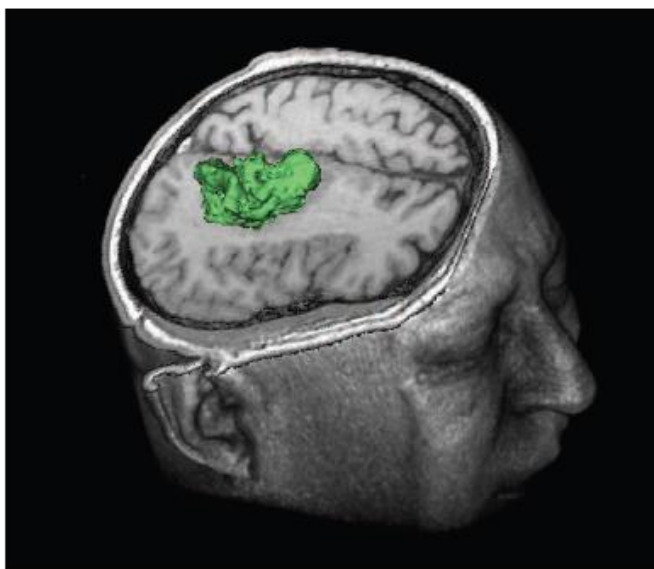


Figura 4. Implementación de la técnica de planos oblicuos y volume rendering, para mejorar la visualización de un modelo tumoral 3D, donde el plano es perpendicular a la dirección de acceso al tumor [28].

Estos algoritmos fueron utilizados en la aplicación final que se llevó a cabo en el presente trabajo de investigación.

1.4.3 Limitaciones

Los software para la simulación, tanto planeación como entrenamiento en neurocirugía, presentan resultados adecuados según lo reportado. Son muy pocos los que se enfocan en la simulación de craneotomías, y los que lo hacen solo permiten cortes regulares del pericráneo utilizando formas geométricas definidas para el acceso visual al cerebro. Tampoco se incluyen en los software enfocados en craneotomía, modelos tridimensionales con la reconstrucción de tractos de materia blanca, que permitan estudiar la relaciones entre los tractos y el tumor. Además, la interacción con los datos tridimensionales se hace generalmente con el mouse del computador, y los software que manejan otro tipo de interacción como el Dextroscope o sistemas hápticos presentan costos elevados, lo que los hace poco accesibles al entorno educativo del medio.

Debido a las limitaciones anteriores, se desarrolló un entorno virtual que permite llevar a cabo la simulación de craneotomías realizando cortes irregulares del pericráneo, además se incluyen modelos tridimensionales de pacientes con lesiones tumorales en el área motora y el tracto corticoespinal, para observar la relación entre el tumor y dicho tracto, el cual está asociado a funciones motoras. Además se incluye en el simulador desarrollado una herramienta manual que mejora la interacción del usuario con el simulador. Todo se desarrolló en código abierto para un software libre, de modo que los costos asociados son mínimos comparados con los simuladores reportados en la literatura, obteniendo de esta forma, un simulador accesible al entorno educativo.

2 VISUALIZACIÓN PARA LA REPRESENTACIÓN TRIDIMENSIONAL DE LOS DATOS DE IMÁGENES MÉDICAS.

En este capítulo, se describen los tipos de imágenes utilizadas, los diferentes software utilizados en el desarrollo del trabajo de investigación y la metodología que se siguió para la implementación de los algoritmos para el procesamiento, visualización y representación tridimensional de imágenes médicas, además se muestran los resultados obtenidos al aplicar dichos algoritmos.

2.1 IMÁGENES MÉDICAS

Los datos de imágenes médicas suelen representarse como un conjunto de imágenes individuales. Cada imagen representa un corte delgado de la parte del cuerpo escaneada y se compone de píxeles individuales. Estos píxeles están dispuestos en una malla de dos dimensiones, donde la distancia entre dos píxeles es típicamente constante en cada dirección. Los datos volumétricos combinan imágenes individuales en una malla tridimensional. En esta representación tridimensional los elementos de datos pasan a llamarse voxels (elementos de volumen) [29].

2.1.1 Selección de imágenes.

En el desarrollo del trabajo de investigación se utilizaron diferentes tipos de imágenes médicas, a continuación se hace una breve revisión sobre éstas.

2.1.1.1 Imágenes de resonancia magnética

Es una técnica de radiología que utiliza el magnetismo, ondas de radio y una computadora para obtener imágenes de las estructuras del cuerpo. El escáner de resonancia magnética es un tubo rodeado por un imán circular que crea un fuerte campo magnético que alinea los protones de los átomos de hidrógeno, que luego se exponen a ondas de radio que hacen girar los protones y producen una señal débil que es detectada por un receptor del escáner. La información recibida es procesada por un computador que se encarga de producir la imagen.

La imagen y la resolución producida por resonancia magnética son muy detalladas y pueden detectar pequeños cambios de las estructuras dentro del cuerpo. Para algunos procedimientos se utilizan agentes de contraste para aumentar la exactitud de las imágenes.

Las imágenes de resonancia magnética se pueden utilizar como un método muy preciso de detección de enfermedades en todo el cuerpo. En la cabeza específicamente, puede detectar traumas en el cerebro tales como sangrado o

hinchazón. También es posible encontrar otras anomalías como aneurismas cerebrales, accidentes cerebro vasculares, tumores en el cerebro, así como tumores o inflamación de la columna vertebral [14].

Las imágenes por resonancia magnética fueron cruciales para el desarrollo de la aplicación porque permiten ver cortes muy finos en varios planos de un tejido en particular, sobre los cuales es posible aplicar técnicas de renderización y obtener un volumen.

2.1.1.2 Imágenes de resonancia magnética por tensor de difusión (DTI)

Es uno de los avances más notables en el campo de la neuroimagen en la última década. Este método ofrece localización in vivo de los tractos de fibras neuronales. Como herramienta clínica, se dirige principalmente a las lesiones intracraneales, es decir, tumores cerebrales y malformaciones vasculares [13]. Una de sus aplicaciones importantes en la planeación neuroquirúrgica se basa en el estudio de la imagen para comprobar los tractos afectados en un tumor y poder operar respetando el mayor número de tractos indemnes.

En el desarrollo del trabajo de investigación se definió como región de interés el tracto corticoespinal, que es una vía motora de materia blanca que empieza en la corteza y termina en las neuronas motoras de la médula espinal, y controla los movimientos de las extremidades y el tronco (Figura 5).

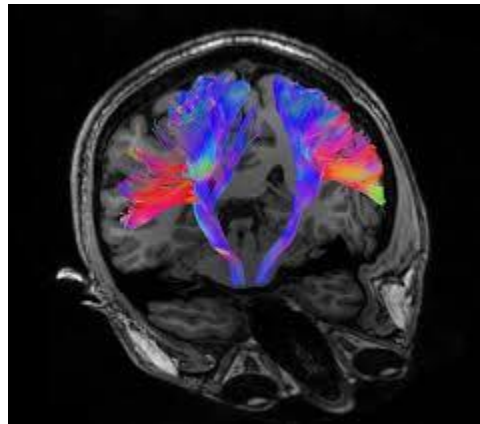


Figura 5. Imagen de difusión del tracto corticoespinal [30].

2.2 TÉCNICAS UTILIZADAS EN EL PROCESAMIENTO DIGITAL DE IMÁGENES

2.2.1 Eliminación de ruido.

Las imágenes de resonancia magnética tienen un papel importante en los procedimientos clínicos modernos y en la investigación. Debido al proceso de adquisición de las imágenes, se observa frecuentemente ruido aleatorio que disminuye la calidad de las imágenes y al momento de realizar una representación volumétrica de un conjunto de imágenes se pueden observar artefactos que no

corresponden a estructuras de la anatomía humana, por lo que se hace necesario someterlas a un preprocesamiento para la eliminación de dicho ruido.

Se han utilizado diferentes técnicas para la eliminación del ruido que con el paso del tiempo se han ido refinando. Entre estas se encuentran los filtros Gaussianos que han tenido una amplia aplicación en la eliminación de ruido de imágenes de resonancia magnética, para la implementación de estos filtros, se debe convolucionar la imagen con un núcleo Gaussiano isotrópico para suavizarla. Sin embargo las componentes de alta frecuencia son removidas con este método, y como consecuencia los bordes de las imágenes se ven borrosos [31].

Para solucionar el problema de efectos borrosos enunciado anteriormente, se han propuesto diferentes técnicas como los filtros de difusión anisotrópica, esta técnica combina una reducción de ruido de alta eficiencia y la capacidad de conservar e incluso mejorar las estructuras importantes de la imagen [32].

Otro tipo de filtro con aplicaciones en imágenes médicas es el "NL means", que no está basado en algoritmos de suavizado sino en un promedio no local de todos los píxeles en la imagen [33], este método ha despertado interés debido a su simpleza y eficacia para reducir el ruido mientras afecta mínimamente a las estructuras de la imagen original [31].

Los filtros de transformación de dominio también tienen aplicación en la eliminación de ruido. Estos métodos asumen típicamente que la señal verdadera puede aproximarse a una combinación lineal de los elementos base. Por lo tanto, preservan los coeficientes de transformación de magnitud elevada que contienen la energía de la señal verdadera y descartan el resto que están relacionados principalmente con el ruido, se puede estimar la señal verdadera libre de ruido. Se han implementado muchos filtros de transformación de dominio para la eliminación de ruido de imágenes, tales como los filtros basados en wavelets, análisis de componentes principales (PCA) y la transformada discreta del coseno (DCT) [31].

Una vez se obtiene una imagen libre de ruido se procede a realizar la segmentación de la misma.

2.2.2 Segmentación

En el procesamiento y análisis de imágenes, la segmentación consiste en la división de la imagen en conjuntos de píxeles, que compartan características comunes o algún otro criterio establecido, por ejemplo su proximidad, color, luminosidad, pertenencia a una estructura concreta, entre otros [34].

En muchas aplicaciones de imágenes biomédicas, las estructuras para segmentar se identifican en los datos, ya sea a través de delineación manual o por medio de métodos computacionales automáticos o semiautomáticos. A continuación se listan algunos de los algoritmos de segmentación más comunes:

Técnicas basadas en frontera [35]: tiene como objetivo resaltar las fronteras del objeto a segmentar. Su principal ventaja es la facilidad de implementación una vez halladas las fronteras, pero su dificultad radica en encontrar dichas fronteras.

La aplicación de este tipo de técnicas se hace en 3 fases:

- Detección de bordes
- Detección de frontera: unión de bordes
- Detección de frontera: alto nivel

Técnicas basadas en regiones [35]: tiene como objetivo resaltar las regiones de características similares tales como nivel de gris, textura, varianza entre otras. Es un procedimiento iterativo que une regiones adyacentes que cumplan ciertos criterios finalizando cuando dejan de haber regiones adyacentes que cumplan el criterio especificado.

Técnicas basadas en umbral [35]: su objetivo es diferenciar objetos de distintos niveles de gris basándose en la siguiente comparación.

Si $Im(x,y) < Th$ Objeto 0

Si $Im(x,y) \geq Th$ Objeto 1

$Im(x,y)$ es una imagen, (x,y) coordenadas

Th es el umbral para diferenciar Objeto 0 y Objeto 1

En las imágenes cerebrales se utilizan diferentes tipos de técnicas de segmentación, dependiendo de la estructura que se desee segmentar y si se tienen imágenes con anatomías normales o imágenes cerebrales con patologías como tumores o edemas. Inicialmente la imagen del cerebro es preprocesada, convirtiendo la imagen a escala de grises, y se recorta la corteza cerebral usando una máscara binaria de extracción. Para imágenes con anatomías normales se sigue un proceso para segmentar la materia blanca, la materia gris y el líquido cefalorraquídeo; en este nuevo proceso se utiliza como entrada la imagen con la corteza cerebral ya segmentada y en la que se identifica la materia gris y la materia blanca, inicialmente la imagen es suavizada utilizando un filtro de convolución Gaussiana, posteriormente, para realizar la segmentación de las estructuras mencionadas anteriormente, se aplica la técnica de operación de gradiente [36].

Para la segmentación de tumores en imágenes patológicas se utiliza el método de crecimiento de regiones (RGM), en el cual se selecciona un punto semilla inicial de la imagen de entrada (patológica), el RGM compara el valor de los píxeles vecinos con el valor inicial del punto semilla, y analiza si tienen características semejantes para añadirlos a la región, es decir, comprueba si los píxeles vecinos pueden ser incluidos en la región de la imagen perteneciente al área patológica [36].

La segmentación de edemas se hace también a partir de la imagen patológica, inicialmente se realiza un histograma de ecualización sobre la imagen, con este proceso la calidad de la imagen se ve incrementada. Luego la imagen es convertida

en una imagen indexada utilizando una función de umbralización multinivel, a partir de éste, con determinados valores para el umbral, se obtiene la segmentación de los edemas [36].

2.2.3 Registro

El registro de imágenes consiste en implementar un proceso de alineación de imágenes, de modo que las características correspondientes se puedan relacionar fácilmente entre imágenes. Estos algoritmos se componen generalmente de cuatro etapas: criterio de registro o criterio para alinear imágenes, transformaciones espaciales, sistema de optimización de parámetros y algoritmo de interpolación [37].

En algunas aplicaciones clínicas se hace necesario combinar la información obtenida de diferentes fuentes para sacar conclusiones en cuanto a un diagnóstico, la planeación de una cirugía, entre otras. El registro es esencial para comparar y analizar los datos de imágenes diferentes en un marco común. Las imágenes a analizar y comparar generalmente pertenecen al mismo paciente, pero estas pudieron ser adquiridas en diferentes instantes en el tiempo, o diferentes modalidades de imágenes[29].

El registro de imágenes es una parte esencial de toda planificación neuroquirúrgica y sistemas de navegación, dado que facilita la combinación de imágenes con información importante, complementaria, estructural y funcional para mejorar la base de información sobre la que el cirujano hace las decisiones críticas [38]. Otra aplicación importante del registro es cuando se desea comparar datos particulares de un paciente específico con un atlas que representa variaciones normales anatómicas, técnica conocida como "Atlas-based matching"[29].

Los métodos para realizar registro se clasifican en dos grupos, a continuación se encontrará una breve descripción de cada uno de los métodos:

Transformaciones rígidas: Son aquellas en las que se traslada y rota la imagen original. En tres dimensiones esto supone tres traslaciones y tres rotaciones posibles, por lo que también se denominan de 6 parámetros. Este tipo de transformación es el que se requiere para registrar imágenes de la misma o distinta modalidad del mismo paciente [39].

Transformaciones no rígidas o elásticas: Este tipo de transformaciones no son lineales, y permiten deformar elásticamente una imagen para que se parezca a la imagen de referencia. Su uso se está ampliando para modelar las diferencias entre sujetos, registrar estudios con un atlas de referencia o representar las modificaciones debidas al desplazamiento de los órganos durante la cirugía [39].

2.2.4 Reconstrucción de volúmenes

Una opción para analizar los datos de imágenes médicas de forma más intuitiva es a través de la visualización de volúmenes. Los datos volumétricos se pueden considerar como un conjunto de imágenes alineadas o cortes de la misma resolución. La visualización de volúmenes tiene como objetivo representar visualmente el conjunto de datos como un todo, es decir, todas las imágenes al mismo tiempo. Por lo tanto, los voxels individuales del conjunto de datos deben ser seleccionados, combinados, ponderados y proyectados sobre el plano de la imagen. El plano de la imagen en sí actúa literalmente como una ventana hacia los datos, que representa la posición y dirección desde el punto de vista del observador que examina el conjunto de datos [29].

Después de realizar el proceso de segmentación, registro y selección de las diferentes imágenes que harán parte del volumen, se aplican diferentes técnicas para manipular los voxels de forma que se puedan proyectar en la pantalla [37]. Entre estas técnicas se encuentra la visualización indirecta (surface rendering), cuyo objetivo es obtener una representación 3D de la superficie generando una representación intermedia del conjunto de datos que después será representado volumétricamente [29]. Un algoritmo típico de esta clase de visualización es el algoritmo de cubos marchantes [40], que es una representación poligonal de una isosuperficie. Dado que esta representación no es el conjunto de datos originales en sí, se llama representación indirecta. La visualización indirecta presenta como inconveniente una pérdida inherente de información.

Otra técnica de visualización de volúmenes empleada comúnmente es la visualización directa (volume rendering). En ésta técnica, el conjunto de datos volumétricos es representado directamente sin generar una meta-representación. La complejidad de los algoritmos de visualización directa está determinada por el número de voxels del conjunto de datos y por el número de píxeles del plano de visión. Por el contrario, la complejidad de una representación poligonal (visualización indirecta) está determinada por el número de polígonos, y no por la resolución del plano de visualización. Por consiguiente, muchos polígonos tendrán sólo una contribución pequeña o incluso insignificante, en particular si tienen un tamaño de un píxel o menos[29].

La visualización directa busca obtener una representación 3D aplicando modelos de iluminación sobre el conjunto de voxels. El inconveniente principal de esta técnica es que requiere una mayor capacidad de cómputo. Para la implementación de los algoritmos de visualización directa se encuentran diferentes software en el mercado como MedInria, MeVisLab, Osirix, 3DSlicer, entre otros [37].

Existen diferentes aplicaciones en las que ambas técnicas se emplean conjuntamente. Una aplicación concreta es un sistema de simulación basado en realidad virtual para neuroendoscopia. El simulador contiene 3 vistas (Figura 6): la parte superior es una vista endoscópica simulada; la parte inferior izquierda es la vista normal con la técnica de visualización indirecta (surface rendering), mientras

que la parte inferior derecha es la vista normal con la técnica de visualización directa (volume rendering) [41].

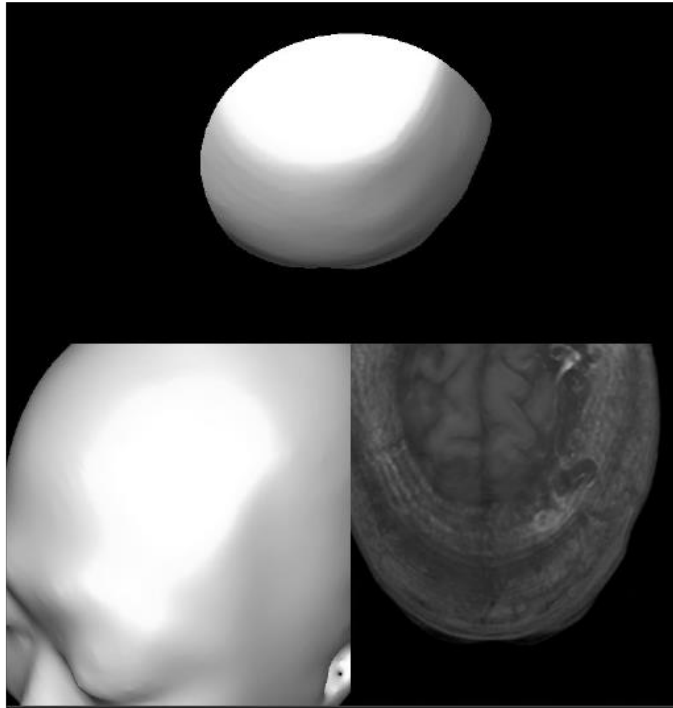


Figura 6. Múltiples vistas en la interfaz de neuroendoscopia [41].

2.3 TÉCNICAS DE VISUALIZACIÓN

Las técnicas de visualización se refieren a los diferentes puntos de vista que un usuario puede elegir para observar y analizar las imágenes ya sea en un modelo 3D o 2D.

A continuación se explican algunas de las técnicas existentes de visualización en aplicaciones de sistemas de cirugía guiada por imágenes:

2.3.1 Visualización 3D individual.

En esta técnica el usuario observa una sola representación 3D de los datos anatómicos y algunos puntos de referencia virtuales. Los cirujanos utilizan una combinación de vistas en 2D y 3D de las representaciones individuales durante la cirugía. Estos puntos de referencia virtuales son particularmente difíciles de encontrar en las vistas 2D, ya que se observan como secciones transversales circulares en dos o tres cortes. La limitación principal en este caso es que se requiere un experto con el software de visualización en la sala de operaciones para interactuar con los datos [42].

2.3.2 Visualización basada en multiventanas gráficas.

Esta técnica de visualización consiste en múltiples configuraciones de ventanas gráficas basadas en software de modelado y CAD. En esta se obtienen las vistas axial, coronal y sagital del modelo. Éstas proporcionan una mayor retroalimentación visual, pero requiere de una práctica significativa para navegar a regiones del cerebro utilizando los tres puntos de vista [42].

2.3.3 Visualización en línea.

Es una técnica de visualización que permite a los cirujanos controlar la posición de una cámara interactivamente. La posición de dicha cámara y la orientación es controlada por la posición y orientación del instrumento quirúrgico. En esta técnica se representa también virtualmente el instrumento quirúrgico. Si bien se puede establecer la exactitud de la ubicación horizontal de la sonda, es difícil determinar la distancia vertical desde el punto de interés [42].

2.3.4 Visualización en línea aumentada.

De las técnicas descritas anteriormente se determinó que los neurocirujanos prefieren utilizar la visualización en línea, pero se tenía problemas para establecer que tan lejos estaba la sonda del punto de referencia virtual [42]. Para mejorar esta situación, se desarrolló una visualización en línea aumentada, en la que se expande la visualización en línea con otras dos visualizaciones de los datos. Estos puntos de vista se obtienen haciendo girar la cámara alrededor del vector definido por el instrumento quirúrgico (la ubicación 3D de la punta del instrumento y el punto medio del instrumento quirúrgico se conocen previamente). Esta técnica de visualización proporciona información visual con respecto a la distancia desde la punta del instrumento hasta el punto de interés y ayuda a los cirujanos a navegar por los puntos de referencia [42].

2.4 SOFTWARE UTILIZADO Y LIBRERIAS

2.4.1 3D Slicer

3D Slicer es una aplicación de código abierto para el procesamiento de imágenes médicas. Como herramienta de investigación clínica, 3D Slicer es similar a una estación de trabajo de radiología que soporta visualizaciones versátiles, además también proporciona una funcionalidad avanzada como la segmentación automática y el registro para una variedad de aplicaciones. A diferencia de una estación de trabajo típica de radiología, 3D Slicer es libre y no está ligada a un hardware específico. Como plataforma de programación, 3D Slicer facilita la traducción y la evaluación de nuevos métodos cuantitativos al permitir que el investigador biomédico se pueda centrar en la implementación del algoritmo y proporciona abstracciones para las tareas comunes de comunicación de datos, visualización y desarrollo de interfaces de usuario. En comparación con otras herramientas que proporcionan los aspectos de esta funcionalidad, 3D Slicer es completamente de

código abierto y puede ser fácilmente ampliado y redistribuido. Además, 3D Slicer está diseñado para facilitar el desarrollo de nuevas funcionalidades en forma de extensiones [8]. La aplicación incluye diferentes librerías como VTK, Qt y MRML [43]–[45]

2.4.2 Python

Es un lenguaje de programación interpretado, cuya sintaxis favorece un código legible. Es multiparadigma, ya que soporta orientación a objetos, programación imperativa y programación funcional. Es un lenguaje interpretado, usa tipado dinámico además es multiplataforma.

3D Slicer incluye una consola de Python que hace que sea posible desarrollar módulos de secuencias de comandos completamente funcionales, permitiendo de este modo la creación de prototipos y simplificar el proceso de desarrollo. Tales módulos pueden aprovechar la funcionalidad Slicer y varias bibliotecas científicas basadas en Python.

2.4.3 Visualization Tool Kit (VTK)

El Kit de herramientas de visualización (VTK) es un sistema de software libre, disponible para la realización de gráficos 3D por computadora, procesamiento de imágenes y visualización. VTK consiste en una biblioteca de clases de C++ y varias capas de interfaz interpretadas como Tcl/Tk, Java, y Python [45].

2.4.4 Qt

Qt es una librería multiplataforma orientado a objetos de código abierto y libre ampliamente usada para desarrollar programas que utilicen interfaz gráfica de usuario, así como también diferentes tipos de herramientas para la línea de comandos y consolas para servidores que no necesitan una interfaz gráfica de usuario.

2.4.5 FSL

FSL es una biblioteca completa de herramientas de análisis para FMRI, MRI y los datos de imagen cerebral DTI. Se ejecuta en Apple y PC (Linux y Windows a través de una máquina virtual), y es de fácil instalación. La mayoría de las herramientas se pueden ejecutar tanto desde la línea de comandos como desde interfaces gráficas de usuario [46], la herramienta utilizada en este trabajo de investigación fue BET (Brain Extraction Tool) que elimina el tejido no cerebral a partir de una imagen de toda la cabeza. También se puede estimar las superficies interior y exterior del cráneo y el cuero cabelludo, si se tienen buenas imágenes de resonancia magnética, potenciadas en T1 y T2, de entrada [47].

2.5 IMPLEMENTACION DE UN ALGORITMO DE PARA LA REPRESENTACIÓN TRIDIMENSIONAL DE LOS DATOS

Para llevar a cabo el desarrollo de este objetivo se implementó un esquema de procesamiento y visualización para la representación tridimensional de los datos. Este esquema más que un algoritmo representa el uso de diferentes software que están disponibles en internet y la implementación en código, usando librerías para el procesamiento de imágenes médicas como ITK [48], de algoritmos para obtener la representación tridimensional de los datos de interés. En la Figura 7 se muestra la metodología seguida para la realización del objetivo. El algoritmo para la segmentación del pericráneo se programó en *python*, debido a que en los reportes de la literatura no se encuentran algoritmos para la representación tridimensional del pericráneo a partir de imágenes de resonancia magnética.

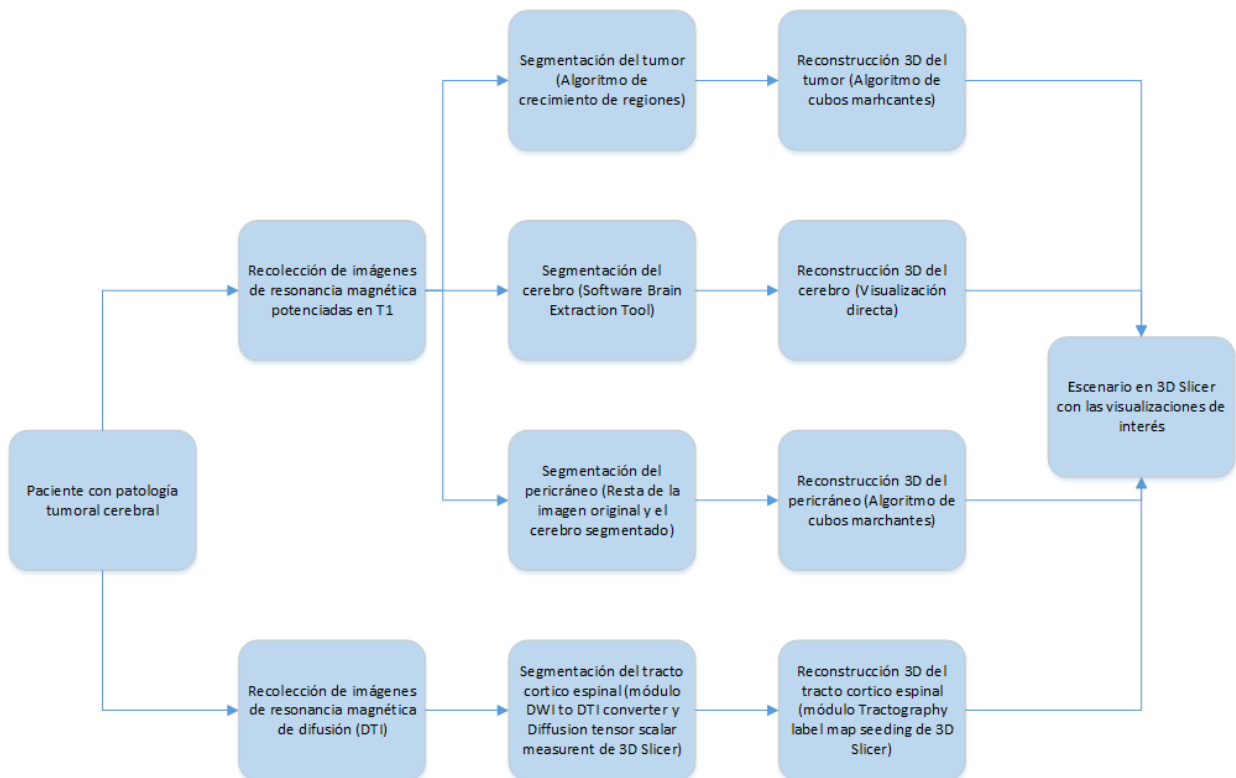


Figura 7. Metodología seguida para la representación tridimensional de los datos, se incluye los algoritmos o herramientas utilizadas en cada paso.

2.5.1 Recolección de las imágenes

Se solicitaron estudios de pacientes que presentaran tumores al Instituto neurológico de Antioquia (INDEA) que incluyeran imágenes de resonancia magnética potenciadas en T1 e imágenes de difusión por resonancia magnética (DTI) (Tabla 2).

Las imágenes potenciadas en T1 (TR = 11 ms., TE = 5.2 ms., FA = 15 , grosor = 1 mm , matriz = 224 x 256 , 176 slices por volumen) y las imágenes de difusión (TR = 7200 ms., TE = 90 ms., FA = 90, grosor = 3 mm , matriz = 128 x 128 , 650 slices por volumen) fueron adquiridas en un Sistema de resonancia magnética de 1.5 T (Siemens SymphonyVision). Estas imágenes fueron utilizadas para la información anatómica de la superficie tridimensional del cerebro.

Inicialmente se planeaba realizar la reconstrucción tridimensional del pericráneo a partir de estudios de tomografía, pero como no fue posible obtener casos de pacientes con tumores con estudios de resonancia potenciadas en T1, datos de difusión y tomografía simultáneamente, se realizó la reconstrucción 3D del pericráneo a partir de las imágenes de resonancia potenciadas en T1.

Tabla 2. Casos de pacientes recolectados

Paciente	T1 MRI	DTI	Patología
1	X		Lesión tumoral frontal derecha
2	X		Infarto en fase crónica
3	X		Lesión tumoral frontal derecha
4	X	X	Lesión tumoral frontal-parietal izquierda
5	X	X	Lesión tumoral frontal-parietal derecha
6	X	X	Lesión tumoral frontal izquierda
7	X	X	Lesión tumoral frontal-temporal-parietal izquierda

2.5.2 Segmentación

Se definieron como estructuras de interés en el desarrollo del simulador, el pericráneo, el cerebro y el tumor obtenidos a partir de las imágenes de resonancia potenciadas en T1, además el tracto corticoespinal obtenido de imágenes de difusión por resonancia magnética. A continuación se ilustran los pasos seguidos para el procesamiento de las imágenes

2.5.2.1 Tumor

La segmentación del tumor realizó utilizando el módulo de Edición disponible en 3D Slicer. Inicialmente se selecciona el área de interés (tumor) y su periferia con diferentes etiquetas (Label), posteriormente se utiliza un algoritmo de crecimiento

de regiones incluido en el módulo que permite la segmentación de cada una de las regiones asociadas a cada etiqueta de forma independiente (Figura 8).

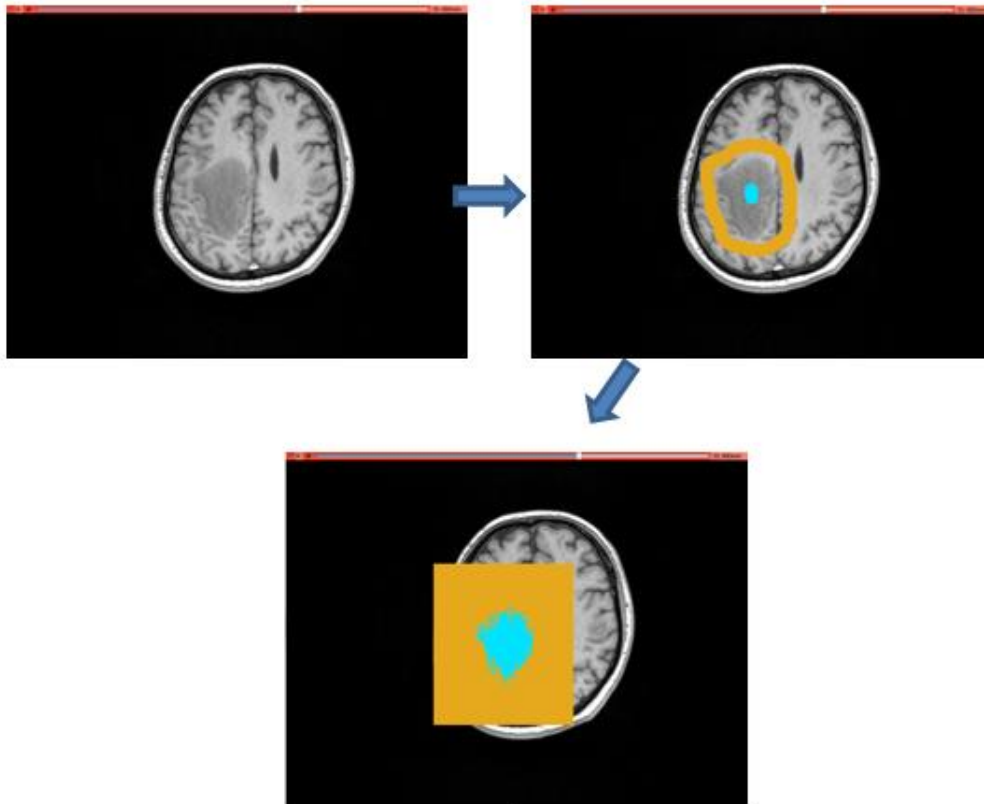


Figura 8. Segmentación del tumor con crecimiento de regiones.

2.5.2.2 Cerebro

La segmentación del parénquima cerebral se llevó a cabo utilizando el software FSL y su herramienta BET (Brain Extraction Tool) [47], la cual borra el tejido no cerebral de una imagen de resonancia de cráneo. Inicialmente, las imágenes de resonancia magnética potenciadas en T1 que se encuentran en formato DICOM se convierten a formato NIfTI utilizando la herramienta dcm2nii. Las imágenes en formato nifti $H(x, y)$ (Figura 9a) se utilizan como dato de entrada en la herramienta BET, la salida de este procesamiento es la imagen del cerebro sin el cráneo $B(x, y)$ (Figura 9b).

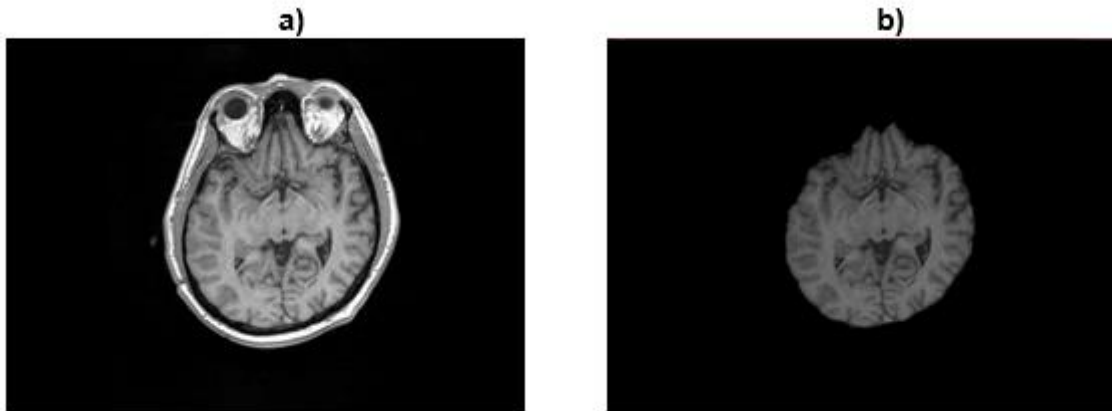


Figura 9. Proceso para la segmentación del cerebro. a) Datos originales $H(x,y)$. b) Extracción del Cerebro $B(x,y)$.

2.5.2.3 Pericráneo

Cuando se utilizan imágenes de resonancia magnética, se debe realizar un procesamiento para la segmentación del pericráneo. Utilizando las imágenes sin cráneo obtenidas en el paso anterior (con BET) $B(x,y)$ (Figura 9b) y la imagen de resonancia original $H(x,y)$ (Figura 9a), se lleva a cabo una resta de las matrices que componen las imágenes $S(x,y) = H(x,y) - B(x,y)$, de esta forma se obtiene una serie de imágenes del cráneo sin el cerebro (Figura 10), posteriormente se aplica un filtro mediano tridimensional para eliminar el ruido de las imágenes.



Figura 10. Segmentación del Pericráneo.

2.5.2.4 Tracto cortico espinal.

El procesamiento de las imágenes de difusión por resonancia magnética se llevó a cabo utilizando los módulos DWI to DTI converter y Diffusion tensor scalar measurement de 3D Slicer. Inicialmente se convirtieron las imágenes en formato DWI (diffusion weighted image, Figura 11a) a DTI (diffusion tensor image, Figura 11b), utilizando el método WSL (weighted least squares) para la estimación de parámetros, luego se caracterizó el tamaño del tensor (trace, Figura 11c) y se

caracterizó la forma del elipsoide de difusión (fa, Figura 11d). Para segmentar la región de interés (Figura 11e), se utiliza el editor de 3D Slicer, para el tracto cortico espinal se selecciona al nivel del pedúnculo cerebral en la vista axial.

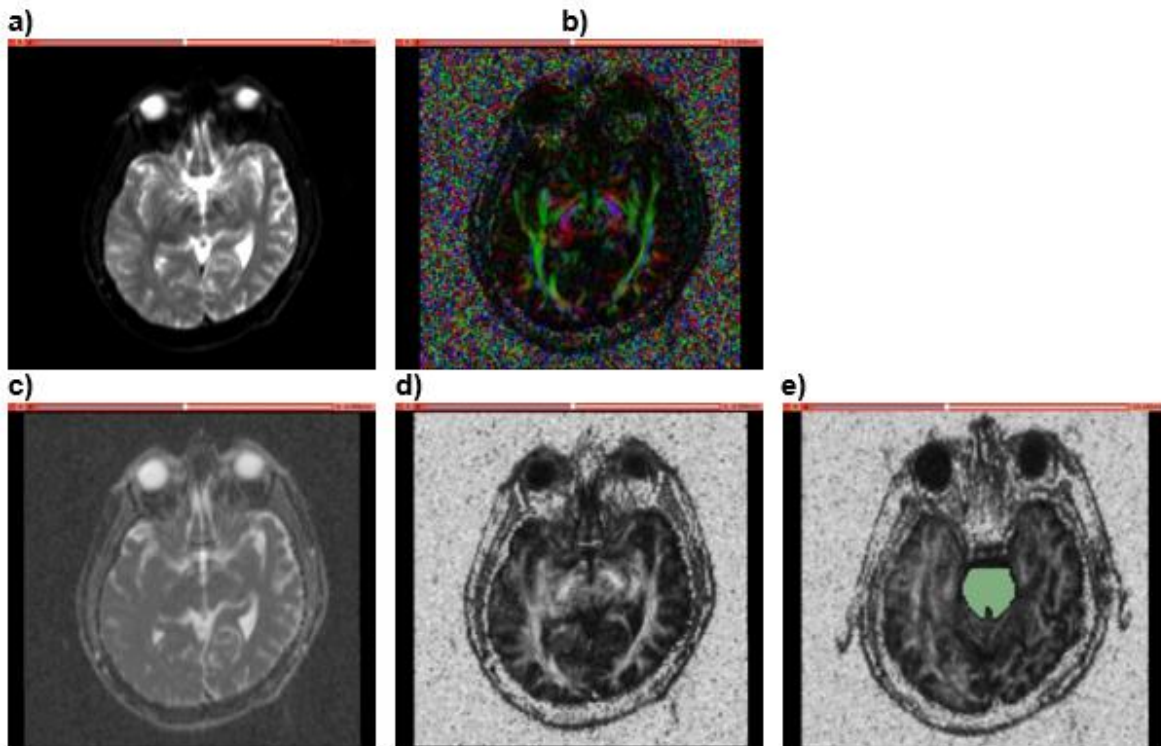


Figura 11. Segmentación del tracto corticoespinal. a) Imagen DWI. b) Imagen DTI. c) Caracterización del tamaño del tensor. d) Caracterización de la forma del elipsoide de difusión. e) Región de interés.

2.5.3 Reconstrucción tridimensional de las estructuras

2.5.3.1 Tumor

El módulo de edición de 3D Slicer permite realizar la representación tridimensional de los datos segmentados y etiquetados (tumor y vecindad) utilizando la herramienta combinar y construir (merge and build), para llevar a cabo este proceso la herramienta implementa un algoritmo de cubos marchantes sobre las estructuras etiquetadas, posteriormente se elimina la estructura correspondiente a la periferia del tumor, y se conserva la correspondiente al tumor (Figura 12).

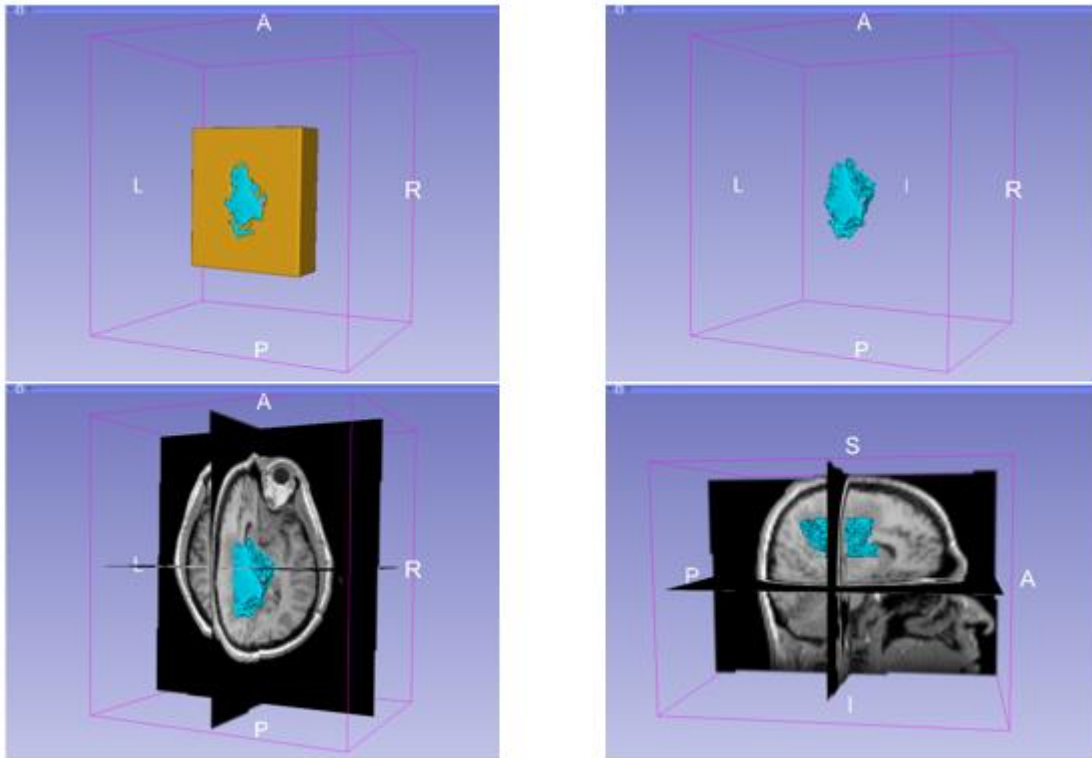


Figura 12. Reconstrucción tridimensional del tumor con cubos marchantes.

2.5.3.2 Cerebro

La representación tridimensional del cerebro se lleva a cabo utilizando un módulo diseñado para implementar el algoritmo de visualización directa de imágenes cerebrales. En la Tabla 3 se especifican los parámetros utilizados en la función de transferencia para el color y opacidad, implementada en el código.

Tabla 3. Función de transferencia para el color y la opacidad

Intensidad del Voxel	R	G	B	Opacidad
0	0	0	0	0
20	0.17	0	0	0
40	0.404	0.15	0.08	0.15
120	0.78	0.61	0.38	0.3
220	0.85	0.835	0.79	0.38
1024	1	1	1	0.5

La Figura 13 muestra el resultado obtenido al aplicar la visualización directa a la imagen del cerebro segmentado $B(x,y)$.

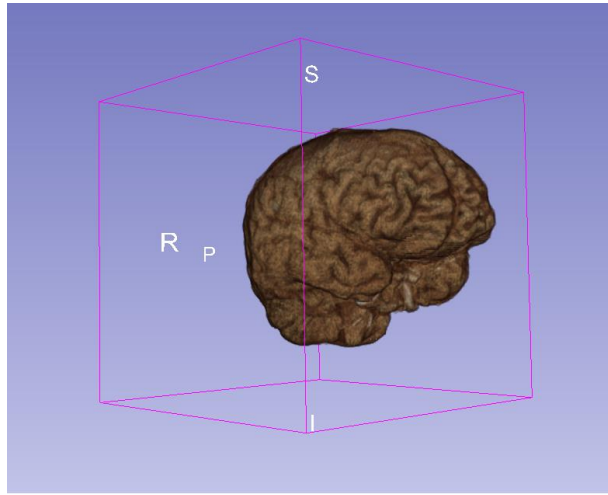


Figura 13. Representación tridimensional del cerebro.

2.5.3.3 *Pericráneo*

La reconstrucción del pericráneo (Figura 14) se llevó a cabo a partir de una umbralización de los valores de la matriz que componen la imagen $S(x,y)$ (Figura 10), y se reconstruye con un algoritmo de cubos marchantes la estructura correspondiente, finalmente se hace un suavizado y un diezmado de la estructura para mejorar la visualización de los datos y reducir la carga computacional.

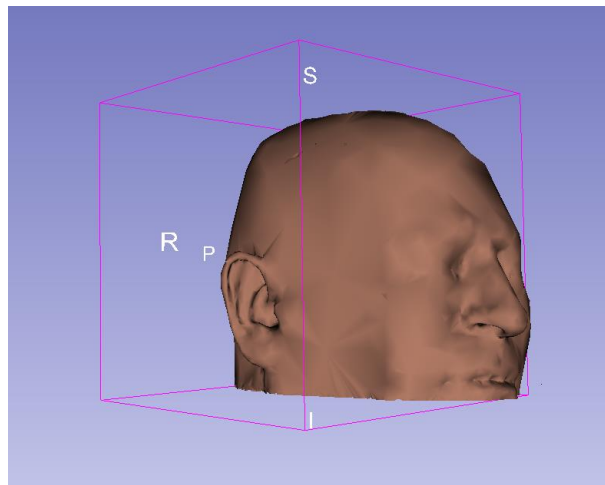


Figura 14. Representación tridimensional del pericráneo.

2.5.3.4 Tracto cortico espinal.

La reconstrucción volumétrica del tracto cortico espinal se lleva a cabo utilizando la imagen de anisotropía fraccional etiquetada, esta etiqueta indica la zona de interés asociada al tracto cortico espinal. Se utiliza el módulo de 3D Slicer “Tractography label map seeding”, que utiliza como dato de entrada la imagen DTI (Figura 15a) y la etiqueta de la región de interés (Figura 15b) y se obtiene a la salida la reconstrucción volumétrica del tracto asociado a dicha región de interés (Figura 15c).

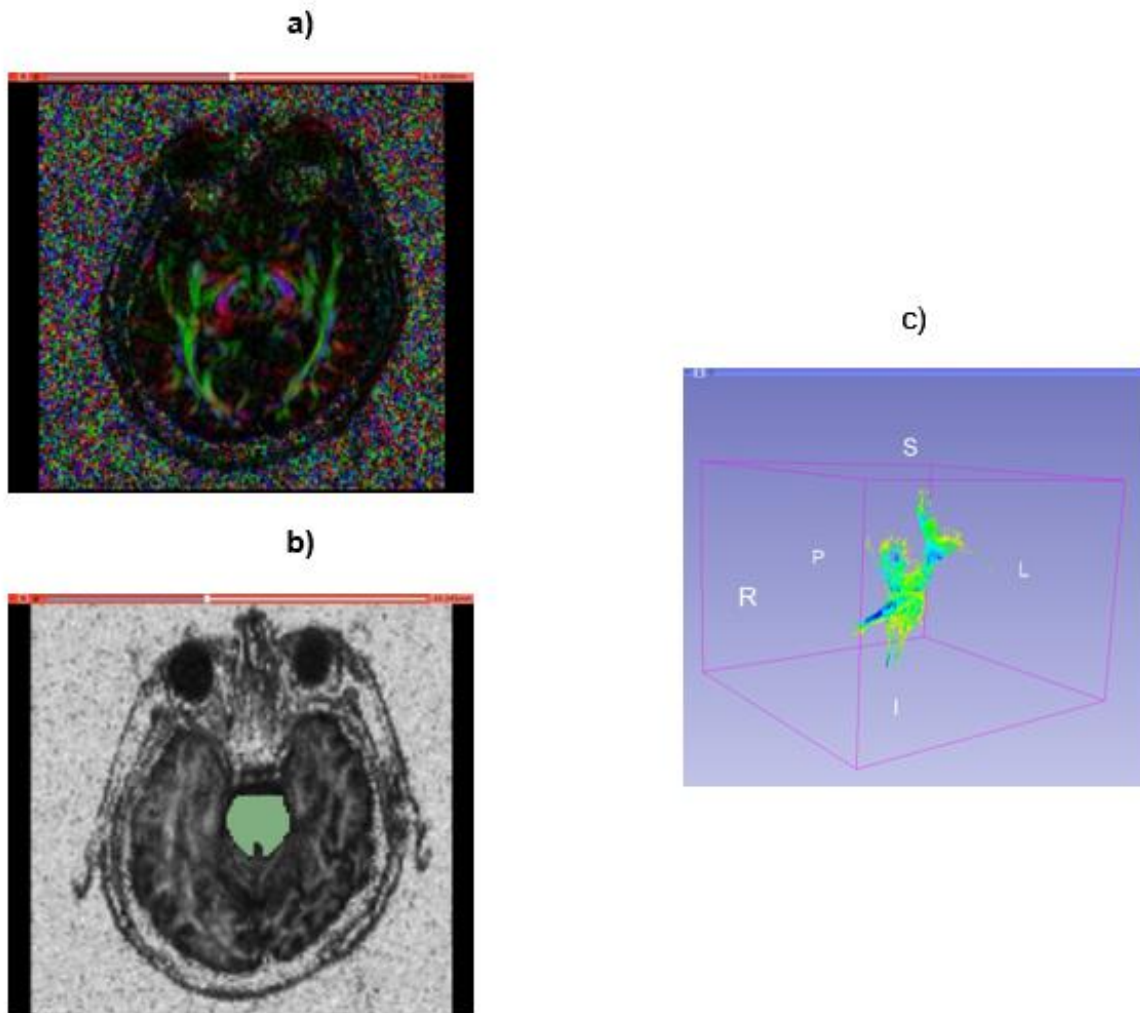


Figura 15. Reconstrucción tridimensional del tracto corticoespinal. a) Imagen DTI. b) Selección de la región de interés. c) Tracto corticoespinal.

2.6 DISCUSION

En este capítulo se ejecutó una serie de procedimientos para llevar a cabo el procesamiento de las imágenes médicas. Para su uso en aplicaciones clínicas se plantea desarrollar estrategias que permitan implementar soluciones automáticas que presenten resultados adecuados para la mayoría de casos y soluciones semiautomáticas para casos más complejos.

Las técnicas de segmentación implementadas son un requisito previo para visualizaciones de alta calidad de las estructuras que se desean analizar y las técnicas de visualización en general tienen una amplia aplicabilidad.

Los métodos para la representación volumétrica de datos implementados mostraron resultados adecuados, en general los métodos se pueden implementar para cualquier estudio de un paciente sin necesidad de hacer grandes modificaciones. Se eligieron casos de pacientes con lesiones tumorales en la zona motora, por esta razón se eligió el tracto cortico espinal en la reconstrucción tridimensional de las imágenes de difusión, así se puede observar la relación del tracto la lesión tumoral. Los software utilizados para el procesamiento de las imágenes son todos de uso libre y código abierto, lo que permite agregar nuevas funcionalidades y especificar los parámetros necesarios para cada escenario elegido.

3 SIMULACIÓN DE CRANEOTOMÍAS EN REPRESENTACIONES TRIDIMENSIONALES Y LA SIMULACIÓN DE LAS VISTAS QUIRÚRGICAS QUE SE OBTIENEN LUEGO DE LA CRANEOTOMÍA VIRTUAL.

En este capítulo se describen las técnicas existente para la exploración de datos volumétricos y se detalla el desarrollo de un algoritmo que permite la simulación de craneotomías en representaciones tridimensionales además se presentan los resultados obtenidos al aplicar el algoritmo desarrollado tal como la simulación de las vistas quirúrgicas que se obtienen luego de la craneotomía virtual.

3.1 CLIPPING

Clipping es una técnica de interacción fundamental para explorar datos volumétricos y es usada para restringir la visualización del volumen a porciones más pequeñas del mismo [29]. En esta técnica se tienen planos de visualización que pueden ser trasladados e inclinados por el usuario para obtener una representación visual óptima de las estructuras objetivo, además si se ajusta una función de transferencia apropiada y se asocia a dichos planos, es posible definir específicamente que porciones de los datos se van a visualizar (Figura 16).

Una variante especial de esta técnica es el Clipping selectivo, que es ampliamente usado para hacer énfasis sobre ciertas estructuras y presentar información contextual [29]. Aunque la aplicación inicial del Clipping selectivo fue la educación en anatomía con el VoxelMan [49], actualmente también es muy usado en planeación de cirugías [50].

Otra variante de la técnica de clipping se presenta cuando se combinan seis planos para definir un subvolumen, y se conoce como clipping de caja [29]. Esta variante de la técnica es especialmente útil cuando se desea explorar en detalle una región determinada, por ejemplo un aneurisma, o la región cercana a un tumor; con ésta técnica es posible presentar la región de interés como un volumen aislado e independiente, o combinarse con otras vistas para permitir que el usuario ubique anatómicamente dicha región.

Específicamente en el campo de las imágenes médicas, se hace uso de la técnica de clipping para explorar los datos volumétricos médicos de forma interactiva; para tal fin, en la actualidad, se hace uso de hardware grafico dedicado, tales como las unidades de procesamiento gráfico, para acelerar los procesos computacionales, y se adelantan investigaciones para plantear nuevos algoritmos que realicen algunas tareas de forma más eficiente [51], además es posible visualizar la representación volumétrica de los datos médicos, e incluso hacer procesos como el clipping en computadores comunes (sin altas especificaciones técnicas) [51]; sin embargo bajo

las arquitecturas actuales de las GPU, no es posible, con rutinas fijas y estandarizadas, recortar correctamente algunos fragmentos de los vértices de un volumen; motivo por el cual se han propuesto nuevos algoritmos que usan tanto múltiples funciones de transferencia como intersecciones entre los rayos de visualización y los planos, buscando aumentar la velocidad con la que se visualiza el volumen recortado y mejorar la calidad de los planos de clipping [52]. Además, en los algoritmos que dependen del hardware, el objetivo final es representar en volumen, pero también explorar y analizar los datos para obtener información útil de ellos, para lo cual es necesario interactuar con las imágenes de forma eficiente y en tiempo real, así, se han propuesto algoritmos que permiten visualizar dinámicamente tanto el plano de clipping como datos de volumen y además permiten adquirir todos los píxeles de dicho plano utilizando una combinación de técnicas de mapeo 3D y 2D [51]. Estas mejoras a la técnica del clipping posibilitan el aumento de la tasa de frames por segundo al momento de representar un volumen de datos y permite visualizar el plano de clipping en tiempo real.

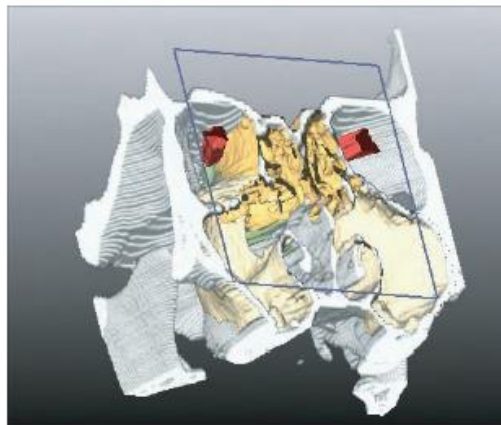


Figura 16 . Volume y surface rendering con un plano de clipping para explorar las relaciones espaciales en una tomografía computarizada de cabeza [29].

3.2 RESECCIÓN VIRTUAL

Para la simulación de vistas quirúrgicas, incluyendo las obtenidas luego de una craneotomía, se utilizan técnicas de resección virtual, entre las cuales se incluyen diferentes metodologías como resección virtual por borrado (Figura 17) que remueve el tejido seleccionado y se puede especificar regiones arbitrarias de manera más natural y más precisa [53].

También se puede encontrar la técnica de resección virtual con un plano de corte deformable, la cual tiene semejanza con la técnica anterior. En esta técnica el usuario dibuja líneas en la superficie 3D de un órgano para iniciar la resección virtual. A partir de estas líneas, que caracterizan el límite del plano de corte, se genera una malla que representa el plano de corte inicial. El plano se deforma localmente para ajustarse a las líneas trazadas por el usuario [54].

Han sido desarrolladas algunas herramientas para simular resecciones en diferentes órganos, principalmente del hígado con el fin de reducir el riesgo de insuficiencia hepática por un hígado remanente muy pequeño después de realizar una resección hepática extendida; para esto se realiza una evaluación virtual del volumen del hígado remanente [55].

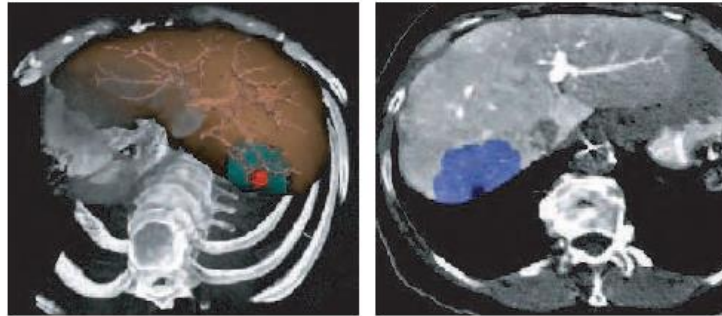


Figura 17. Resección de un área específica de tejido hepático con una esfera por "borrado"(izquierda). Se utiliza clipping selectivo para mostrar los vasos sanguíneos que podrían ser afectados en una resección real. El resultado se muestra también en 2D (derecha) [29].

3.3 CORTE DE DATOS VOLUMÉTRICOS

Las herramientas de corte son de gran importancia para la simulación de cirugías, pues gracias a ellas los usuarios pueden mover un dispositivo de corte, generalmente un escalpelo virtual, a través de los datos volumétricos y simular los procedimientos de corte [29].

En el ámbito de entrenamiento a personal médico, es necesario saber que sección del modelo geométrico fue tocado por el dispositivo de corte (por medio de detección de colisiones), y realimentar físicamente al usuario. Además se requiere de ambientes escenográficos y de manipulación realistas y se ha de tener en cuenta las interacciones que se puedan dar entre los dispositivos de corte y el tejido simulado, tales como las deformaciones elásticas y hemorragias que puedan ocurrir [29].

Para la simulación de cortes, es necesario incluir dentro del escenario 3D los dispositivos de corte que se puedan usar, como bisturís, tijeras, taladros, entre otros, y sus comportamientos con los modelos, esto se realiza por medio de algoritmos de detección de colisiones que permiten determinar que parte del modelo general está siendo afectada por cada corte.

Debido a que los requisitos de precisión y velocidad son altos en el corte virtual de superficies, y los dispositivos de corte no son necesariamente convexos, la aplicación de algoritmos de detección de colisiones en el ámbito de los ambientes virtuales para la simulación de cirugías se constituye en un área en desarrollo. En dichas aplicaciones las colisiones entre órganos deformables y dispositivos de corte

deben ser detectadas y procesadas, además debido al proceso de corte pueden ocurrir cambios topológicos que deben reflejarse en el modelo [56].

Existen muchos métodos para el proceso de detección de colisiones, como los algoritmos de jerarquías de volumen independientes (Bounding volume hierarchies), los cuales realizan una partición del modelo general en múltiples volúmenes independientes y se asocian los volúmenes a un nodo dentro de un árbol de jerarquías. Este algoritmo ha demostrado tener buenos resultados si se simulan cortes sobre objetos rígidos o modelos deformables reducidos [56]. Sin embargo si se desean simular cortes sobre superficies deformables, se deben usar métodos de partición espacial, en los cuales es posible representar cambios independientes en la topología del objeto. Dentro de este enfoque se han propuestos diversos métodos, tales como los octrees [57], arboles BSP [58] y los voxel grids [59]. Sin embargo, cuando se requiere probar el comportamiento de objetos deformables frente a objetos rígidos tales como un escalpelo, el método de distancia de campos se constituye como una solución sencilla que provee información sobre el contacto entre los objetos y de la profundidad de penetración [60]. Aunque los algoritmos anteriormente mencionados brindan algunas soluciones al problema de detección de colisiones en aplicaciones médicas, aún no se tiene un enfoque que se adapte a todas las situaciones.

Otro aspecto importante, en el ámbito de las simulaciones quirúrgicas con fines de entrenamiento, es la realimentación física que percibe el usuario al momento de hacer el corte, para lo cual se utilizan las interfaces hápticas, las cuales brindan retroalimentación y apoyo a los usuarios permitiéndoles percibir una retroalimentación física de la acción que están llevando a cabo virtualmente. La retroalimentación háptica, está basada en fuerzas computacionales que representan la interacción entre los dispositivos de corte y la anatomía de un paciente; este proceso computacional debe ser muy eficiente, pues para que el usuario perciba una retroalimentación realista, la tasa de actualización del estímulo debe ser mayor a 1KHz [29].

3.4 CRANEOTOMIA

Una craneotomía es un procedimiento para realizar una abertura ósea en el cráneo. Una sección del cráneo, llamada colgajo óseo, se elimina con el fin de acceder al cerebro. Una craneotomía puede ser pequeña o grande dependiendo del problema, pero entre mayor en tamaño sea mayor será las complicaciones para el paciente. Se puede llevar a cabo para tratar tumores cerebrales, hematomas (coágulos de sangre), aneurismas o malformaciones arteriovenosas, fracturas de cráneo, objetos extraños, inflamación del cerebro o infección. El colgajo óseo normalmente se reemplaza al final del procedimiento con pequeñas placas y tornillos. En general, una craneotomía estará precedida por una resonancia que proporciona una imagen del cerebro que el cirujano utiliza para planificar la ubicación precisa para la eliminación de hueso y el ángulo apropiado de acceso a las áreas del cerebro [61],

siendo esto un limitante al depender de la capacidad del cirujano de extrapolar representaciones bidimensionales para resolver un problema en tres dimensiones.

3.5 ALGORITMO DESARROLLADO PARA LA SIMULACIÓN DE CRANEOTOMIAS

En la Figura 18 se resume el procedimiento desarrollado para llevar a cabo la craneotomía virtual, el algoritmo fue desarrollado en *Python* y se utilizaron las librerías VTK, Qt y MRML para su uso en 3D Slicer. En el Anexo 1 se encuentra el código fuente en *Python* del algoritmo de craneotomía virtual.

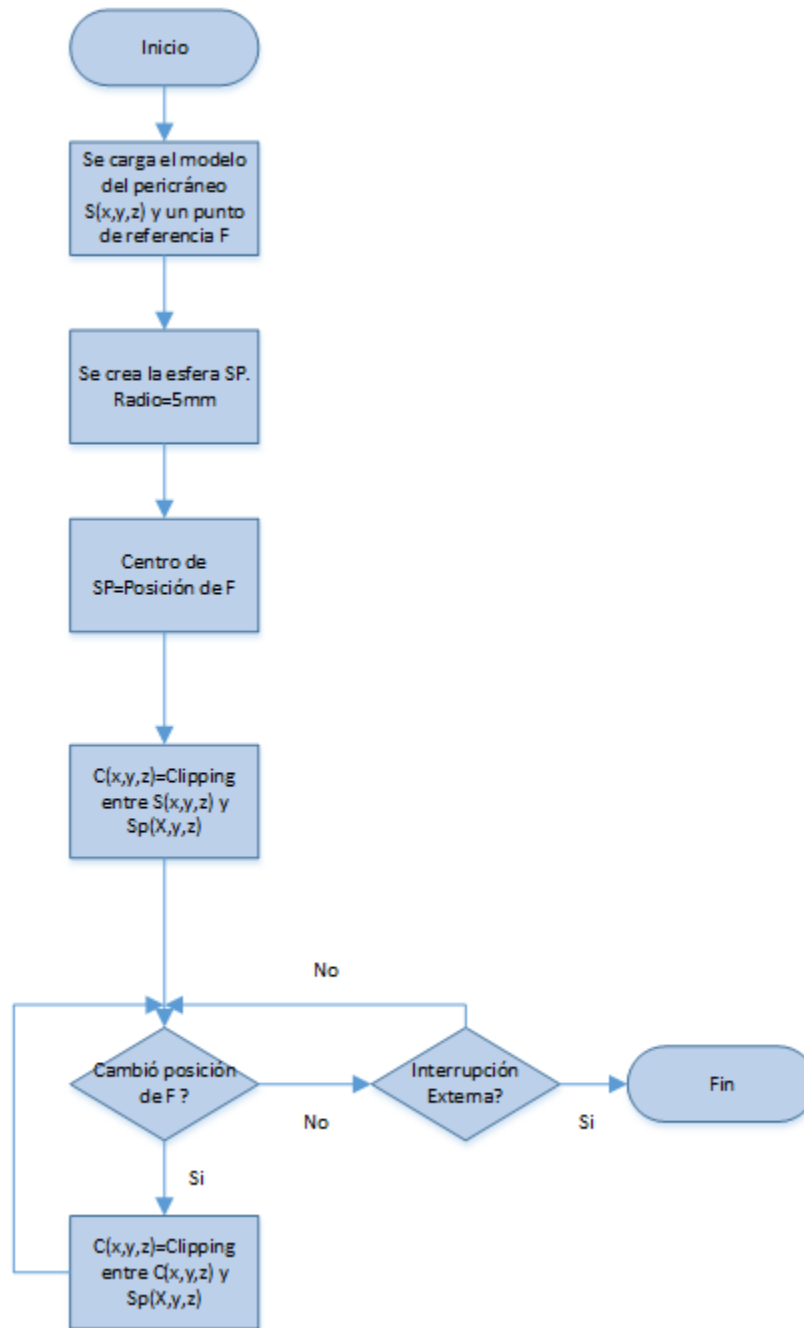


Figura 18. Algoritmo de craneotomía virtual.

El método implementado para llevar a cabo la simulación de craneotomías en representaciones tridimensionales, fue la resección virtual por borrado, que utiliza formas 3D como “gomas de borrar” que eliminan los tejidos seleccionados [29].

El objetivo del método desarrollado es utilizar las formas geométricas disponibles en la librería VTK e implementar un algoritmo de clipping, que excluye ciertas formas geométricas de la visualización, a partir de la intersección de dicha forma geométrica con un modelo del pericráneo de un paciente; este algoritmo se implementa de forma continua y dinámica, de forma que el tejido intersectado por la forma geométrica en el modelo desaparece.

La forma geométrica seleccionada fue una esfera de radio 5 mm, se selecciona este tamaño con el fin de que la acción de borrar sea más precisa.

El modelo a utilizar será el pericráneo reconstruido volumétricamente del paciente a partir de imágenes de resonancia magnética o tomografía, con el algoritmo de cubos marchantes. También se realizará un modelo de un cono (radio 10 mm, altura 50 mm) que simule la herramienta de craneotomía (Figura 19), la posición de la esfera está asociada a la punta del cono, así, cuando la punta del cono se intersecte con el modelo del pericráneo se borrará parte del tejido de éste.

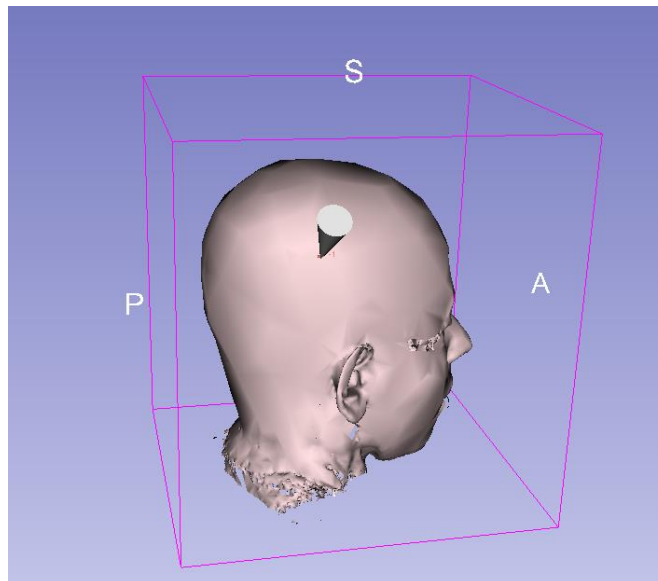


Figura 19. Reconstrucción volumétrica del pericráneo y representación de la herramienta de craneotomía simulada con un modelo de cono.

Inicialmente se cargan las imágenes del pericráneo del paciente deseado reconstruidas volumétricamente, $S(x,y,z)$, y se crea un marcador de referencia F , cuya posición se asocia a un periférico de entrada y permite moverlo. El periférico de entrada realizará el papel de instrumento de craneotomía, puede ser un escalpelo o taladro virtual y estará representado en la escena por un cono.

Para simular el objeto que realizará el borrado se crea una esfera pequeña (radio 5 mm) $SP(x,y,z)$ cuyas coordenadas del centro estarán asociadas con la posición del marcador de referencia F . La esfera $SP(x,y,z)$ se utiliza para realizar un clipping entre ésta y la imagen 3D reconstruida $S(x,y,z)$.

El objeto obtenido del clipping será la imagen del pericráneo con un pequeño orificio $C(x,y,z)$. Cada vez que el marcador de referencia cambie de posición, se crea una nueva esfera, se le asocia la nueva posición y se realiza el clipping con el objeto obtenido del clipping anterior, así se logra la apariencia de borrado en el tejido seleccionado.

Una ventaja importante del algoritmo es no tener que hacer uso de un timer (usualmente se utiliza para realizar procesos repetitivos en este tipo de aplicaciones) para realizar el clipping entre cada una de las esferas y el modelo del pericráneo, basta con agregar un observador al marcador de referencia, de esta forma cada vez que el marcador de referencia cambie de posición, se puede generar una nueva acción en la escena, de esta forma se mejora la capacidad de cómputo y la craneotomía virtual se visualiza de forma más fluida.

Una vez se implementa el algoritmo de craneotomía virtual sobre el modelo reconstruido de un pericráneo, se obtiene la vista del interior (Figura 20).

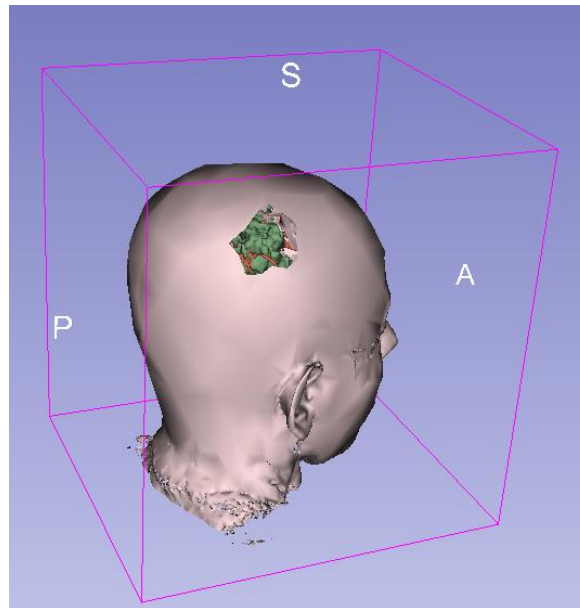


Figura 20. Vista obtenida después de realizar la craneotomía virtual.

En la Figura 21 se observa la aplicación de la craneotomía virtual en un paciente con una lesión tumoral en el hemisferio derecho, se puede observar posterior a la realización de la craneotomía la relación entre el tumor y el tracto cortico espinal. Este tipo de exploración de datos volumétricos permite al usuario tomar decisiones más acertadas acerca del camino a seguir en un procedimiento in vivo.

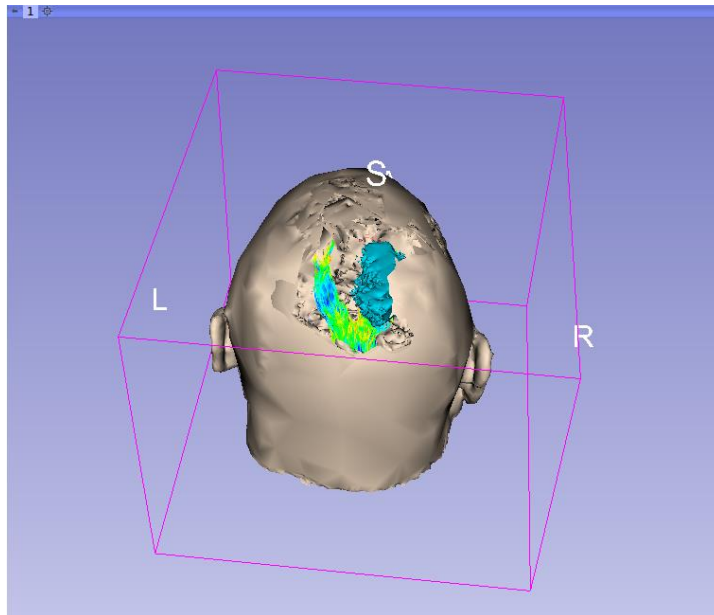


Figura 21. Craneotomía virtual en paciente con lesión tumoral hemisferio derecho.

En el Anexo 1 se encuentra el código fuente en *Python* del algoritmo de craneotomía virtual.

3.6 DISCUSIÓN

Uno de los principales objetivos de la planificación de las intervenciones es determinar las rutas más seguras para la cirugía. Estas rutas tienen que tratar de mantener intactas las estructuras importantes tales como los vasos sanguíneos, los tractos de materia blanca y las principales áreas funcionales. El algoritmo desarrollado está diseñado para realizar una craneotomía virtual en 3D Slicer, utilizando una técnica de resección virtual mediante el borrado, que permite visualizar el camino de la intervención desde el sitio de acceso, facilitando y reduciendo la abstracción de la relación de la trayectoria, el objetivo, y sus estructuras adyacentes, lo que permite al cirujano o residente realizar una mejor planificación para cualquier procedimiento que incluya una craneotomía.

En el medio se han reportado algunos sistemas para la exploración de datos volumétricos, estos sistemas incluyen herramientas para visualizar las regiones de interés utilizando herramientas de recorte que permiten el corte regular (esférico, cúbico, cilíndrico y elipsoidal) de la anatomía del cerebro [21][22]. La diferencia del algoritmo desarrollado sobre estos sistemas radica en que los cortes que se realizan con el algoritmo pueden tener cualquier geometría, incluso formas irregulares, lo que lo acerca más a una craneotomía real.

El algoritmo tiene la ventaja que se desarrolló en un lenguaje ampliamente utilizado (Python) para una herramienta de código abierto (3D Slicer), por lo que es accesible para uso educativo.

Además el algoritmo puede ser implementado en sistemas de planificación de neurocirugía, el único requisito es que la reconstrucción volumétrica del pericráneo o del cráneo se haga a partir de un algoritmo de visualización indirecta (cubos marchantes). Las estructuras contenidas al interior del pericráneo pueden variar según la necesidad y la disponibilidad de otros tipos de imágenes.

En un trabajo futuro se mejorará el algoritmo de forma tal que el clipping se pueda hacer con una sola esfera, sin necesidad de crear una nueva cada vez que se mueva el marcador de referencia, pero conservando la propiedad de obtener el tejido borrado, así se puede mejorar el rendimiento del algoritmo, además se incluirá un algoritmo de detección de colisiones que funcione de forma fluida y rápida, para poder llevar a cabo la simulación de una craneotomía virtual más parecida a la craneotomía real, también se incluirán funcionalidades que permitan medir distancias y asociarlas con medidas reales del paciente, para que los especialistas puedan utilizar esta herramienta en la planeación de neurocirugías y puedan tener una referencia más precisa de donde se debe realizar el corte. También se desarrollará un algoritmo de craneotomía virtual a partir de reconstrucciones volumétricas generadas con algoritmos de visualización directa.

4 IMPLEMENTACION DE UN PROTOCOLO PARA LA MANIPULACIÓN DEL ENTORNO POR MEDIO DE UN PERIFÉRICO DE ENTRADA QUE PERMITE RELACIONAR COORDENADAS EN EL ESPACIO FÍSICO CON EL ESPACIO VIRTUAL.

En este capítulo se describe algunos de los dispositivos existentes para la interacción representaciones tridimensionales de imágenes médicas y además contiene la información sobre la implementación de un protocolo para la manipulación del entorno por medio de un periférico de entrada que permite relacionar coordenadas en el espacio físico con el espacio virtual.

En la Figura 22 se resumen los pasos metodológicos seguidos para el desarrollo del tercer objetivo específico.

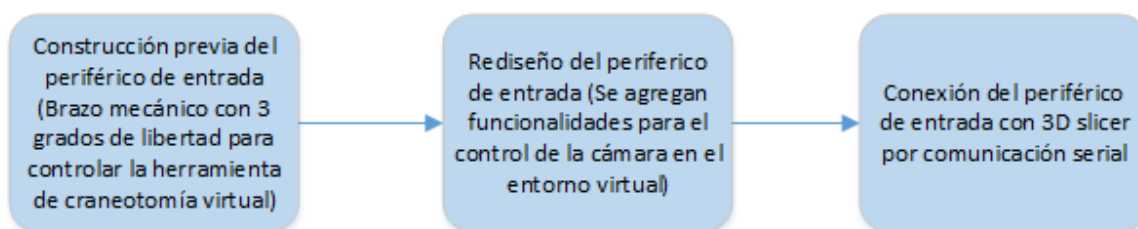


Figura 22. Metodología seguida para la manipulación del entorno por medio de un periférico de entrada.

4.1 PERIFÉRICOS DE ENTRADA UTILIZADOS EN SIMULADORES DE CIRUGÍA.

En el desarrollo del trabajo de investigación se definieron los métodos más adecuados de interacción con la aplicación, utilizando un periférico de entrada que reemplaza el mouse y permite relacionar coordenadas en el espacio físico con el espacio virtual, para de esta forma lograr una interacción más intuitiva y una simulación lo más cercanamente posible a la realidad.

El uso de componentes de hardware disponibles en el mercado fue una opción que se evaluó, debido a las innovaciones en electrónica durante los últimos años, que involucran dispositivos de costos moderados y alta precisión hacen que sea una alternativa razonable para la interacción con la aplicación.

Algunos de los dispositivos disponibles en el mercado son:

Kinect: Es un controlador de videojuegos que permite interactuar con la consola sin necesidad de tener contacto físico con un controlador tradicional, este dispositivo reconoce comandos de voz, gestos, objetos e imágenes. Se puede implementar en simuladores de cirugía gracias a que reconoce los movimientos de las manos del usuario en el espacio 3D utilizando una combinación de cámaras de espectro infrarrojo y visible. Este dispositivo tiene un costo aproximado de 150 dólares [62].

Leap Motion: Es un dispositivo de detección de movimiento, que permite la interacción humano-computador, el cual fue creado para mejorar la interacción 3D, ya que esta no es tan intuitiva cuando se utiliza el teclado y el mouse [63]. El dispositivo utiliza dos cámaras y tres LEDs infrarrojos, y registra un área semiesférica, a una distancia de aproximadamente 1 metro. Está diseñado para un seguimiento de los dedos u objetos similares que se cruzan en el área de registro, con una precisión espacial de 0,01 mm [63]. Tiene un costo aproximado de 80 dólares.

SpaceNavigator: Es un mouse 3D que permite una navegación tridimensional, tiene un sensor óptico de precisión avanzado con 6 grados de libertad. El modo de trabajo es bimanual, con una mano se maneja el mouse 3D para posicionar el modelo o navegar por el entorno, mientras la otra mano usa el ratón convencional simultáneamente, para seleccionar, crear o editar. Tiene un costo aproximado de 90 dólares [64].

NDI Aurora tracking: El sistema de seguimiento NDI Aurora® es una tecnología de navegación diseñada específicamente para aplicaciones médicas. Basada en la tecnología electromagnética, sin requisitos de línea de vista, Aurora sigue sensores miniaturizados diseñados para su integración en herramientas e instrumentos quirúrgicos, tales como agujas, catéteres, sondas y microscopios. La capacidad de colocar sensores de Aurora en la punta de un instrumento permite la localización de los instrumentos flexibles dentro del cuerpo [65].

4.2 PERIFÉRICO DE ENTRADA ELEGIDO

Para poder aplicar los métodos más adecuados de interacción con la aplicación desarrollada, se decidió utilizar un periférico de entrada que reemplace el mouse y que permita relacionar coordenadas del espacio físico con el espacio virtual, para de esta forma lograr una interacción más intuitiva y una simulación lo más cercanamente posible a la realidad.

El periférico de entrada elegido fue desarrollado en conjunto con los estudiantes Andrés Felipe Botero y David Esteban Higueta, quienes hicieron el desarrollo electrónico para obtener el grado de Bioingenieros [66]. Se fabricó un dispositivo mecánico el cual consta de dos eslabones y tres articulaciones móviles, que permiten un movimiento relativo entre las partes de los eslabones, proporcionándole al dispositivo 3 grados de libertad. El dispositivo permite mediante su manipulación,

capturar la posición de un actuador final para posteriormente enviar la información al software de visualización 3D-Slicer.

El brazo mecánico cuenta con tres partes móviles que le permiten el movimiento con respecto a los tres ejes coordenados. En las articulaciones del brazo se ubican los sensores (3 potenciómetros) que registran el desplazamiento en grados de cada articulación (Figura 23).

Transformaciones para el brazo mecánico diseñado

La posición final del actuador del brazo mecánico diseñado se obtuvo mediante la siguiente secuencia de transformaciones y translaciones (Figura 23).

Las transformaciones quedan definidas de la siguiente forma:

Primera transformación: rotación en un ángulo α sobre el eje Z.

Segunda transformación: rotación en un ángulo γ sobre el eje X.

Tercera transformación: traslación una distancia L a lo largo del eje Z.

Cuarta transformación: rotación en un ángulo β sobre el eje X.

La ecuación (1) muestra la matriz de transformación resultante, la cual es obtenida a partir de las transformaciones anteriormente descritas.

$$T(x, y, z) = R(z, \alpha) \cdot R(x, \gamma) \cdot T(z, L) \cdot R(x, \beta) \quad (1)$$

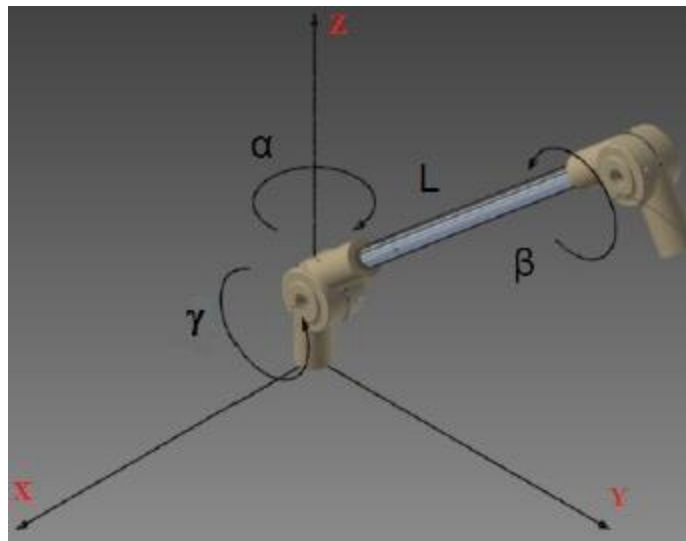


Figura 23. Transformaciones aplicadas al brazo mecánico.

4.3 REDISEÑO DEL PERIFÉRICO DE ENTRADA

El periférico de entrada elegido, mostró un funcionamiento adecuado y cumple con los requisitos definidos para su uso en el simulador, aun así se decidió agregar nuevas funcionalidades para lograr una mejor interacción con el entorno de simulación.

Se agregaron 3 potenciómetros independientes al brazo mecánico que permiten controlar la posición de la cámara virtual para facilitar la manipulación de las imágenes y un interruptor que permite iniciar o detener la simulación de la craneotomía virtual.

Mediante los seis potenciómetros, tres ubicados en las articulaciones móviles del brazo mecánico y tres potenciómetros independientes, se detecta el cambio en el valor de voltaje que tiene cada resistor, cada uno de los valores de voltajes generados son tratados por cada uno de los seis puertos de conversión análogo-digital del microcontrolador Arduino UNO, además se habilita un puerto digital del microcontrolador que permita almacenar el estado del interruptor (0 o 1).

En la Figura 24 se observa el esquema del circuito, la conexión del brazo mecánico, los potenciómetros independientes a la tarjeta del microcontrolador Arduino UNO. El pin 5V y GND suministran el voltaje a cada uno de los potenciómetros y el interruptor, los pines de la entrada analógica A0...A5, reciben el voltaje variable de cada uno de los potenciómetros y lo convierten a un valor digital el pin 7 recibe un valor digital 0 o 1 al abrir o cerrar el interruptor que controla el inicio de la craneotomía.

Se observa que en la tarjeta están incluidos los componentes encargados de la transmisión y recepción de datos vía serial *TX* y *RX*, sin necesidad de agregar módulos adicionales.

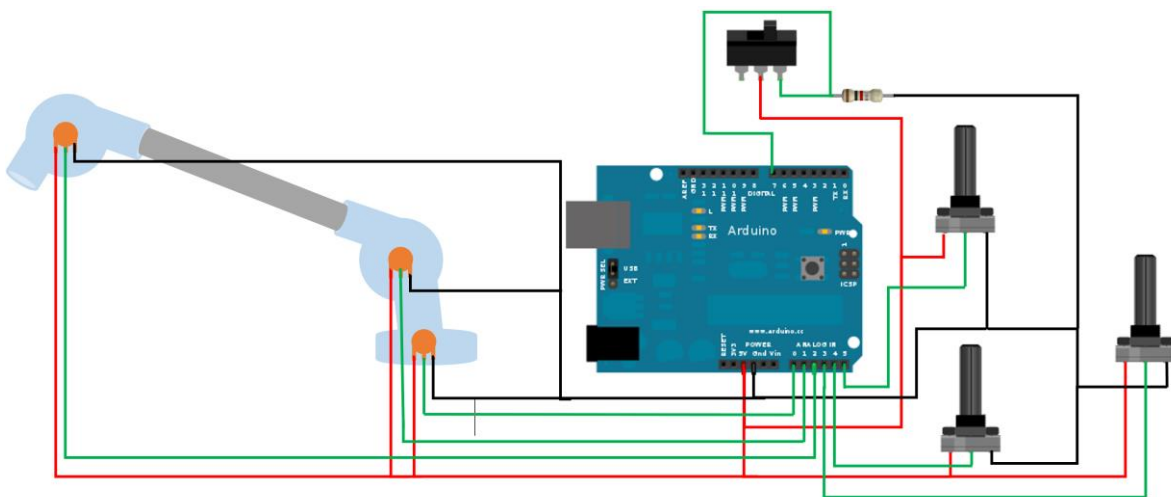


Figura 24. Esquema del circuito desarrollado.

Toda la información recolectada por el microcontrolador es enviada vía comunicación serial a un computador. El dato es adquirido por el software de visualización, para luego convertir este cambio de voltaje en el ángulo de desplazamiento o de rotación de cada articulación, de rotación en los potenciómetros independientes y el estado del interruptor. El siguiente diagrama

muestra el flujo de información hasta llegar al programa de visualización 3D-Slicer (Figura 25).

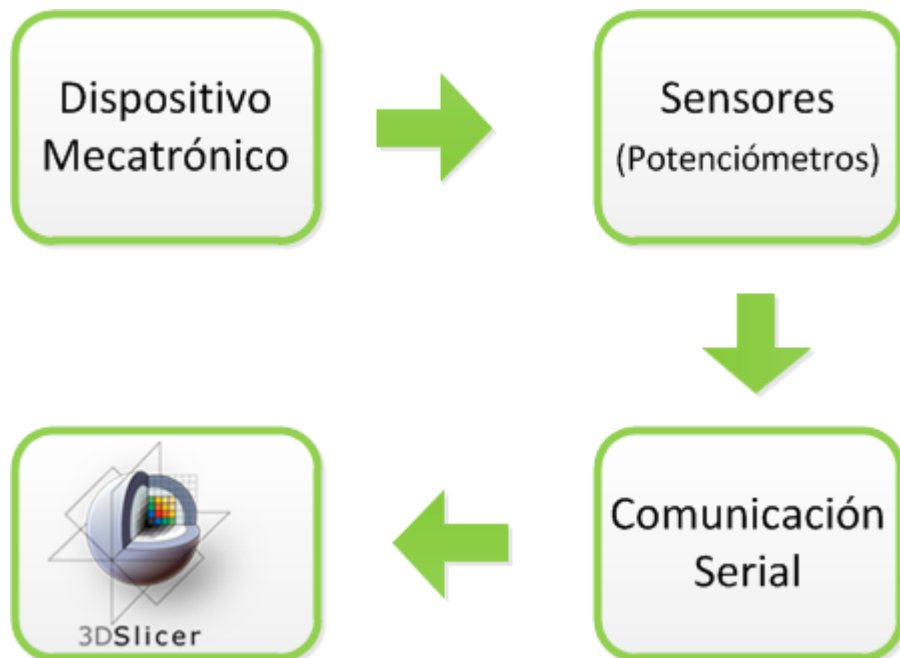


Figura 25. Flujo de información para la conexión del dispositivo mecatrónico con 3D Slicer.

La información entregada por los potenciómetros y el interruptor se trasmite vía comunicación serial al software 3D Slicer. El protocolo de comunicación serial, permite el envío de un bit de información de manera secuencial, esto es, un bit a la vez y a un ritmo acordado entre el emisor y el receptor. Aunque este tipo de comunicación es más lenta que la comunicación en paralelo, el envío de información se hace de manera confiable y su implementación es sencilla y eficiente.

Para que los comunicar el sistema de adquisición con el software de visualización se deben configurar las siguientes características:

Velocidad de transmisión (Baud rates): Indica el número de bits por segundo transmitidos, sus unidades son los baudios y para la transmisión se usó una velocidad de 9600 baudios.

Puerto de comunicación: Éste puerto es el que se habilita para la comunicación entre dispositivos, su nomenclatura depende del sistema operativo en el que se trabaje, el código desarrollado permite el reconocimiento automático del puerto habilitado.

Bits de parada: Los bits de parada ayudan a verificar cuando inicia o termina una secuencia o paquete de información. La información que se quiere transmitir son los cambios en voltaje de los tres potenciómetros ubicados en las articulaciones del brazo mecánico (Dato1, Dato2 y Dato3), de los tres potenciómetros independientes

(Dato4, Dato5 y Dato6) y el estado del interruptor (Dato7). Para cada dato enviado por cada potenciómetro, se envía un bit de arranque que indica el inicio de la secuencia.

Los datos enviados por cada potenciómetro y el interruptor, están precedidos cada uno por un bit de arranque. La Figura 26 ilustra el protocolo implementado para el envío de la información.

0	Dato1	1	Dato2	2	Dato3	3	Dato4	4	Dato5	5	Dato6	6	Dato7
----------	--------------	----------	--------------	----------	--------------	----------	--------------	----------	--------------	----------	--------------	----------	--------------

Figura 26. Paquete de información enviada al computador. Los números 0,1,2,3,4,5 y 6 son bits de arranque.

El software 3D Slicer requiere la instalación de librerías adicionales para reconocer el puerto serial y poder recibir la información utilizando este método de comunicación, la librería instalada se llama pyserial, es específica para Python, es de fácil acceso y de uso libre.

En la Figura 27 Se resume en un diagrama de flujo el tratamiento de la información enviada desde el sistema de adquisición por el software 3D Slicer para decidir que segmento controlar. Los Datos 1, 2 y 3 están asociados al brazo mecánico el cual controla en la escena la herramienta de craneotomía virtual que está representada por el cono en la escena (Figura 19), los datos 4 5 y 6, están asociados a los potenciómetro independientes y se encargan del control de la cámara de la escena y el dato 7 está asociado al interruptor que permite decidir si se desea realizar o no la craneotomía.

En el Anexo 2, se encuentra el código fuente del algoritmo para la recepción de los datos vía serial en 3D Slicer y la transformación de los mismos.

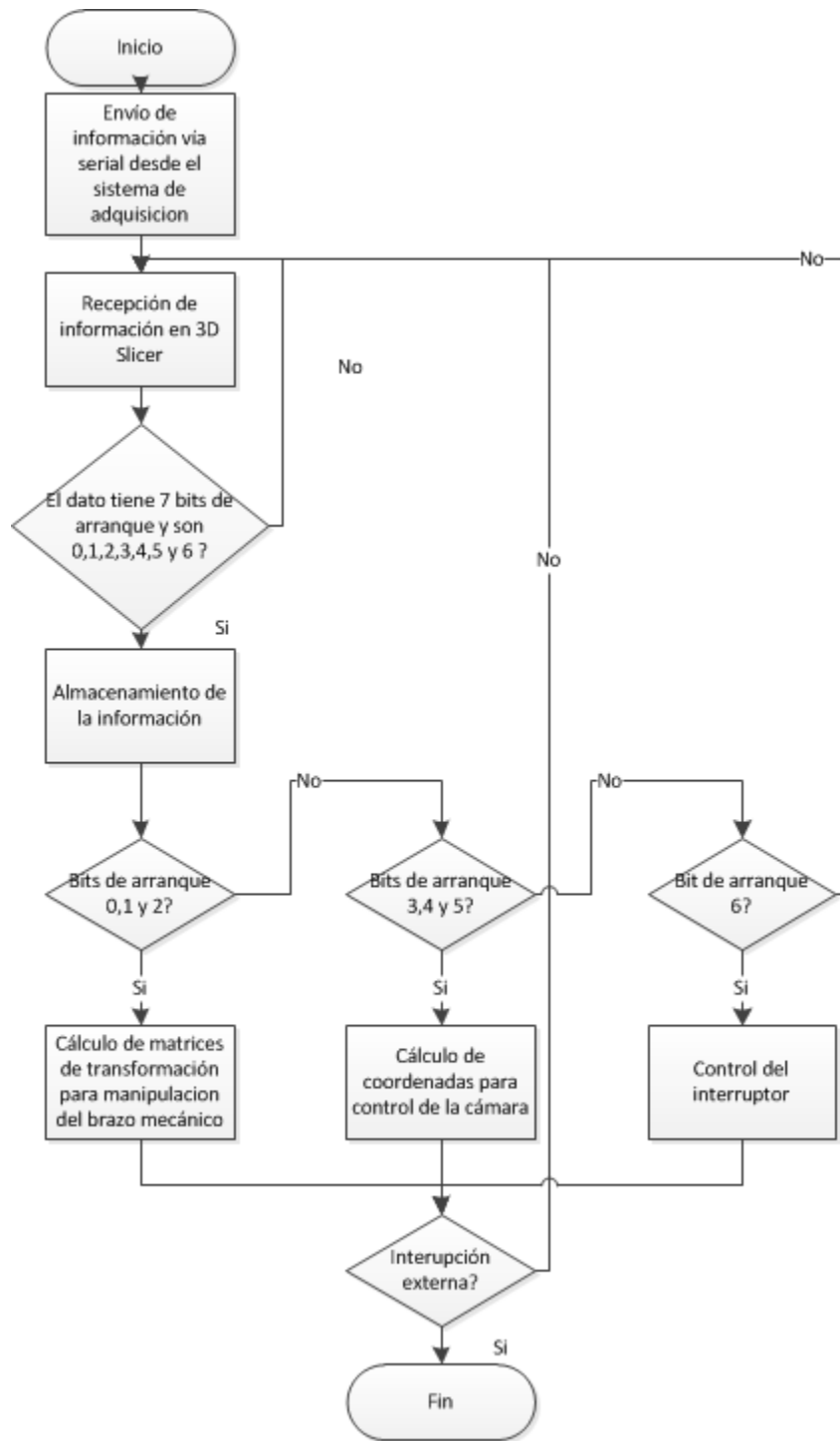


Figura 27. Diagrama de flujo que resume los pasos a seguir para la adquisición de la información desde el periférico de entrada hacia el software de visualización.

Transformaciones para el control de la cámara

En 3D Slicer, el control de la cámara de la escena que incluye la reconstrucción tridimensional de los objetos se puede hacer desde la interfaz gráfica de usuario o con el mouse. La cámara puede rotar alrededor del eje de guiñada (*yaw* en inglés), de cabeceo (*pitch*) y de alabeo (*roll*) y además se pueden hacer acercamientos y alejamientos (*Zoom in/out*), sin embargo la manipulación desde el mouse de estas acciones no siempre son naturales al usuario y en la interfaz gráfica no son accesibles a simple vista. Por esta razón se decidió realizar el control de la cámara utilizando la librería VTK, variando su posición desde el periférico de entrada.

La posición de la cámara en la escena de 3D Slicer se encuentra en coordenadas cartesianas, como los potenciómetros utilizados son de tipo rotatorio y para que el movimiento realizado por el usuario corresponda al movimiento expresado en la escena de 3D Slicer, se decidió que el control de la cámara debía hacerse en coordenadas esféricas, así el primer potenciómetro controla el ángulo φ , el segundo potenciómetro controla el ángulo θ , y el tercer potenciómetro controla el radio r , lo cual se traduce en un control del zoom de la escena (Figura 28).

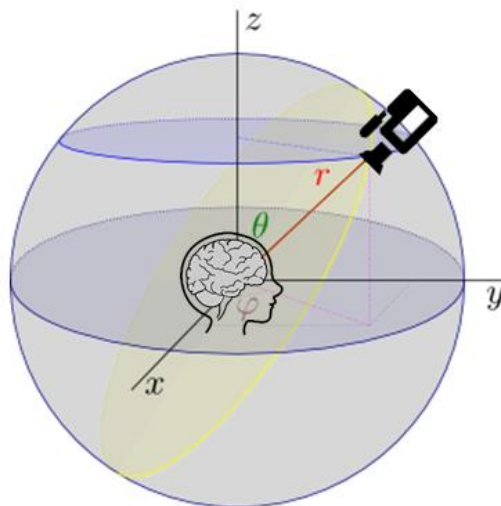


Figura 28. Operación de los potenciómetros independientes y su relación con la cámara de la escena.

En la Figura 29, se observa el prototipo final del periférico de entrada que incluye el brazo mecánico para el control de la herramienta de craneotomía virtual y los potenciómetros independientes que se encargan del control de la cámara.



Figura 29. Periférico de entrada.

4.4 DISCUSION

En 3D-Slicer controlar funciones como rotar, girar y acercar o alejar las imágenes en escena, hacen necesario utilizar el mouse y el teclado. Además para llevar a cabo funciones adicionales como el control de la herramienta de craneotomía virtual se hace necesario también el uso del mouse o usar la interfaz gráfica de usuario (GUI).

En respuesta al reto de hacer que la tecnología sea lo más natural posible para el usuario, el dispositivo mecatrónico desarrollado permite controlar diferentes funciones de la escena y manipular la herramienta de craneotomía, sin necesidad de utilizar otro dispositivo o la GUI, lo que permite realizar movimientos en un espacio tridimensional real y reproducirlos en la escena virtual, proporcionando una interacción más intuitiva y simple

La elección del periférico de entrada se basó en la facilidad que proporciona diseñar un dispositivo desde el principio para adecuarlo a las necesidades específicas requeridas. El periférico de entrada elegido se propuso como un trabajo de grado de pregrado, que consistió en la construcción de un dispositivo de posicionamiento mecatrónico con conexión al 3D Slicer. El dispositivo en sus orígenes fue construido con un microcontrolador *MC9S08JM60* de Freescale Semiconductors, que utiliza el entorno integrado de desarrollo *Codewarrior* para su programación, dicho entorno necesita una licencia comercial; además el microcontrolador necesita también el módulo de comunicación serial FT232RL lo que aumenta los costos del dispositivo, así como su tamaño. Por esta razón se decidió utilizar el microcontrolador *Arduino*

Uno, cuyo software es de código abierto y la transmisión serial está incluida en la misma tarjeta lo que disminuye el espacio necesario.

Al dispositivo se agregaron nuevas funciones, como controlar la cámara a partir de potenciómetros rotatorios, lo cual lo acerca más al objetivo de su implementación que es lograr una interacción más intuitiva con el entorno de simulación. Además como el algoritmo de craneotomía virtual es computacionalmente costoso, se agregó un interruptor para llevar a cabo la craneotomía virtual solamente cuando el usuario así lo decida.

El dispositivo rediseñado junto con el algoritmo de craneotomía virtual fueron presentados en la conferencia "*Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*". En el trabajo "Algorithm for simulation of craniotomies assisted by peripheral for 3D virtual navigation".

5 IMPLEMENTACIÓN DE LOS ALGORITMOS DESARROLLADOS EN UNA APLICACIÓN DE USO LIBRE.

5.1 DESCRIPCIÓN DEL SIMULADOR DESARROLLADO

Los algoritmos desarrollados se unificaron para construir una aplicación que permita llevar a cabo la simulación de craneotomías. La aplicación se desarrolló en 3D Slicer, el cual sigue un paradigma modular en la entrega al usuario de tareas de funciones específicas. Cada módulo es una unidad funcional completa diseñada para servir a una tarea específica. Uno de los objetivos principales de 3D Slicer es proporcionar a los ingenieros biomédicos, desarrolladores y científicos los componentes esenciales para la creación rápida de prototipos y el desarrollo eficiente de las herramientas de análisis de imágenes biomédicas. Desde el principio, Slicer ha sido diseñado para ser abierto y extensible [8].

El módulo desarrollado para la simulación de craneotomías en 3D Slicer se compone de tres pasos (Figura 30), el primer paso permite la selección del caso con el que se desea trabajar, hay 7 casos disponibles y cada uno incluye la reconstrucción volumétrica de las estructuras de interés asociadas a cada paciente. En la Tabla 4, se muestran las estructuras disponibles para cada uno de los casos de pacientes presentados.

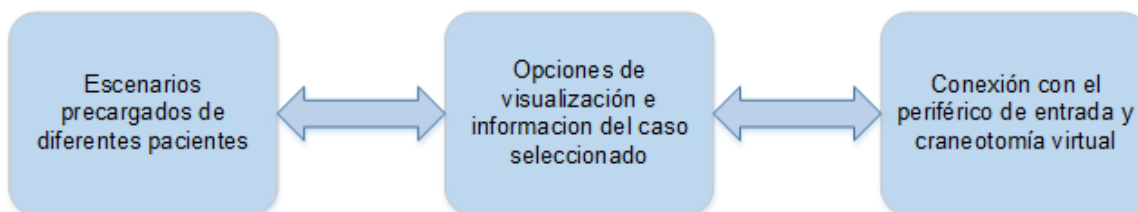


Figura 30. Flujo de información en el simulador desarrollado.

Tabla 4. Reconstrucciones volumétricas disponibles para cada caso en el simulador

Caso / Estructura	Pericráneo	Cerebro	Lesión Tumoral	Tracto Córtricoespinal	Árbol Venoso
1	X	X	X		X
2	X	X	X	X	
3	X	X	X	X	
4	X	X	X	X	
5	X	X	X	X	
6	X	X	X		
7	X	X	X		

El segundo paso del simulador permite definir si el caso seleccionado previamente es realmente el caso con el que se desea trabajar, puesto que se puede elegir una opción para informarse sobre las estructuras disponibles, además se puede ocultar el pericráneo y/o el cerebro para llevar a cabo una toma de decisión correcta.

En el tercer paso del simulador se realiza la craneotomía virtual. En este paso se realiza la conexión del 3D Slicer con el periférico de entrada para la recepción de datos vía serial, además se incluye el algoritmo de resección virtual desarrollado para llevar a cabo la craneotomía virtual en la reconstrucción volumétrica del pericráneo del caso elegido en el primer paso.

A continuación se describe detalladamente cada una de las funciones disponibles en el simulador.

Al iniciar el módulo desarrollado, llamado *Simulador* se despliega una lista con todos los escenarios de diferentes pacientes precargados para realizar la simulación (Figura 31).

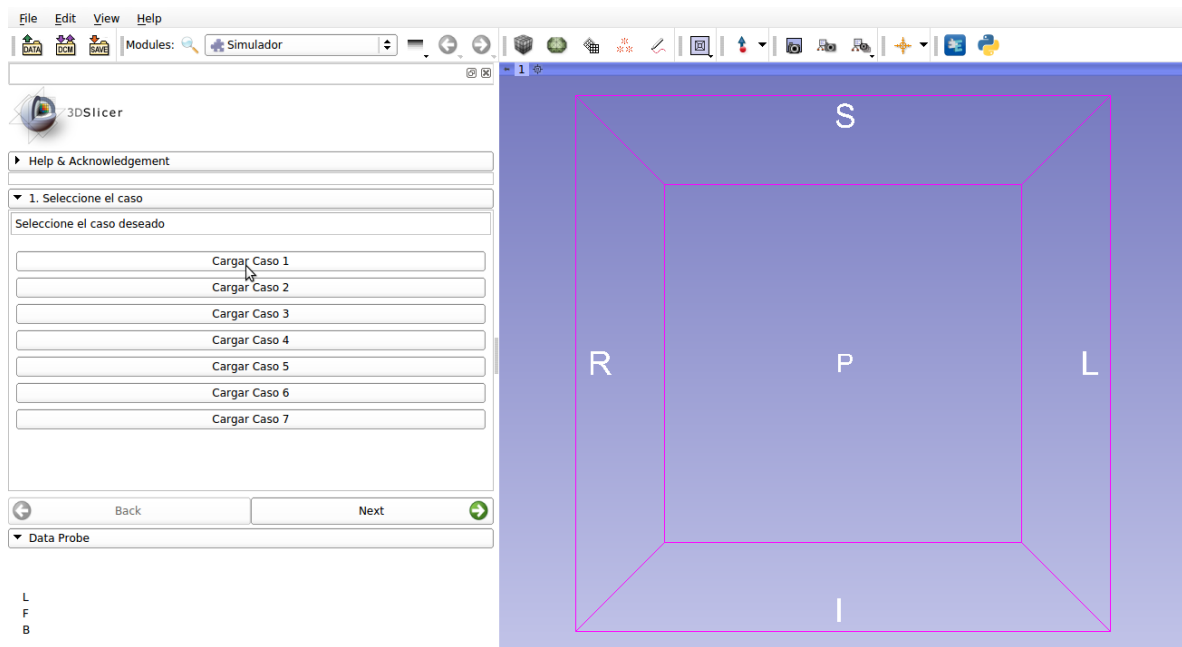


Figura 31. Simulador: Paso 1, Selección del caso.

Una vez se selecciona el caso deseado se carga en la escena de 3D Slicer y se debe hacer clic en *Next* para continuar con el paso 2 (Figura 32).

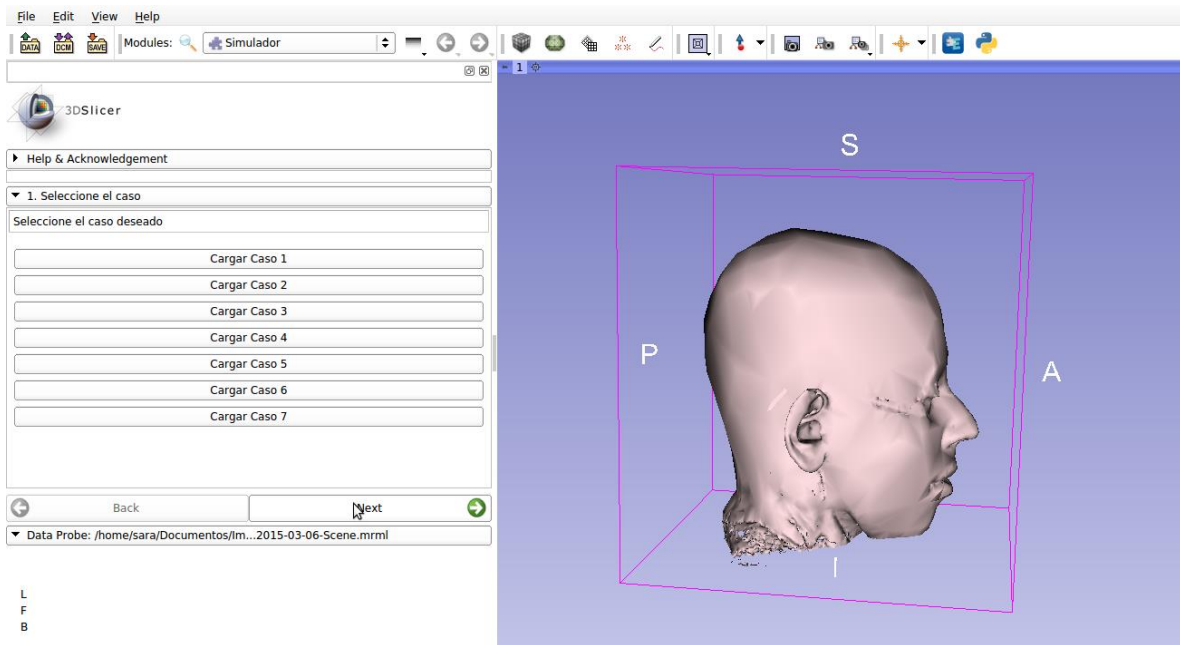


Figura 32. Simulador: Paso 1, Caso cargado en la escena.

En el paso 2, se puede observar la información de las estructuras disponibles en el caso seleccionado, y se presentan algunas opciones de visualización (Figura 33).

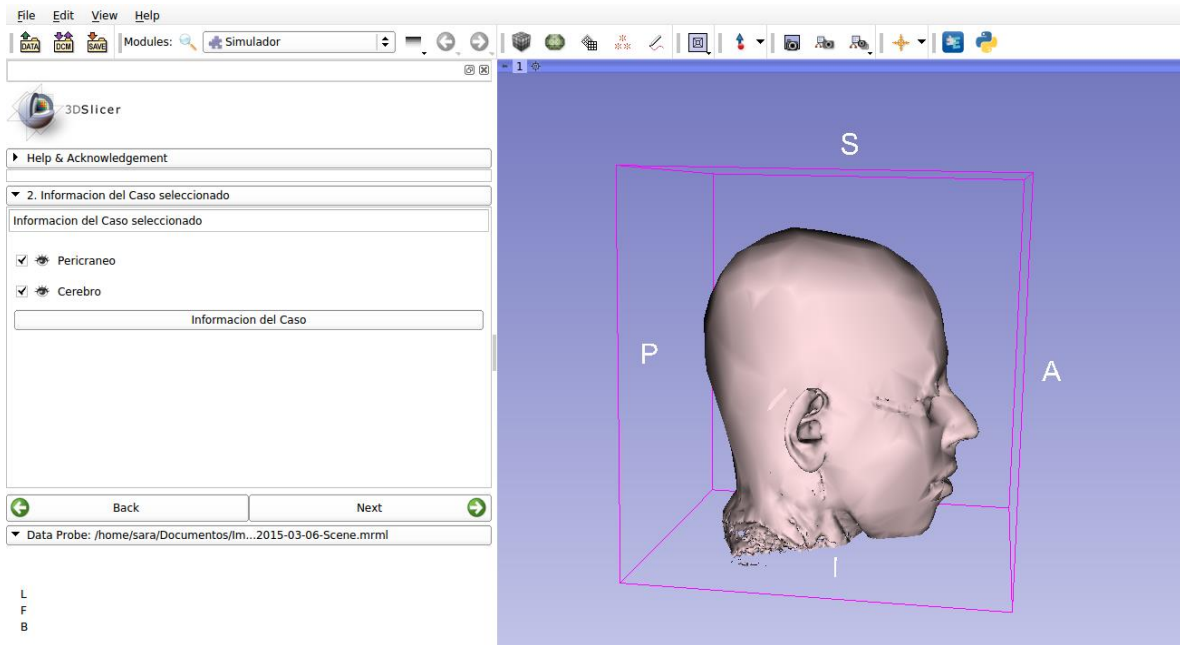


Figura 33. Simulador: Paso 2, Información del caso seleccionado.

Al elegir la opción *Pericráneo* se puede activar o desactivar la visualización de dicha estructura (Figura 34).

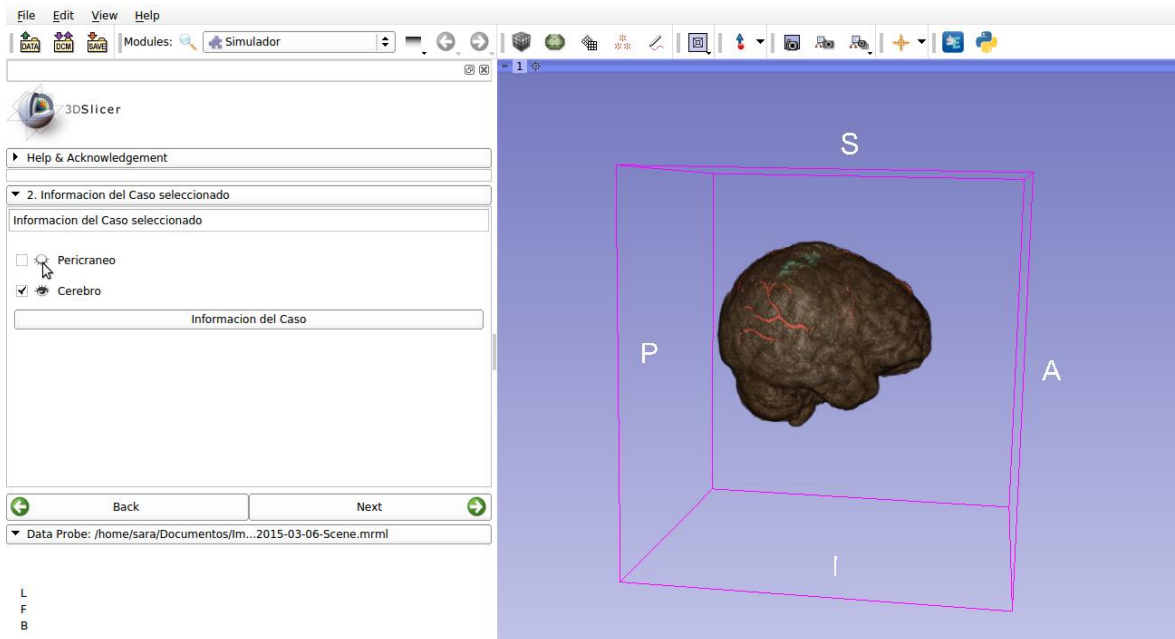


Figura 34. Simulador: Paso 2, Opción de visualización del pericráneo.

La opción *Cerebro* permite también activar o desactivar la reconstrucción volumétrica del cerebro para identificar las estructuras internas y poder tomar la decisión si ese es el caso con el que se desea trabajar (Figura 35).

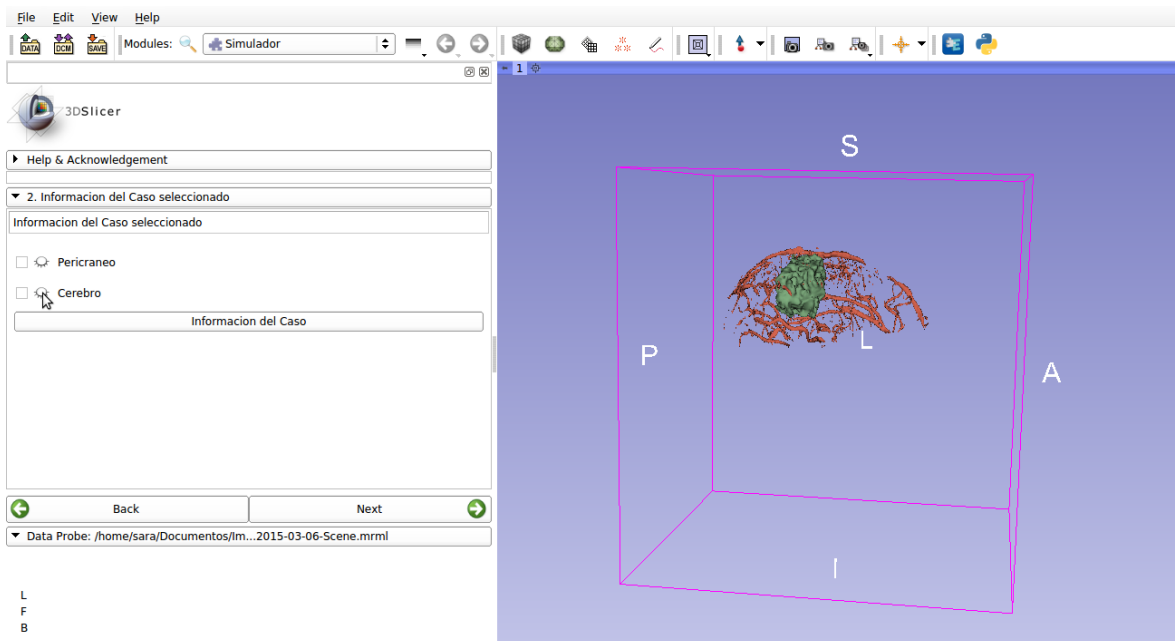


Figura 35. Simulador: Paso 2, opción de visualización del cerebro.

El botón *Información del Caso* despliega una lista de las estructuras disponibles en el caso seleccionado (Figura 36).

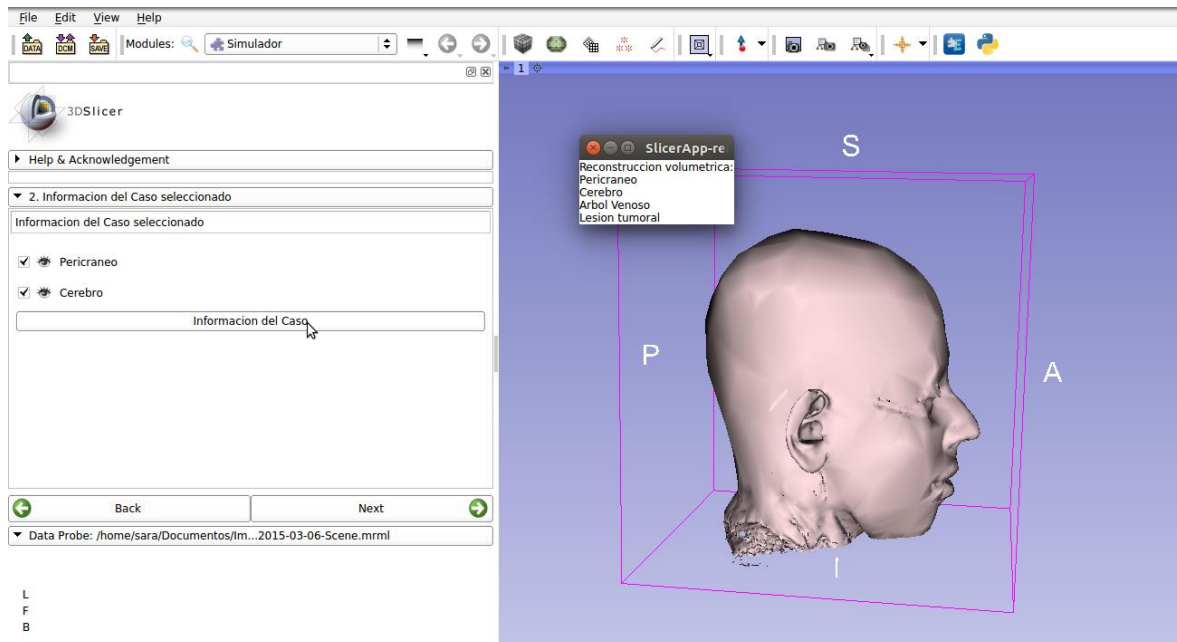


Figura 36. Simulador: Paso 2, Lista de estructuras disponibles.

El usuario puede hacer clic en el botón *Back* si el caso seleccionado no es el caso con el que se desea realizar la craneotomía, regresando al paso 1 para seleccionar otro caso, o puede hacer clic en el botón *Next* si desea iniciar la simulación de craneotomía con el caso seleccionado.

En el paso 3 se puede iniciar la simulación de la craneotomía, al hacer clic en el botón *Iniciar*. Para iniciar la simulación es necesario que el periférico de entrada esté conectado, de no ser así el programa lo recordará con un mensaje (Figura 37).

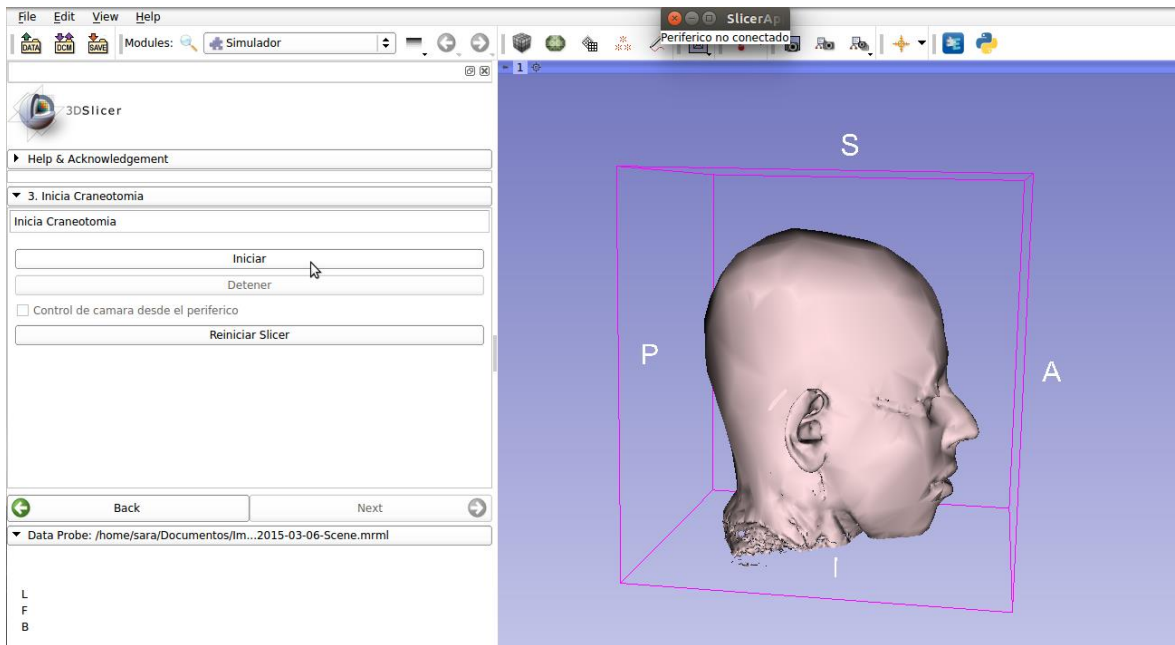


Figura 37. Simulador: Paso 3, Iniciar craneotomía, periférico no conectado.

Si el periférico se encuentra conectado el botón de *Iniciar* se deshabilita, se habilitan los botones *Detener* y *Control de cámara desde el periférico*, en escena aparece la herramienta de craneotomía representada por un cono e inicia la transmisión de datos desde el sistema de adquisición del periférico de entrada hacia el software 3D Slicer. Es en este momento que se puede realizar la craneotomía virtual, controlando el cono con el brazo mecatrónico del periférico de entrada y el interruptor disponible para elegir cuando realizar la resección (Figura 38).

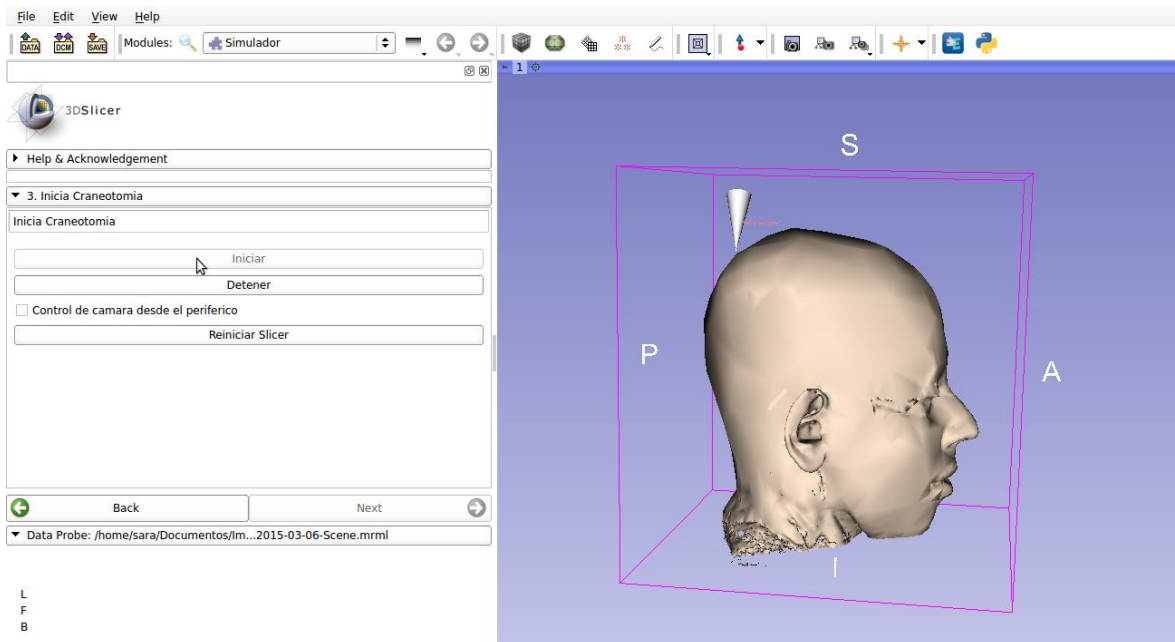


Figura 38. Simulador: Paso 3, Iniciar craneotomía, periférico conectado.

La cámara se controlará desde el periférico de entrada solo si el usuario así lo solicita desde el software activando la opción *Control de cámara desde el periférico*, una vez se selecciona esta opción la cámara del entorno se puede manipular con los potenciómetros independientes disponibles en el periférico de entrada (Figura 39).

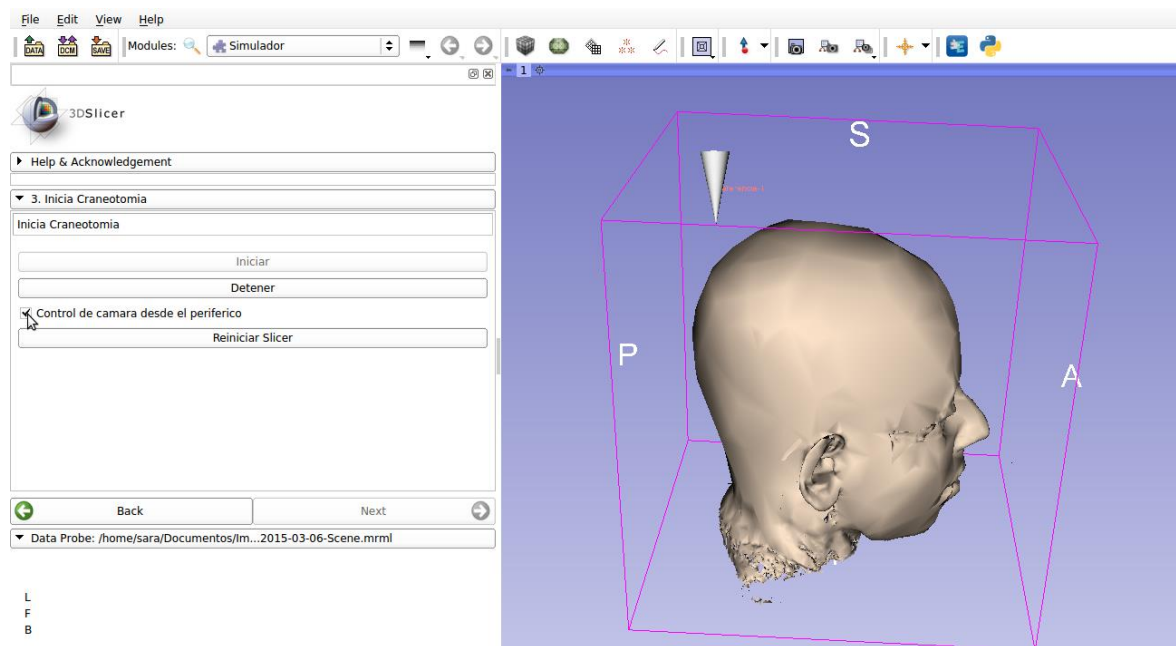


Figura 39. Simulador: Paso 3, Activar control de la cámara desde el periférico.

El botón *Detener* interrumpe el envío de información desde el periférico de entrada hacia el software 3D Slicer, y también interrumpe el algoritmo de craneotomía virtual. Al hacer clic se deshabilita este botón y la opción *Control de cámara desde el periférico* y se habilita la opción iniciar, así el usuario puede iniciar la craneotomía virtual en el momento que lo requiera (Figura 40).

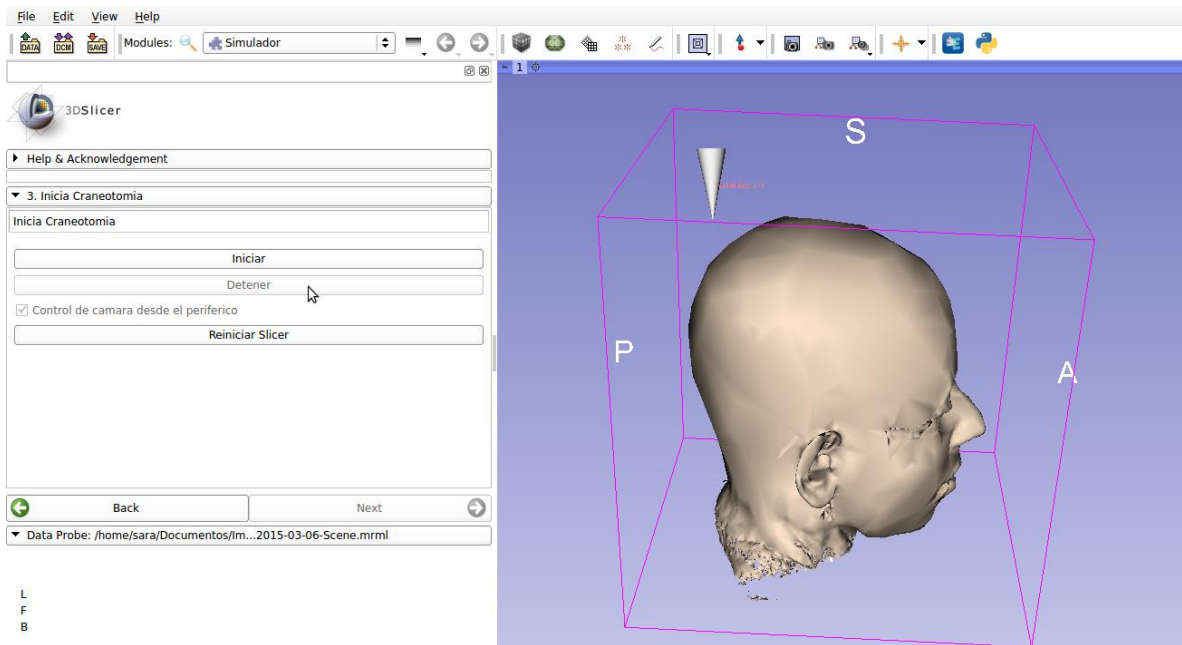


Figura 40. Simulador: Paso 3, Detener craneotomía.

En la Figura 41 se muestra la interacción en tiempo real de un usuario con el simulador, utilizando el periférico de entrada.



Figura 41. Simulador y conexión con el periférico de entrada.

5.2 DISCUSION

El simulador desarrollado se caracteriza por su fácil manejo, le presenta al usuario 7 posibles escenarios para llevar a cabo la simulación de craneotomías en alguno de ellos, además agregar nuevos escenarios solo requiere modificar una línea del código fuente, lo que hace este simulador ideal para en un futuro agregar una opción de planeación de neurocirugías.

Cada uno de los pasos del simulador tiene la opción de regresar al paso anterior, evitando que el usuario deba reiniciar el módulo si su selección no es la deseada y presenta mensajes claros cuando hay información faltante para continuar. El paso 2 presenta información sobre las estructuras disponibles, así el usuario puede decidir si esa es la patología que quiere trabajar y si las estructuras disponibles son suficientes para continuar con la simulación, este paso se puede obviar si el usuario ya conoce los escenarios disponibles y continuar directamente a la simulación de la craneotomía virtual; este paso presenta de manera sencilla las opciones de visualización disponibles sin dar lugar a confusión al usuario. El paso 3 permite iniciar la craneotomía virtual siempre y cuando el periférico de entrada se encuentre conectado y enviando los datos al simulador, en este paso se puede elegir el control de la cámara desde el periférico de entrada facilitando la visualización y el acceso a algunas zonas de interés, además de hacer zoom sin involucrar el mouse, así la simulación es más intuitiva.

Se decidió hacer el simulador en 3 pasos sencillos y con pocos botones para facilitar el manejo del mismo al usuario y que el tiempo de capacitación sea menor, además este presenta como ventaja adicional que se puede usar tanto en Windows como en Linux, así el usuario tiene la posibilidad de elegir su sistema operativo favorito según su criterio, necesidades y disponibilidad.

El entorno virtual desarrollado permite llevar a cabo la simulación de craneotomías realizando cortes irregulares del pericráneo, además se incluyen modelos tridimensionales de pacientes con lesiones tumorales en el área motora y el tracto corticoespinal, para observar la relación entre el tumor y dicho tracto, el cual está asociado a funciones motoras. Además se incluye una herramienta manual que mejora la interacción del usuario con el simulador. Todo se desarrolló en código abierto para un software libre, de modo que los costos asociados son mínimos comparados con los simuladores reportados en la literatura, obteniendo de esta forma, un simulador accesible al entorno educativo

En un trabajo futuro se propone realizar pruebas de usabilidad con residentes de neurocirugía para obtener una retroalimentación y realizar las mejoras que sean pertinentes en el simulador.

CONCLUSIONES Y TRABAJO FUTURO

Los sistemas de neurocirugía guiada por imágenes se han utilizado en un contexto clínico y educativo, por lo que en la práctica, los procedimientos de neurocirugía se han vuelto más seguros y más eficientes.

Los escenarios construidos a partir de casos reales de pacientes con alteraciones en el área motora, utilizando algoritmos tradicionales de visualización, presentan un comportamiento adecuado en la parte visual, además los objetos reconstruidos son livianos, lo que se traduce en una disminución de la carga computacional a la hora de interactuar con ellos, pero sin perder su nitidez y calidad.

El sistema desarrollado permite una manipulación fácil de las diferentes estructuras tridimensionales y permite realizar la simulación de una técnica útil y común en los procedimientos neuroquirúrgicos, la craneotomía, la cual permite realizar el corte de pericráneo utilizando formas irregulares, lo que lo acerca más al procedimiento quirúrgico real.

El periférico de entrada diseñado cumple con la función por la cual fue elegido de hacer que la tecnología sea lo más natural posible para el usuario, además se puede conectar a computadoras con sistema operativo Windows o Linux y la conexión con 3D Slicer a través de comunicación serial se hace de forma rápida y fluida, de esta forma no se ralentizan los procesos.

La aplicación de uso libre desarrollada se caracteriza por su fácil manejo y por no sobrecargar al usuario con opciones complicadas para su uso, presentando opciones de fácil comprensión y posibilidades de visualización para un mejor entendimiento de la información.

En un trabajo futuro se buscará mejorar los algoritmos implementados para permitir la planeación de neurocirugías para pacientes específicos sin necesidad de tener que realizar una capacitación extensa en el manejo del software, para llevar a cabo esto se necesitan desarrollar algoritmos que permitan realizar de forma automática o semiautomática la segmentación de las estructuras de interés. También se sugiere conectar el software 3D Slicer con un software que permita procesar imágenes de resonancia magnética funcional con el fin de reconstruir tridimensionalmente las zonas de activación para a partir de esta información, tomar decisiones acerca del abordaje adecuado en una neurocirugía.

Se agregarán diferentes funciones para mejorar el algoritmo de craneotomía virtual, tales como la implementación de un algoritmo de detección de colisiones y metodologías que permitan al neurocirujano calcular la trayectoria de menor riesgo. También se desarrollará un algoritmo de craneotomía virtual a partir de reconstrucciones volumétricas generadas con algoritmos de visualización directa.

El diseño del periférico de entrada será mejorado, utilizando diferentes materiales y cambiando los potenciómetros comunes por unos potenciómetros de precisión con el fin de que el movimiento de las herramientas controladas por el periférico sea más preciso.

En el futuro Se tendrá en cuenta la retroalimentación provista por los residentes de neurocirugía para mejorar la usabilidad y funcionalidad del simulador, de esta forma la herramienta desarrollada podrá ser utilizada con fines educativos.

BIBLIOGRAFIA

- [1] M. S. Donaldson, J. M. Corrigan, L. T. Kohn, and others, *To err is human: building a safer health system*, vol. 6. National Academies Press, 2000.
- [2] K. Aquilina, P. Edwards, and A. Strong, "Principles and practice of image-guided neurosurgery," in *Neurosurgery*, Springer, 2005, pp. 123–138.
- [3] F. A. Jolesz, *Intraoperative Imaging and Image-Guided Therapy*. 2014.
- [4] R. J. Maciunas, "Computer-assisted neurosurgery," *Clin. Neurosurg.*, vol. 53, p. 267, 2006.
- [5] L. Joskowicz and R. Taylor, "Computers in Imaging and Guided Surgery," *Comput. Sci. Eng.*, pp. 51–64, 2001.
- [6] T. M. Peters, "Image-guidance for surgical procedures," *Phys. Med. Biol.*, vol. 51, no. 14, p. R505, 2006.
- [7] S. Chan, F. Conti, K. Salisbury, and N. H. Blevins, "Virtual reality simulation in neurosurgery: Technologies and evolution," *Neurosurgery*, vol. 72, no. January, pp. 154–164, 2013.
- [8] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. V Miller, S. Pieper, and R. Kikinis, "3D Slicer as an image computing platform for the Quantitative Imaging Network.," *Magn. Reson. Imaging*, vol. 30, no. 9, pp. 1323–41, Nov. 2012.
- [9] O. V. Larsen, J. Haase, L. R. Østergaard, and others, "The virtual brain project," *Stud Heal. Technol Inf.*, vol. 81, pp. 256–262, 2001.
- [10] A. Alaraj, M. G. Lemole, J. H. Finkle, R. Yudkowsky, A. Wallace, C. Luciano, P. P. Banerjee, S. H. Rizzi, and F. T. Charbel, "Virtual reality training in neurosurgery: Review of current status and future applications.," *Surg. Neurol. Int.*, vol. 2, p. 52, Jan. 2011.
- [11] A. Radetzky, M. Rudolph, S. Starkie, B. Davies, and L. M. Auer, "ROBO-SIM: a simulator for minimally invasive neurosurgery using an active manipulator.," *Stud. Health Technol. Inform.*, vol. 77, pp. 1165–1169, 1999.
- [12] J. Georgii, M. Eder, L. Kovacs, A. Schneider, M. Dobritz, and R. Westermann, "Advanced Volume Rendering for Surgical Training Environments," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 2, p. S285, 2007.
- [13] K. Yamada, K. Sakai, K. Akazawa, S. Yuen, and T. Nishimura, "MR tractography: a review of its clinical applications.," *Magn. Reson. Med. Sci.*, vol. 8, no. 4, pp. 165–74, Jan. 2009.
- [14] "Magnetic Resonance Imaging (MRI Scan)." [Online]. Available: http://www.medicinenet.com/mri_scan/article.htm. [Accessed: 26-Apr-2015].
- [15] M. Smits, M. W. Vernooij, and G. C. Houston, "Incorporating Functional MR Imaging into Diffusion Tensor Tractography in the Preoperative Assessment of the Corticospinal Tract in Patients with Brain Tumors," *AJNR Am J Neuroradiol*, 2007.

- [16] R. A. Kockro, L. Serra, Y. Tseng-Tsai, C. Chan, S. Yih-Yian, C. Gim-Guan, E. Lee, L. Y. Hoe, N. Hern, and W. L. Nowinski, "Planning and simulation of neurosurgery in a virtual reality environment," *Neurosurgery*, vol. 46, no. 1, pp. 118–137, 2000.
- [17] N. Brown, S. Natsupakpong, S. Johannsen, S. Manjila, Q. Cai, V. Liberatore, A. R. Cohen, and M. C. Cavusoglu, "Virtual environment-based training simulator for endoscopic third ventriculostomy," *Stud. Health Technol. Inform.*, vol. 119, pp. 73–75, 2006.
- [18] N. I. Phillips and N. W. John, "Web-based surgical simulation for ventricular catheterization," *Neurosurgery*, vol. 46, no. 4, pp. 933–937, 2000.
- [19] A. Alaraj, C. J. Luciano, D. P. Bailey, A. Elsenousi, B. Z. Roitberg, A. Bernardo, P. P. Banerjee, and F. T. Charbel, "Virtual reality cerebral aneurysm clipping simulation with real-time haptic feedback," *Neurosurgery*, vol. 11, pp. 52–58, 2015.
- [20] S. Delorme, D. Laroche, R. Diraddo, and R. F. Del Maestro, "NeuroTouch: A physics-based virtual simulator for cranial microneurosurgery training," *Neurosurgery*, vol. 71, no. SUPPL.1, pp. 32–42, 2012.
- [21] A. Joshi, D. Scheinost, K. P. Vives, D. D. Spencer, L. H. Staib, and X. Papademetris, "Novel interaction techniques for neurosurgical planning and stereotactic navigation," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1587–1594, 2008.
- [22] G. Wong, C. Zhu, A. Ahuja, and W. Poon, "Craniotomy and CLipping of Intracranial Aneurysm in a Stereoscopic Virtual Reality Enviroment," *Neurosurgery*, vol. 61, no. 3, pp. 564–569, 2007.
- [23] G. Echegaray, I. Herrera, I. Aguinaga, C. Buchart, and D. Borro, "A brain surgery simulator," *IEEE Comput. Graph. Appl.*, vol. 34, no. 3, pp. 12–18, 2014.
- [24] C. A. Diaz, D. Posada, H. Trefftz, and J. Bernal, "Development of a Surgical Simulator to training Laparoscopic procedures," *Int. J. Educ. Inf. Technol.*, vol. 2, no. 1, 2008.
- [25] B. A. Pérez Gutierrez, "Módulo de navegación para un sistema de entrenamiento virtual aplicado a cirugía de la base del cráneo," *Trab. grado para optar al título Magíster en Ing. - Ing. Sist. Univ. Nac. Colomb.*, 2008.
- [26] O. A. Vivas, C. F. Rengifo, and V. H. Mosquera, "Simulador quirúrgico para operaciones de laparoscopia utilizando interfaces hápticas," *CIE2011*, pp. 1–4.
- [27] J. F. Ochoa and J. L. Ascencio, "Implementación de un método para la visualización de electrodos subdurales en cirugía de epilepsia," *Rev. Colomb. Radiol.*, pp. 3628–3631, 2013.
- [28] A. F. Botero, D. E. Higueta, M. Ramírez, J. J. Velásquez, M. A. Rincón, S. I. Duque, and J. F. Ochoa, "Técnicas de Visualización para el Entrenamiento Neuroquirúrgico en Software Libre.," *PAN Am. Heal. CARE Exch. (PAHCE)*.

Conf. Work. Exhib. Coop. / LINKAGES, 2013.

- [29] B. Preim and D. Bartz, *Visualization in medicine*, vol. 21. 2006.
- [30] “Not your typical graph or chart, but a 3D reconstruction/visualization of my brain’s diffusion data, detailing the neurons connecting the motor regions of my brain to my spinal cord (Corticospinal Tract) [OC]: dataisbeautiful.” [Online]. Available: https://www.reddit.com/r/dataisbeautiful/comments/35ba77/not_your_typical_graph_or_chart_but_a_3d/. [Accessed: 10-Feb-2017].
- [31] X. Lin and T. Qiu, “Denoise MRI images using sparse 3D transformation domain collaborative filtering,” *Proc. - 2011 4th Int. Conf. Biomed. Eng. Informatics, BMEI 2011*, vol. 1, pp. 233–236, 2011.
- [32] G. Gerig, O. Kubler, R. Kikinis, and F. a Jolesz, “Nonlinear anisotropic filtering of MRI data,” *IEEE Trans. Med. Imaging*, vol. 11, no. 2, pp. 221–232, 1992.
- [33] a. Buades and B. Coll, “A non-local algorithm for image denoising,” *Comput. Vis. Pattern*, vol. 2, no. 0, pp. 60–65, 2005.
- [34] J. A. Escario, J. V Martínez-Quiñones, J. Aso, R. Arregui, and P. G. Albero, “Análisis y proceso de imagen. Fundamentos y aplicaciones en neurología y neurocirugía,” *Rev. Neurol.*, vol. 53, no. 8, pp. 494–503, 2011.
- [35] J. R. Martinez De Dios, “Técnicas de segmentación,” *Curso*. .
- [36] S. Javeed Hussain, a. Satya Savithri, and P. V. Sree Devi, “Segmentation of Brain MRI With Statistical and 2D Wavelet Features by Using Neural Networks,” *Trendz Inf. Sci. Comput. (TISC), 2011 3rd Int. Conf. IEEE.*, pp. 154–159, 2011.
- [37] J. F. Ochoa, I. Aristizabal, J. D. Lemos, and H. Barrientos, “Evaluation of Free Distribution Tools for Image Guide Surgery,” *Rev. Politécnica*, vol. 10, no. 6, pp. 90–100, 2010.
- [38] P. Risholm, A. J. Golby, and W. Wells, “Multimodal image registration for preoperative planning and image-guided neurosurgical procedures,” *Neurosurg. Clin. N. Am.*, vol. 22, no. 2, pp. 197–206, 2011.
- [39] J. Pascau, J. D. Gispert, S. Reig, and M. Desco, “Registro de imágenes en medicina nuclear,” *Rev.R.Acad.Cienc.Exact.Fis.Nat. (Esp)*, vol. 96, pp. 29–43, 2002.
- [40] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3D surface construction algorithm,” *ACM SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, 1987.
- [41] C. Y. Tang, W. Chin, Y. P. Chui, W. S. Poon, and P. A. Heng, “A virtual reality-based surgical simulation system for virtual neuroendoscopy,” *IEEE ICIT 2007 - 2007 IEEE Int. Conf. Integr. Technol.*, vol. I, pp. 253–258, 2007.
- [42] a. Joshi, D. Scheinost, R. Globinsky, K. P. Vives, D. D. Spencer, L. H. Staib, and X. Papademetris, “Augmented inline-based navigation for stereotactic image guided neurosurgery,” *Proc. - Int. Symp. Biomed. Imaging*, vol. 1, pp. 1869–1872, 2011.

- [43] “Mind Reading Markup Language (MRML).” [Online]. Available: <http://ifaq.wap.org/computers/mrml.html>.
- [44] “Qt | Cross-platform software development for embedded & desktop.” [Online]. Available: <https://www.qt.io/>.
- [45] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit, Third Edition*. Kitware Inc.
- [46] M. W. Woolrich, S. Jbabdi, B. Patenaude, M. Chappell, S. Makni, T. Behrens, C. Beckmann, M. Jenkinson, and S. M. Smith, “Bayesian analysis of neuroimaging data in FSL,” *Neuroimage*, vol. 45, no. 1, pp. S173--S186, 2009.
- [47] M. Jenkinson, M. Pechaud, and S. Smith, “BET2: MR-based estimation of brain, skull and scalp surfaces,” in *Eleventh annual meeting of the organization for human brain mapping*, 2005, vol. 17, p. 167.
- [48] L. Ibanez, W. Schroeder, L. Ng, and J. Cates, “The ITK software guide,” 2005.
- [49] “Über Voxel-Man.” [Online]. Available: <http://www.voxel-man.de/ueber/>. [Accessed: 01-May-2015].
- [50] B. Preim and C. P. Botha, *Visual Computing for Medicine: Theory, Algorithms, and Applications*. Newnes, 2013.
- [51] Y. Fu, Y. Xiao, and S. Wang, “A novel interactive volume-exploring method for medical data based on texture-mapping techniques,” *2007 IEEE/ICME Int. Conf. Complex Med. Eng. C. 2007*, pp. 562–565, 2007.
- [52] Q. Zhang, R. Eagleson, and T. M. Peters, “GPU-based image manipulation and enhancement techniques for dynamic volumetric medical image visualization,” in *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, 2007, pp. 1168–1171.
- [53] B. Preim, W. Spindler, K. Oldhafer, and H. Peitgen, “3D-Interaction Techniques for Planning of Oncologic Soft Tissue Operations.,” *Graph. Interface*, 2001.
- [54] O. Konrad-Verse, B. Preim, and A. Littmann, “Virtual Resection with a Deformable Cutting Plane.,” *SimVis*, no. 1, pp. 203–214, 2004.
- [55] J. R. van der Vorst, R. M. van Dam, R. S. A. van Stiphout, M. A. van den Broek, I. H. Hollander, A. G. H. Kessels, and C. H. C. Dejong, “Virtual liver resection and volumetric analysis of the future liver remnant using open source image processing software,” *World J. Surg.*, vol. 34, no. 10, pp. 2426–2433, 2010.
- [56] J. Klein, O. Friman, M. Hadwiger, B. Preim, F. Ritter, A. Vilanova, G. Zachmann, and D. Bartz, “Visual computing for medical diagnosis and treatment,” *Comput. Graph.*, vol. 33, no. 4, pp. 554–565, 2009.
- [57] S. Bandi and D. Thalmann, “An adaptive spatial subdivision of the object space for fast collision detection of animated rigid bodies,” in *Computer Graphics Forum*, 1995, vol. 14, no. 3, pp. 259–270.
- [58] S. Melax, “Dynamic Plane Shifting BSP Traversal,” *Graph. Interface*, vol. c, pp. 213–220, 2000.

- [59] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in *ACM SIGGRAPH 2005 Courses*, 2005, p. 42.
- [60] C. Sigg, R. Peikert, and M. Gross, "Signed distance transform using graphics hardware," in *Visualization, 2003. VIS 2003. IEEE*, 2003, pp. 83–90.
- [61] "Craniotomy," *Mayfield Clinic, Neurosurgery*. [Online]. Available: <http://www.mayfieldclinic.com/PE-Craniotomy.htm>.
- [62] "Kinect for Windows." [Online]. Available: <http://www.microsoft.com/en-us/kinectforwindows/>. [Accessed: 08-May-2015].
- [63] "Leap Motion | Mac & PC Motion Controller for Games, Design, & More." [Online]. Available: <https://www.leapmotion.com/#faq>. [Accessed: 08-May-2015].
- [64] "3Dconnexion: SpaceNavigator." [Online]. Available: <http://www.3dconnexion.com/index.php?id=26&redirect2=www.3dconnexion.com>. [Accessed: 08-May-2015].
- [65] "Medical Aurora - Medical." [Online]. Available: <http://www.ndigital.com/medical/products/aurora/>. [Accessed: 08-May-2015].
- [66] D. esteban Higueta and A. F. Botero, "Diseño de un dispositivo de posicionamiento con conexión al software 3D-Slicer," 2014.

ANEXOS

ANEXO 1: CÓDIGO FUENTE CRANEOTOMÍA VIRTUAL

A continuación se presenta el código desarrollado en Python para realizar la craneotomía virtual:

```
import sys
import numpy

class clipping:
    def __init__(self,parent):
        parent.title="Craneotomia"
        parent.contributors=["Sara Duque, John Fredy Ochoa"]
        parent.helpText="Entorno de craneotomía virtual "
        parent.acknowledgementText = "" <b>proyecto clipping </b>""
        self.parent = parent

class clippingWidget:
    def __init__(self, parent=None):
        if not parent:
            self.parent = slicer.qMRMLWidget()
            self.parent.setMRMLScene(slicer.mrmlScene)
            self.layout = self.parent.layout()
            self.parent.show()
            self.setup()
        else:
            self.parent = parent
            self.layout = parent.layout()
            self.labelNode = None
            self.observerTags=[0]*5

    def setup(self):

        self.clipping_collpasible_button = ctk.ctkCollapsibleButton()
        self.clipping_collpasible_button.text = "Craneotomia"
        self.layout.addWidget(self.clipping_collpasible_button)
        self.clipping_collpasible_button.collapsed = False

        self.clipping_form = qt.QFormLayout(self.clipping_collpasible_button)
        self.clipping_frame = qt.QFrame(self.clipping_collpasible_button)
        self.clipping_frame.setLayout(qt.QHBoxLayout())
        self.clipping_form.addWidget(self.clipping_frame)

        self.clipping_model_frame = qt.QFrame(self.clipping_collpasible_button)
        self.clipping_model_frame.setLayout(qt.QHBoxLayout())
        self.clipping_form.addWidget(self.clipping_model_frame)

        self.dinamic_clipping = qt.QPushButton("Clipping")
        self.clipping_form.addWidget(self.dinamic_clipping)
        self.dinamic_clipping.connect('clicked()',self.asociarReferenciaEsfera)
```

```

def crearEsfera(self):

    self.radio=5
    nombre_esfera='esfera'
    #Crear la esfera a utilizar
    escena = slicer.mrmlScene
    self.sphere = vtk.vtkSphere()
    self.sphere.SetRadius(self.radio)

    self.spheres = vtk.vtkSphereSource()
    self.spheres.SetRadius(self.radio)

def CargarModelo(self):
    band_esc1="12"
    esc="/home/sara/Documentos/ImagenesTesisNii/Paciente"+band_esc1+"/Escena/2015-04-07-
Scene.mrml"
    slicer.util.loadScene(esc)
    escena=slicer.mrmlScene
    nodo=slicer.util.getNode('Piel')
    escena.RemoveNode(nodo)
    self.reader=vtk.vtkPolyDataReader()

filename="/home/sara/Documentos/ImagenesTesisNii/Paciente"+band_esc1+"/Escena/Piel.vtk"
    self.reader.SetFileName(filename)
    self.reader.Update()

def cliperPoly(self,input_modelo):

    self.cliper = vtk.vtkClipPolyData()
    self.cliper.SetInputConnection(input_modelo.GetOutputPort())
    self.cliper.SetClipFunction(self.sphere)
    self.cliper.GenerateClipScalarsOn()
    self.cliper.SetValue(0.0)
    self.new_modelo=self.cliper.GetOutputPort();

#para mostrar en el slicer
def muestraObjeto(self,obj):

    escena = slicer.mrmlScene
    nombre_modelo='modelo'
    self.newModel = slicer.vtkMRMLModelNode()
    self.newModel.SetName(nombre_modelo)
    self.newModel.SetScene(escena)
    self.newModel.SetAndObservePolyData(obj.GetOutput())

    slicer.mrmlScene.AddNode(self.newModel)
    self.newModelDisplay = slicer.vtkMRMLModelDisplayNode()
    self.newModelDisplay.SetInputPolyData(self.newModel.GetPolyData())
    self.newModelDisplay.SetColor(1,0.9,0.8)
    self.newModelDisplay.SetBackfaceCulling(0)
    self.newModelDisplay.SetSliceIntersectionVisibility(1)
    self.newModelDisplay.SetVisibility(1)
    self.newModelDisplay.SetOpacity(1)
    slicer.mrmlScene.AddNode(self.newModelDisplay)

```

```

self.newModel.SetAndObserveDisplayNodeID(self.newModelDisplay.GetID())

def crearReferencia(self):
    #Referencia para desplazar la esfera
    fnode = slicer.vtkMRMLMarkupsFiducialNode()
    # Cambiar el nombre
    fnode.SetName('F')
    # Agregar una referencia (Fiducial)
    posicion = [100,100,100]
    fnode.AddFiducialFromArray(posicion)
    # Mostrar el nodo en la escena del Slicer
    slicer.mrmlScene.AddNode(fnode)

def onReferenciaMov(self,caller, event):
    #Funcion que movera la esfera

    posicion = numpy.array(numpy.zeros(3))
    referencia = slicer.util.getNode('F')
    referencia.GetNthFiducialPosition(0,posicion) # asigna el valor de la posicion 0, al vector
posicion

##CLIPING CONTINUO CON EVENTOS
self.crearEsfera()
self.spheres.SetCenter(posicion[0],posicion[1],posicion[2])
self.sphere.SetCenter(posicion[0],posicion[1],posicion[2])
self.cliperPoly(self.cliper)
self.newModel.SetAndObservePolyData(self.cliper.GetOutput())
self.newModelDisplay.SetInputPolyData(self.newModel.GetPolyData())
self.newModel.SetAndObserveDisplayNodeID(self.newModelDisplay.GetID())

def asociarReferenciaEsfera(self):
    #Enlaza todas las funciones creadas
    self.crearEsfera()
    self.crearReferencia()
    self.CargarModelo()
    self.cliperPoly(self.reader)
    self.muestraObjeto(self.cliper)
    referencia = slicer.util.getNode('F')
    referencia.AddObserver(referencia.PointModifiedEvent,self.onReferenciaMov)

```

ANEXO 2: CÓDIGO FUENTE PARA LA RECEPCIÓN DE DATOS VIA SERIAL

En este anexo se encuentra el código fuente para la recepción de datos via serial en 3D Slicer y su transformación para el movimiento en la escena de la cámara y de la herramienta de craneotomía virtual.

```
def recibirDato(self):

    try:
        datoR = self.s.read(70)
        print "dato: ",datoR,self.puertoCOM
        for i in range(50):
            if(datoR[i] == '0'):
                if(datoR[i + 5] == '1'):
                    if(datoR[i + 10] == '2'):
                        if(datoR[i + 15] == '3'):
                            if(datoR[i + 20] == '4'):
                                if(datoR[i + 25] == '5'):
                                    if(datoR[i + 30] == '6'):
                                        self.bandTransformacion = 1
                                        self.dato = datoR[(i + 50)]
                                        self.dato0 = int(self.dato[1:5])
                                        self.dato1 = int(self.dato[6:10])
                                        self.dato2 = int(self.dato[11:15])
                                        self.dato3 = int(self.dato[16:20])
                                        self.dato4 = int(self.dato[21:25])
                                        self.dato5 = int(self.dato[26:30])
                                        self.flagCraneotomy = int(self.dato[31])
                                        break
                                    else:
                                        self.bandTransformacion = 0

            self.acumulador1 += self.dato0
            self.acumulador2 += self.dato1
            self.acumulador3 += self.dato2
            self.acumulador4 += self.dato3
            self.acumulador5 += self.dato4
            self.acumulador6 += self.dato5
            self.cont = self.cont + 1
            ndatosfilt=10
            if self.cont> ndatosfilt:
                self.cont=0

        if(self.bandTransformacion == 1 and self.cont == ndatosfilt): # and self.cont == 1):
            self.cont = 0
            self.acumulador1 = (self.acumulador1 / ndatosfilt)
            self.acumulador2 = (self.acumulador2 / ndatosfilt)
            self.acumulador3 = (self.acumulador3 / ndatosfilt)
            self.acumulador4 = (self.acumulador4 / ndatosfilt)
            self.acumulador5 = (self.acumulador5 / ndatosfilt)
            self.acumulador6 = (self.acumulador6 / ndatosfilt)

            self.contadorCrearModel = self.contadorCrearModel + 1

            grados1 = ((270.0 / 4095.0) * (int(self.acumulador1)))
```

```

grados2 = ((270.0 / 4095.0) * (int(self.acumulador2)) + 10.0)
grados3 = (((270.0 / 4095.0) * (int(self.acumulador3)))) - 90.0

self.acumulador1 = 0
self.acumulador2 = 0
self.acumulador3 = 0

grados1 = int(grados1)
grados2 = int(grados2)
grados3 = int(grados3)
#####Control de la Cámara
rmax=700
v1=(self.acumulador4*np.pi/4095)
v2=(self.acumulador5*np.pi/2048)
v3=(self.acumulador6*rmax/4095)
x0=v3*np.sin(v1)*np.cos(v2)
y0=v3*np.sin(v1)*np.sin(v2)
z0=v3*np.cos(v1)
self.camera.SetPosition([x0,y0,z0])
#####Fin camara
if(int(self.dato0) > 4100 or int(self.dato1) > 4100 or int(self.dato2) > 4100):
    pass

else:
    mat1_1 = np.matrix([[1, 0, 0, -100.0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]])
    mat1_2 = np.matrix([[1, 0, 0, 0], [0, 1, 0, -100.0], [0, 0, 1, 0], [0, 0, 0, 1]])
    mat1_3 = np.matrix([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, -100.0], [0, 0, 0, 1]])
    mat1_4 = mat1_1 * mat1_2
    mat1 = mat1_3 * mat1_4

    #rotacion alrededor del eje z
    mat2 = np.matrix([[np.cos((np.pi) * grados1 / 180), -np.sin((np.pi) * grados1 / 180), 0, 0],
        [np.sin((np.pi) * grados1 / 180), np.cos((np.pi) * grados1 / 180), 0, 0],
        [0, 0, 1, 0],
        [0, 0, 0, 1]])
    #rotacion alrededor del eje y

    mat3 = np.matrix([[np.cos((np.pi) * grados2 / 180), 0, np.sin((np.pi) * grados2 / 180), 0],
        [0, 1, 0, 0],
        [-np.sin((np.pi) * grados2 / 180), 0, np.cos((np.pi) * grados2 / 180), 0],
        [0, 0, 0, 1]])
    #traslacion alrededor del eje z

    mat4 = np.matrix([[1, 0, 0, 0],
        [0, 1, 0, 0],
        [0, 0, 1, 200.0],
        [0, 0, 0, 1]])
    #rotacion alrededor del eje y

    mat5 = np.matrix([[np.cos((np.pi) * grados3 / 180), 0, np.sin((np.pi) * grados3 / 180), 0],
        [0, 1, 0, 0],
        [-np.sin((np.pi) * grados3 / 180), 0, np.cos((np.pi) * grados3 / 180), 0],
        [0, 0, 0, 1]])

    mat6 = mat1 * mat2
    mat7 = mat6 * mat3

```

```

mat8 = mat7 * mat4
mat9 = mat8 * mat5
mat10 = np.array(mat9)
self.TransformationMatrix0 = mat10
# translacion

self.x = mat10[0][3]
self.y = mat10[1][3]
self.z = mat10[2][3]

if self.cont_posicion1 == 0:
    self.posicion_anterior = [int(self.x), int(self.y), int(self.z)]
    self.cont_posicion1 = self.cont_posicion1 + 1
elif self.cont_posicion1 == 1:
    self.cont_posicion1 = 0
    self.posicion_posterior = [int(self.x), int(self.y), int(self.z)]
    self.deltaX = math.fabs(self.posicion_posterior[0] - self.posicion_anterior[0])
    self.deltaY = math.fabs(self.posicion_posterior[1] - self.posicion_anterior[1])
    self.deltaZ = math.fabs(self.posicion_posterior[2] - self.posicion_anterior[2])

limdelta=20

if ((self.deltaX > limdelta) or (self.deltaY > limdelta) or (self.deltaZ > limdelta)):
    self.flagDeltaFuera = 1
else:
    self.flagDeltaFuera = 0

lim=1000

if ((self.x > lim or self.x < - lim) or (self.y > lim or self.y < - lim) or (self.z > lim or self.z < -
lim)): #or (self.flagDeltaFuera == 1):
    pass
else:
    self.transform_cil.GetMatrixTransformToParent().SetElement(0, 3, mat10[0][3])
    self.transform_cil.GetMatrixTransformToParent().SetElement(1, 3, mat10[1][3])
    self.transform_cil.GetMatrixTransformToParent().SetElement(2, 3, mat10[2][3])

## rotacion
self.transform_cil.GetMatrixTransformToParent().SetElement(0, 0, mat10[0][0])
self.transform_cil.GetMatrixTransformToParent().SetElement(0, 1, mat10[0][1])
self.transform_cil.GetMatrixTransformToParent().SetElement(0, 2, mat10[0][2])

self.transform_cil.GetMatrixTransformToParent().SetElement(1, 0, mat10[1][0])
self.transform_cil.GetMatrixTransformToParent().SetElement(1, 1, mat10[1][1])
self.transform_cil.GetMatrixTransformToParent().SetElement(1, 2, mat10[1][2])

self.transform_cil.GetMatrixTransformToParent().SetElement(2, 0, mat10[2][0])
self.transform_cil.GetMatrixTransformToParent().SetElement(2, 1, mat10[2][1])
self.transform_cil.GetMatrixTransformToParent().SetElement(2, 2, mat10[2][2])

self.transform_cil.GetMatrixTransformToParent().SetElement(3, 0, 0)
self.transform_cil.GetMatrixTransformToParent().SetElement(3, 1, 0)
self.transform_cil.GetMatrixTransformToParent().SetElement(3, 2, 0)
self.transform_cil.GetMatrixTransformToParent().SetElement(3, 3, 1)

```

```
if((self.deltaX == 0) and (self.deltaY == 0) and (self.deltaZ == 0)):
    pass
else:
    lista = slicer.util.getNode('Referencia')
    lista.SetNthFiducialPosition(0,int(self.x), int(self.y), int(self.z))
```


ANEXO 3: MANUAL DE USUARIO

1. INTRODUCCIÓN

Este manual está dirigido a entregar las pautas de operación de la aplicación “Simulador”. Esta aplicación se desarrolló en 3D Slicer, la cual consiste en un entorno para el acceso a operaciones avanzadas de procesamiento de imágenes y la manipulación intuitiva de imágenes volumétricas, el cual puede manipularse por medio de un periférico de entrada, mejorando y facilitando la forma de interacción usuario-sistema. El entorno incluye diferentes escenarios correspondientes a información tridimensional de pacientes que presentan una lesión tumoral y permite la realización de craneotomía virtual y la manipulación de representaciones tridimensionales.

2. REQUERIMIENTOS PARA EL USO DE LA APLICACIÓN

- 3D SLICER 4.3
- Librería pyserial
- Código fuente (Simulador)
- Escenarios precargados

3. GUIA DE USO

Al abrir 3D Slicer, este inicia con el módulo desarrollado, llamado *Simulador*, en la primera sesión se despliega una lista con todos los escenarios de diferentes pacientes precargados para realizar la simulación (Figura 1).

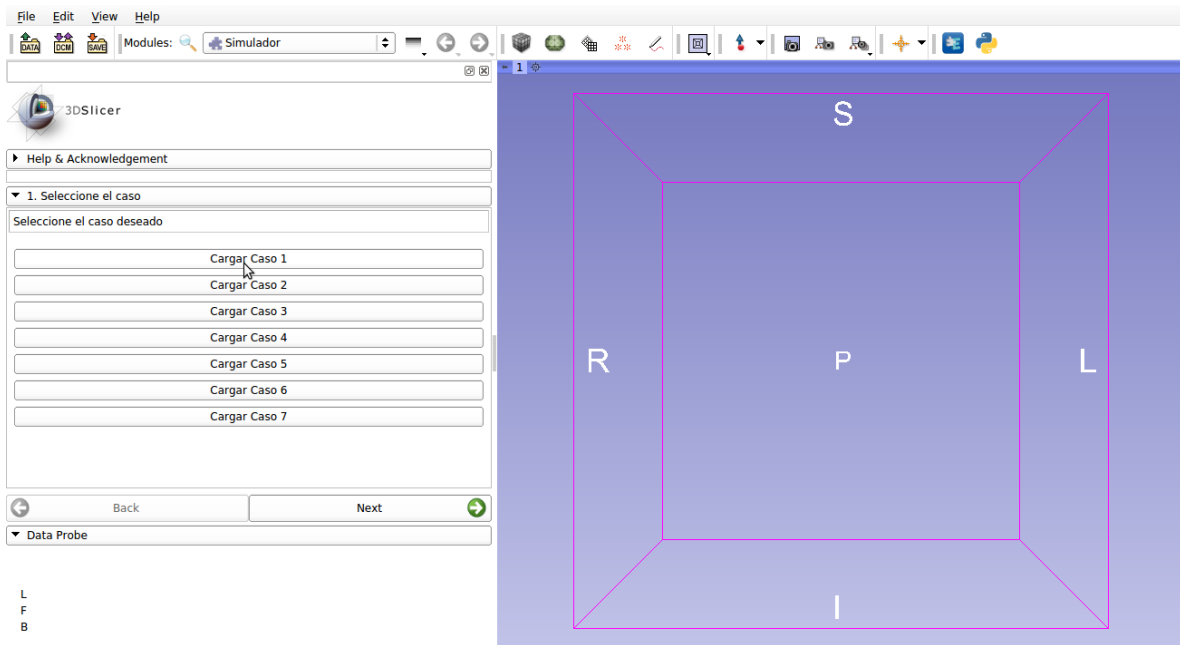


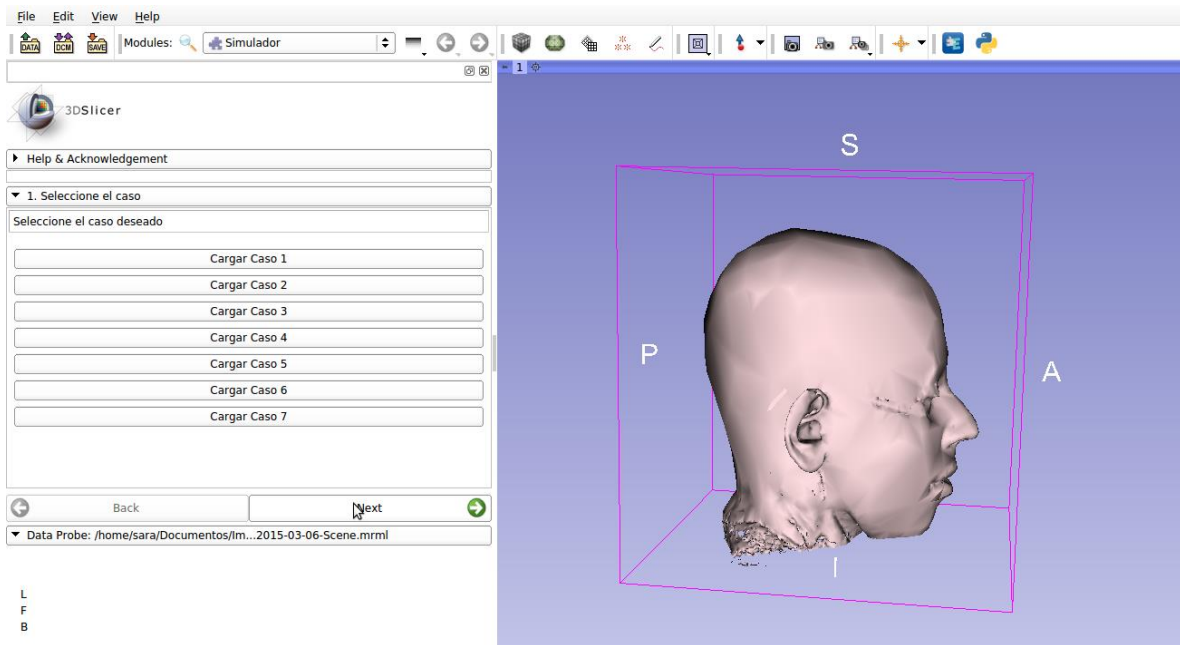
Figura 1. Simulador: Paso 1, Selección del caso.

La siguiente tabla incluye las estructuras disponibles en cada uno de los casos precargados.

Tabla 1. Reconstrucciones volumétricas disponibles para cada caso en el simulador

Caso / Estructura	Pericráneo	Cerebro	Lesión Tumoral	Tracto Córtilcoespinal	Árbol Venoso
1	X	X	X		X
2	X	X	X	X	
3	X	X	X	X	
4	X	X	X	X	
5	X	X	X	X	
6	X	X	X		
7	X	X	X		

Una vez se selecciona el caso deseado se carga en la escena de 3D Slicer y se debe hacer clic en *Next* para continuar con el paso 2 (Figura 2).



42Figura 2. Simulador: Paso 1, Caso cargado en la escena.

En el paso 2, se puede observar la información de las estructuras disponibles en el caso seleccionado, y se presentan algunas opciones de visualización (Figura 3).

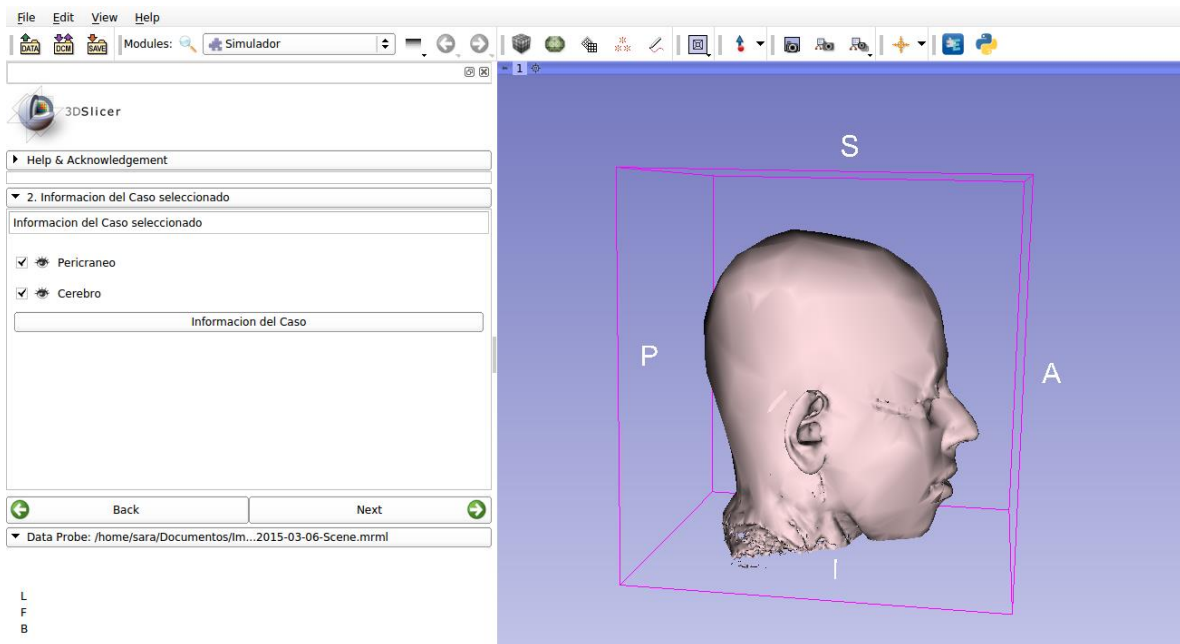


Figura 3. Simulador: Paso 2, Información del caso seleccionado.

Al elegir la opción *Pericráneo* se puede activar o desactivar la visualización de dicha estructura (Figura 4).

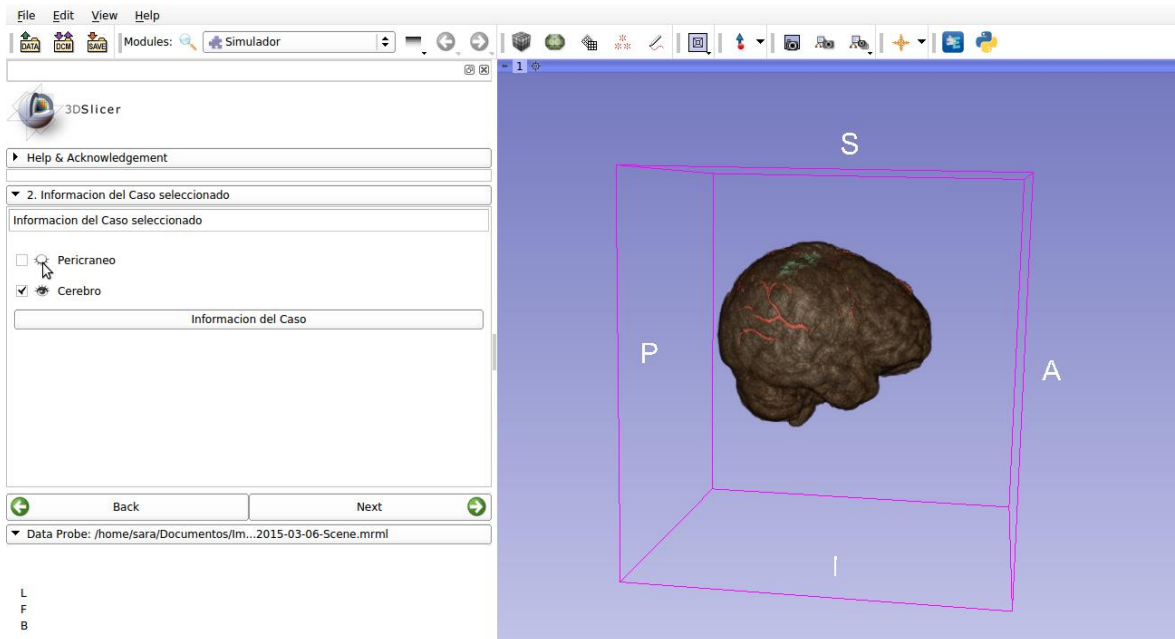


Figura 4. Simulador: Paso 2, Opción de visualización del pericráneo.

La opción *Cerebro* permite también activar o desactivar la reconstrucción volumétrica del cerebro para identificar las estructuras internas y poder tomar la decisión si ese es el caso con el que se desea trabajar (Figura 5).

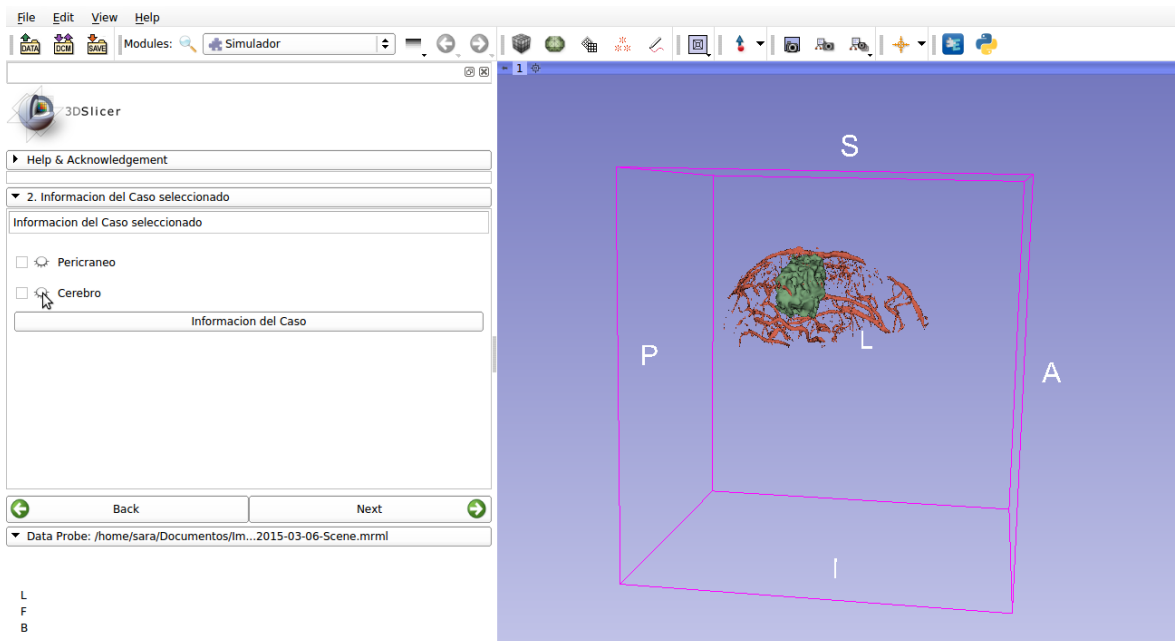


Figura 543. Simulador: Paso 2, opción de visualización del cerebro.

El botón *Información del Caso* despliega una lista de las estructuras disponibles en el caso seleccionado (Figura 6).

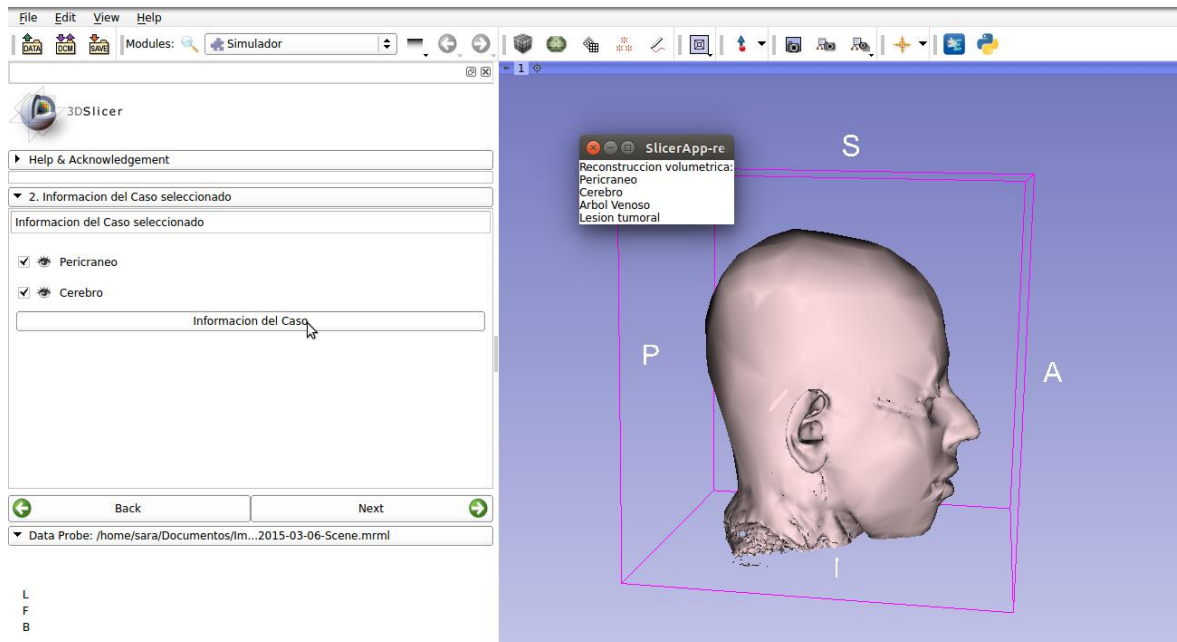


Figura 6. Simulador: Paso 2, Lista de estructuras disponibles.

El usuario puede hacer clic en el botón *Back* si el caso seleccionado no es el caso con el que se desea realizar la craneotomía, regresando al paso 1 para seleccionar otro caso, o puede hacer clic en el botón *Next* si desea iniciar la simulación de craneotomía con el caso seleccionado.

En el paso 3 se puede iniciar la simulación de la craneotomía, al hacer clic en el botón *Iniciar*. Para iniciar la simulación es necesario que el periférico de entrada esté conectado, de no ser así el programa lo recordará con un mensaje (Figura 7).

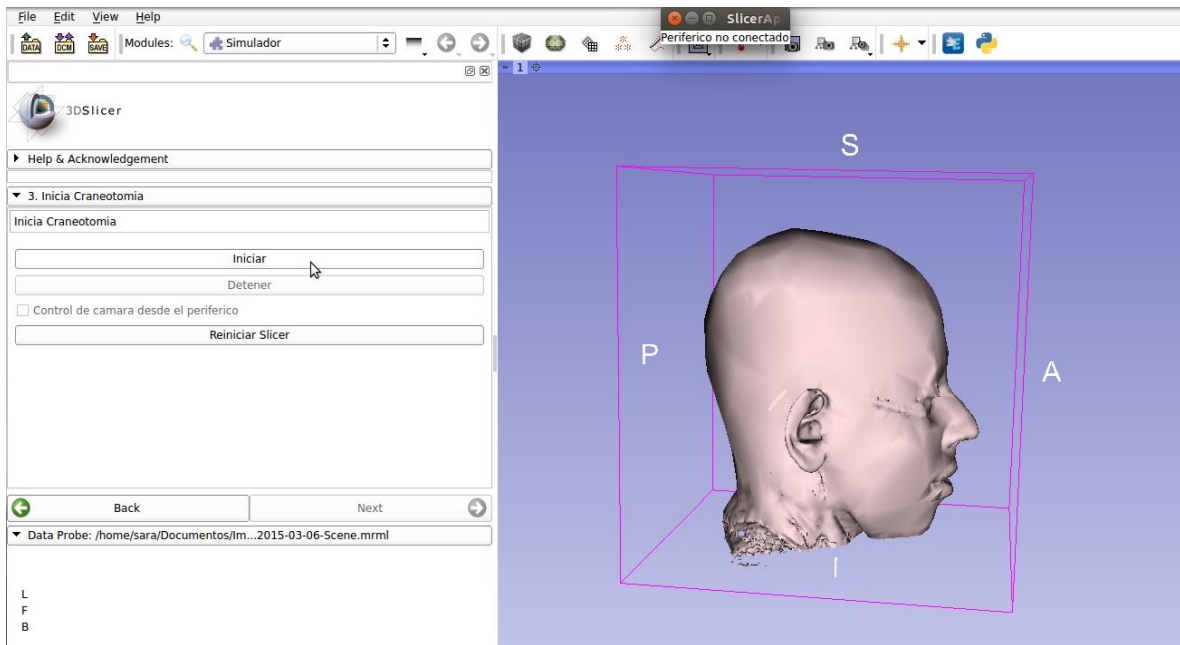


Figura 7. Simulador: Paso 3, Iniciar craneotomía, periférico no conectado.

Si el periférico se encuentra conectado el botón de *Iniciar* se deshabilita, se habilitan los botones *Detener* y *Control de cámara desde el periférico*, en escena aparece la herramienta de craneotomía representada por un cono e inicia la transmisión de datos desde el sistema de adquisición del periférico de entrada hacia el software 3D Slicer. Es en este momento que se puede realizar la craneotomía virtual, controlando el cono con el brazo mecatrónico del periférico de entrada y el interruptor disponible para elegir cuando realizar la resección (Figura 8).

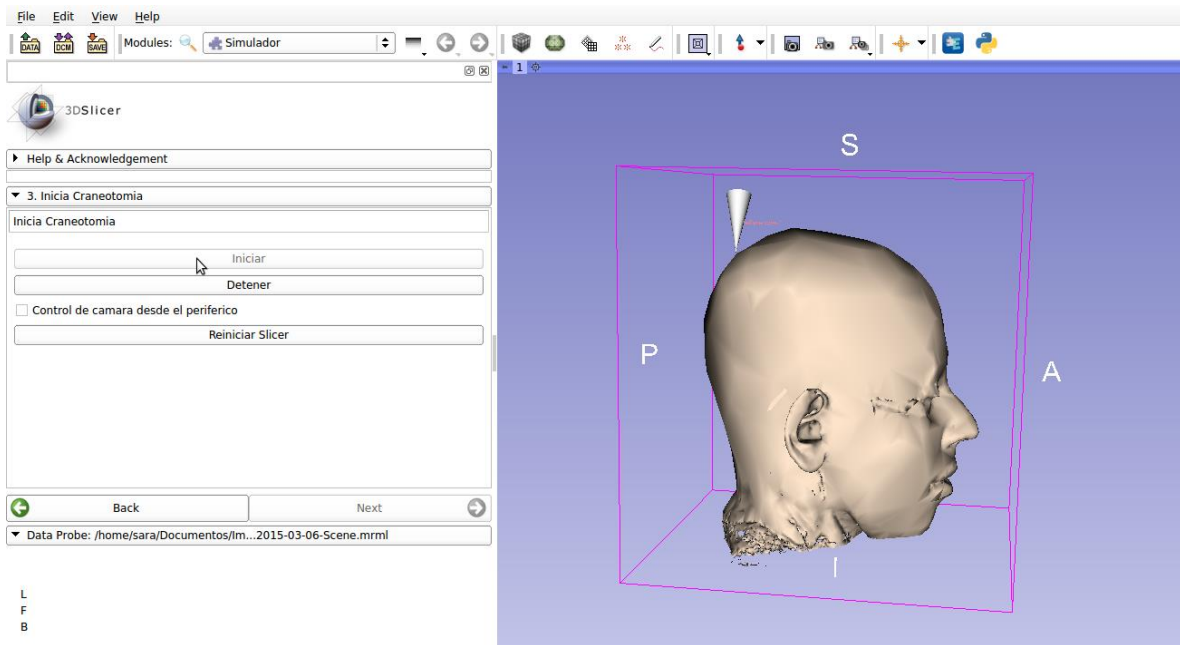


Figura 8. Simulador: Paso 3, Iniciar craneotomía, periférico conectado.

El movimiento del cono se realiza con el brazo mecánico rotando las articulaciones 1,2 y 3. Una vez se ubica el cono en la posición deseada se puede proceder a realizar la craneotomía. Para esto se desplaza hacia la izquierda el interruptor 7, si se desea interrumpir la perforación se desplaza el interruptor hacia la derecha (Figura 9).



Figura 9. Periférico de entrada y cada una de sus partes.

La cámara se puede controlar también desde el periférico de entrada solo si el usuario así lo solicita desde el software activando la opción *Control de cámara desde el periférico* (Figura 10), una vez se selecciona esta opción la cámara del entorno se puede manipular con los potenciómetros independientes disponibles en el periférico de entrada. Los potenciómetros 4 y 5 permiten rotar la imagen, y el potenciómetro 6 permite acercar o alejar la imagen (Figura 9)

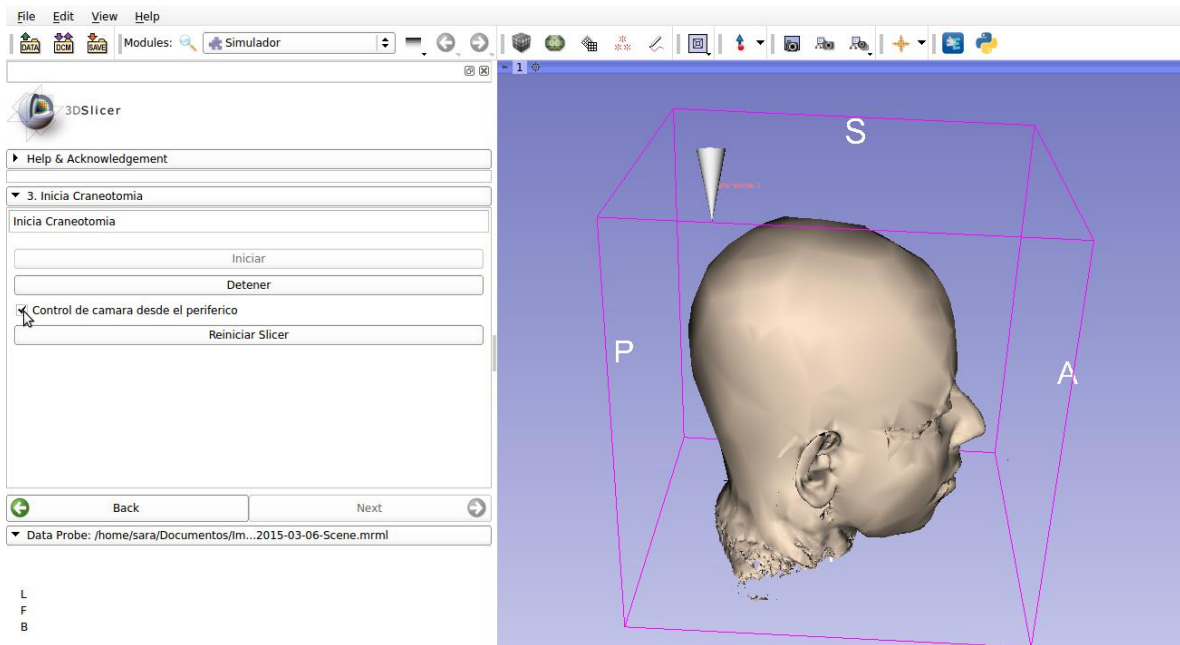


Figura 10. Simulador: Paso 3, Activar control de la cámara desde el periférico.

El botón *Detener* interrumpe el envío de información desde el periférico de entrada hacia el software 3D Slicer, y también interrumpe el algoritmo de craneotomía virtual. Al hacer clic se deshabilita este botón y la opción *Control de cámara desde el periférico* y se habilita la opción iniciar, así el usuario puede iniciar la craneotomía virtual en el momento que lo requiera (Figura 11Figura 40).

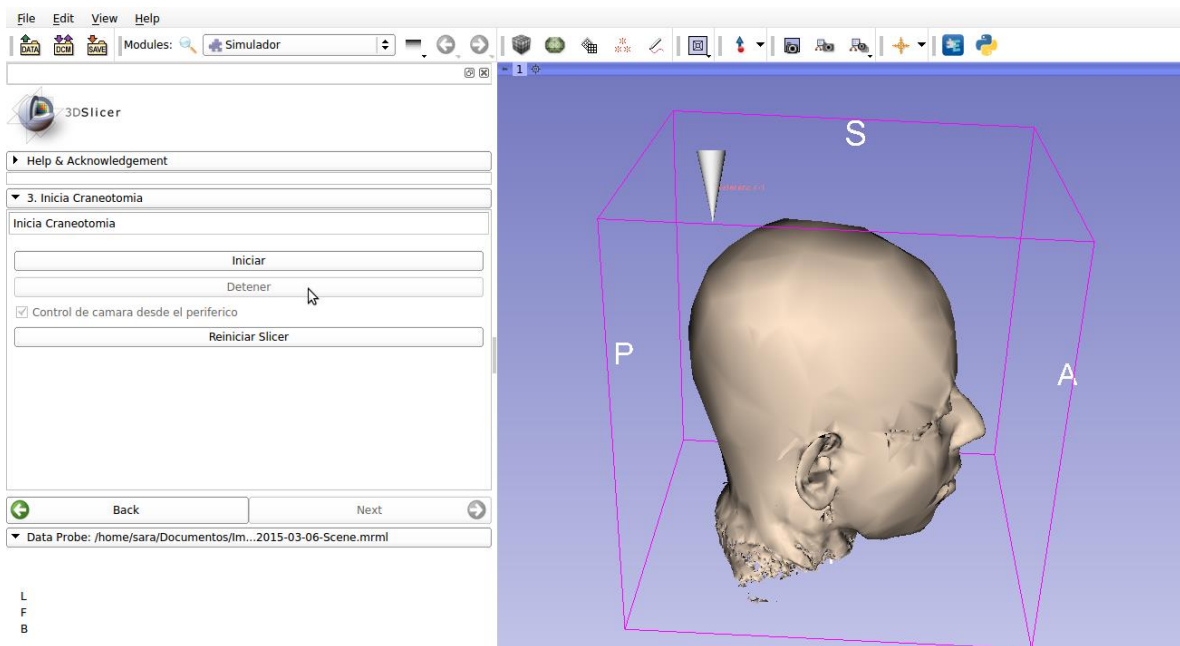


Figura 11. Simulador: Paso 3, Detener craneotomía.

ANEXO 4: ENCUESTA DE USABILIDAD

Para llevar a cabo la encuesta de usabilidad se les pedirá a los usuarios realizar una craneotomía virtual en alguno de los casos disponibles en el entorno virtual.

Según su experiencia con el entorno virtual, indique su grado de satisfacción con cada uno de los aspectos listados a continuación (Use una escala de 0 hasta 5, donde 5 es totalmente de acuerdo y 0 muy totalmente en desacuerdo)		
#	Pregunta	Evaluación
1	En general, estoy satisfecho con lo fácil que es usar el entorno virtual.	
2	Es simple usar este entorno virtual.	
3	Pude terminar efectivamente la actividad, usando el entorno virtual.	
4	Puede completar la actividad rápidamente usando el entorno virtual.	
5	Me sentí cómodo usando el entorno virtual.	
6	El entorno virtual muestra mensajes de error y pude solucionarlos rápidamente.	
7	Los textos e íconos son legibles e intuitivos.	
8	La información que se del manual de usuario fue clara.	
9	La interfaz del entorno virtual es agradable.	
10	La representación tridimensional de los datos es adecuada.	
11	La velocidad al interactuar con el entorno virtual es adecuada.	
12	En general estoy satisfecho con el entorno virtual.	