

**SISTEMA DE LOCALIZACIÓN DE TEJIDO EPIDÉRMICO PARA
AUTOMATIZACIÓN DE UNA SUTURA**



**Universidad
del Cauca®**

**Alan David Eraso Vargas
Jesus Eiber Velasco Otero**

**Director: Mg. Hermes Fabián Vargas Rosero
Co-Director: Mg. Elena Muñoz España**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Ingeniería Electrónica y Telecomunicaciones
Popayán
2022**

**SISTEMA DE LOCALIZACIÓN DE TEJIDO EPIDÉRMICO PARA
AUTOMATIZACIÓN DE UNA SUTURA**

**Alan David Eraso Vargas
Jesus Eiber Velasco Otero**

**Trabajo de grado presentando a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para optar al título de:
Ingeniero en Electrónica y Telecomunicaciones**

**Director:
Mg. Hermes Fabián Vargas Rosero**

**Co-Director:
Mg. Elena Muñoz España**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Ingeniería Electrónica y Telecomunicaciones
Popayán
2022**

Agradecimientos

En primera instancia por permitirnos estar presentes para lograr realizar nuestra formación académica y por darnos la sabiduría suficiente para enfrentar los retos que a diario se presentarón, en segunda instancia a nuestros padres por apoyarnos a creer en cada uno de nosotros y a nuestro director de tesis Ing. Fabián Vargas por su apoyo tanto intelectual como personal, para la realización de este proyecto.

Tabla de Contenido

1. Introducción	8
1.1. Planteamiento del problema	8
1.2. Trabajos relacionados	9
1.3. Objetivo	16
1.3.1. Objetivo General	16
1.3.2. Objetivos Específicos	16
2. Marco teórico	17
2.1. Visión por computadora en la robótica quirúrgica	17
2.1.1. Elementos externos de un sistema de visión por computadora enfocado a RAS semiautónomo en un procedimiento de sutura epidérmica superficial	19
2.1.2. Elementos internos de un sistema de visión por computadora enfocado a RAS semiautónomo en un procedimiento de sutura epidérmica superficial	21
2.2. Técnicas de reconstrucción 3D	22
2.3. Descripción de las características de la cámara Realsense D435	23
2.4. Modelos de visión por computadora	25
2.5. Software de localización	25
2.5.1. Método de Otsu	26
2.5.2. Algoritmo Watershed	26
2.5.3. Algoritmo Grabcut	26
2.6. Transformaciones homogéneas	28
2.6.1. Coordenadas homogéneas	28
2.6.2. Transformación de coordenadas	29
3. Metodología de trabajo	31
3.1. Establecimiento de los requisitos del sistema	32
3.1.1. Requisitos funcionales	32
3.1.2. Requisitos no funcionales	32
3.2. Diseño del sistema	33
3.2.1. Evaluación y selección de los componentes físicos del sistema de visión	33
3.2.1.1. Selección del dispositivo de captura	33
3.2.1.2. Posición espacial del dispositivo de captura	34
3.2.1.2.1. Ángulo de inclinación de la cámara con respecto al plano:	34
3.2.1.2.2. Distancia de la cámara con respecto al plano:	36
3.2.2. Estructura física del experimento	38
3.2.2.1. Cámara de profundidad D435	38

3.2.2.2.	Pinza Laparoscópica	38
3.2.2.3.	Luxómetro	38
3.2.2.4.	Almohadilla de sutura	39
3.2.2.5.	Soporte móvil	39
3.2.2.6.	Lámpara	40
3.2.3.	Análisis y selección del entorno y herramientas de programación para el sistema	41
3.2.3.1.	Librería Open CV-Python	42
3.2.3.2.	Robot Operating System (ROS)	42
3.2.3.2.1.	RosBridge	42
3.2.3.3.	Unity 3D	42
3.2.4.	Análisis y selección del modelo de visión por computadora para el sistema	43
3.2.5.	Ubicación de la punta de la pinza, usando marcadores	44
3.2.6.	Correlación del instrumental y los puntos de contorno de la herida en un mismo plano de referencia	48
3.2.7.	Arquitectura del sistema	49
3.2.7.1.	Arquitectura hardware del sistema	49
3.2.7.2.	Arquitectura Software del Sistema	50
3.3.	Implementación del sistema	51
3.3.1.	Esquema de la configuración cámara y captura de imagen de color y profundidad	52
3.3.2.	Esquema de selección ROI	52
3.3.3.	Esquema de la selección del punto de referencia	53
3.3.4.	Esquema de la segmentación	54
3.3.5.	Esquema de la localización de la herida, punto máximo y mínimo	55
3.3.6.	Esquema de la calibración entre píxeles de la imagen a color y profundidad	55
3.3.7.	Esquema del cálculo de las coordenadas x, y, z real por pixel	56
3.3.8.	Esquema de detección del instrumental quirúrgico	56
3.3.9.	Esquema de detección de la punta del instrumental	57
3.3.10.	Esquema de la función de transformación de coordenadas	57
3.3.11.	Esquema del enlace RosBridge	58
3.3.12.	Esquema de la virtualización de la escena quirúrgica	59
4.	Pruebas y Resultados	60
4.0.1.	Calibración de la cámara	60
4.1.	Cálculo del valor esperado medido por cinta milimétrica	61
4.1.1.	Prueba de precisión en función de la intensidad de luz y distancia de captura	62
4.1.1.1.	Prueba con la cámara en sentido horizontal	65
4.1.1.2.	Error absoluto en posición 'x','y' y 'z' del punto máximo y mínimo del contorno analizado	67
4.1.1.3.	Resultados y consideraciones a tener en cuenta en la puesta en marcha del sistema de localización de tejido epidérmico	67
4.2.	Prueba de localización y seguimiento de la punta del instrumental	69
4.3.	Prueba en conjunto, instrumental y herida	72

5. Conclusiones y trabajos futuros	74
5.1. Conclusiones	74
5.2. Trabajos Futuros	75
Bibliografía	76
Bibliografía	76
Anexos	80
A. Conexión Windows-Linux por medio de Rosbridge	81
A.0.1. Configuración Servidor:	81
A.0.2. Configuración Cliente:	82
B. Manual de usuario de la interfaz	85
C. Dispositivos de captura 3D	87
D. Emulación de la escena quirúrgica en Unity	90

Lista de Figuras

1.1.	Multiplicidad de sistemas coordenados.	10
1.2.	Sistema de navegación quirúrgica basado en realidad aumentada.	11
2.1.	Fases del proceso de la visión artificial al determinar una imagen	18
2.2.	Técnicas de iluminación	20
2.3.	Elementos del funcionamiento de la visión artificial al determinar una imagen	21
2.4.	Tipos de escáneres para la reconstrucción 3D	22
2.5.	Triángulo de exposición	24
2.6.	Clasificación modelos de visión por computadora por su funcionalidad	25
2.7.	Modelos de segmentación	27
2.8.	Representación de un punto en coordenadas homogéneas	28
2.9.	Transformación de coordenadas	29
3.1.	Desarrollo en cascada	31
3.2.	Imagen de realsense-viewer 45 grados: 1: Imagen de color; 2: Imagen de profundidad; 3: Infrarrojo izquierdo; 4: Infrarrojo derecho	35
3.3.	Imagen de realsense-viewer 90 grados: 1: Imagen de color; 2: Imagen de profundidad; 3: Infrarrojo izquierdo; 4: Infrarrojo derecho	36
3.4.	Cámara de profundidad D435	38
3.5.	Pinza laparoscópica	39
3.6.	Luxómetro	39
3.7.	Almohadilla de sutura	39
3.8.	Soporte móvil	40
3.9.	Lámpara	40
3.10.	Diagrama general	41
3.11.	Logo de Open CV-Python	42
3.12.	Logo de la herramienta Unity 3D	43
3.13.	Porcentajes de píxeles no correlacionados según la segmentación	44
3.14.	Representación geométrica de la pinza y la herida en un plano 2D	45
3.15.	Ubicación de los marcadores en la pinza	45
3.16.	Proyección del valor Vf añadido sobre el segmento de recta AB	46
3.17.	46
3.18.	47
3.19.	Esquemático de los planos de referencia	48
3.20.	Arquitectura hardware	49
3.21.	Arquitectura software	50
3.22.	Módulos del sistema	51
3.23.	Esquema de configuración de la cámara	52
3.24.	Esquema de selección ROI	53
3.25.	Región de interes seleccionada	53

3.26. Esquema de selección del punto de referencia	53
3.27. Punto de referencia seleccionado(cuadro rojo)	54
3.28. Esquema de la segmentación	54
3.29. Imagen segmentada	54
3.30. Esquema de la localización de la herida	55
3.31. Puntos máximo y mínimo de la herida	55
3.32. Esquema de la calibración entre píxeles de la imagen a color y profundidad	56
3.33. Esquema del cálculo de coordenadas x, y, z real por pixel	56
3.34. Esquema de la detección del instrumental	57
3.35. Detección de la pinza quirúrgica	57
3.36. Esquema de la detección de la punta del instrumental	57
3.37. Esquema de la función de transformación de coordenadas	58
3.38. Esquema del enlace RosBridge	58
3.39. Esquema de virtualización	59
4.1. Plantilla usada para calibrar dispositivo de captura paralelo a un plano de referencia según su distancia	61
4.2. Medidas del valor esperado, gráfica con resolución 1024x720	61
4.3. Distancias a la herida: 1: 17.1 cm; 2: 23.6 cm; 3: 28.6 cm	62
4.4. Desviación estándar a 500 lux	63
4.5. Distancias a la herida: 1: 17.1 cm; 2: 23.6 cm; 3: 28.6 cm	63
4.6. Desviación estándar a 1500 lux	64
4.7. Distancias a la herida: 1: 17.1 cm; 2: 23.6 cm; 3: 28.6 cm	65
4.8. Desviación estándar a 2000 lux	65
4.9. Distancias a la herida: 1: 17.1 cm; 2: 23.6 cm; 3: 28.6 cm	66
4.10. Desviación estándar	66
4.11. Ubicación de los elementos para el seguimiento de la pinza quirúrgica	69
4.12. Seguimiento de la pinza quirúrgica: 1: frame 1; 2: frame 2	70
4.13. Seguimiento de la pinza quirúrgica: 3: frame 3; 4: frame 4	70
4.14. error absoluto de la posición de la punta en los puntos 1,2,3 y 4 en mm	71
4.15. Error absoluto, distancia de la punta del instrumental al punto inferior y superior de la herida	71
4.16. Simulación de la escena en Unity	72
4.17. Ubicación del punto máximo, mínimo y recorrido de la pinza sobre la herida	72
4.18. Ubicación del punto máximo, mínimo y recorrido de la pinza sobre la herida	73
4.19. Ubicación del punto máximo, mínimo y recorrido de la pinza sobre la herida	73
4.20. Ubicación del punto máximo, mínimo y recorrido de la pinza sobre la herida	73
A.1. Archivo launch	81
A.2. Conexión ROS	82
A.3. Interfaz Unity	82
A.4. Objeto Rosconnector	83
A.5. URL de Linux	83
A.6. Conexión exitosa	84
B.1. Inicio de la interfaz	85

C.1. Kinect V1	87
C.2. Kinect V2	88
C.3. Cámara de profundidad D415	88
C.4. Cámara de profundidad D435	88
C.5. Cámara ZED	89
D.1. Etapas de la ejecución: 1: lanzamiento del roscore; 2: lanzamiento del archivo principal	90
D.2. Inicialización del roscore	91
D.3. Ejecución del archivo .py	91
D.4. Inicialización del programa en Unity	92
D.5. Escena quirúrgica	92
D.6. Cliente conectado	92
D.7. Temas	93

Lista de Tablas

1.1. Investigaciones	15
3.1. Comparación de dispositivos	34
3.2. Rango de la cámara	37
4.1. Desviación estandar en mm, lux 500	62
4.2. Desviación estandar en mm, lux 1500	64
4.3. Desviación estándar en mm, Lux 2000	64
4.4. Comparación en dispersión, cámara en vertical y horizontal; lux 2000 .	66
4.5. Valor promedio obtenido de una muestra de 1000 datos, lux 500, sentido vertical	67
4.6. Valor promedio obtenido de una muestra de 1000 datos, lux 1500, sentido vertical	67
4.7. Valor promedio obtenido de una muestra de 1000 datos, lux 2000. sentido vertical	67

Capítulo 1

Introducción

En el presente trabajo, se propone un sistema de localización de tejido epidérmico e instrumental quirúrgico semiautonomo, con el fin de determinar las condiciones que se deben tener en cuenta en el diseño de un sistema de visión por computadora, enfocado a la cirugía asistida por robot.

Para ello, se realizó una investigación de los modelos de visión por computadora encontrados en trabajos relacionados, con lo cual se construyó un marco de desarrollo sobre el cual se abordó el diseño del sistema. Como objetivos para esta investigación se presenta el de localizar una herida en un cuerpo de superficie suave que simule el tejido epidérmico, desarrollar un método de seguimiento de instrumental quirúrgico y evaluar el sistema conjunto de localización y seguimiento, considerando un marco de referencia espacial común.

Este trabajo está dividido en capítulos de la siguiente manera: en el primer capítulo se plantea el problema, se especifican los objetivos del trabajo, y se presenta un compendio de los trabajos previos relacionados; en el segundo capítulo, se detallan los conceptos teóricos tenidos en cuenta para el desarrollo de la investigación; en el tercer capítulo se presenta la metodología utilizada, en la cual se describen los requisitos funcionales y no funcionales, luego se diseña e implementa el sistema; en el cuarto capítulo se realizan pruebas y se toman resultados observando la precisión del sistema a los cambios de luminosidad y distancia; finalmente en el quinto capítulo se presentan las conclusiones y trabajos futuros.

1.1. Planteamiento del problema

En el ámbito de la ciencia médica, las intervenciones de tipo quirúrgico, suponen una acción invasiva sobre una estructura anatómica de alguna zona del cuerpo. En particular, las cirugías tienen como finalidad hacer parte de un tratamiento para la solución de un problema o ayudar para la confirmación de un diagnóstico previo [1].

En la actualidad, la convergencia de los avances en informática y comunicaciones con la medicina en general y la cirugía en particular, ha dado como resultado un acelerado proceso de informatización de todas las áreas de la medicina. En este ámbito, se destaca la cirugía robótica, donde el cirujano tele opera el instrumental [1]. Los robots tradicionales para este tipo de cirugías, han tenido gran aceptación en la comunidad científica, por las significativas ventajas conseguidas frente a la cirugía convencional, como: disminución del sangrado, el tamaño de las cicatrices y el dolor post-operatorio, así como el menor riesgo de infección y tiempo de estancia hospitalaria [2] [3].

Sin embargo, a pesar de las facilidades que permiten los robots quirúrgicos, manipularlos adecuadamente es complejo, especialmente en labores como sutura. La automatización de la sutura mediante robots quirúrgicos propone un enfoque para solventar la complejidad de dirigir al robot en este tipo de tareas. No obstante, actualmente existen limitaciones en cuanto a la correlación entre la posición de los actores de la escena quirúrgica respecto a un marco de referencia común, dichos actores son el tejido, el instrumental y el robot. La ubicación de cada uno de los actores en tiempo real, favorece al seguimiento de los cambios que se tengan en el espacio de trabajo. Siendo lo anterior, un desafío tecnológico en cuanto a precisión, orientación y ubicación del tejido; junto con el rastreo y el movimiento de instrumentos quirúrgicos operantes [4] [5]. En particular, el desafío para lograr la manipulación adecuada de suturas en epidermis, converge en limitaciones tecnológicas en cuanto a sistemas de visión con capacidad de virtualizar la herida en un marco de referencia común al robot operante [6].

Como ya se ha mencionado, para el control de los objetos del escenario quirúrgico, es necesario reconocer la importancia de utilizar sistemas de visión que trabajen en tiempo real. Esto debido a la complejidad de un proceso de sutura, en el cual las funciones de operación del sistema deben irse adaptando conforme al progreso del procedimiento [6], de ahí la necesidad de un sistema dinámico. En la actualidad existen tecnologías de visión que se podrían extrapolar, como por ejemplo las utilizadas en el desarrollo de carros autónomos, las cuales se fundamentan en Lidar¹; en algunos sistemas industriales autónomos de selección y organización de productos, se ha utilizado cámaras de profundidad estéreo; en sistemas de rastreo se han utilizado dispositivos IMU sobre cámaras de seguimiento estéreo o mono estéreo. Sin embargo, para el caso de la sutura, las investigaciones aún apuntan a resolver de forma versátil y menos compleja la carencia de ubicación espacial del robot en un marco de referencia común a la herida, teniendo en cuenta la precisión que se debe tener en procedimientos quirúrgicos y las implicaciones de la tecnología.

Con lo expuesto anteriormente, en el marco de la sutura semiautomática, este trabajo pretende buscar una solución al desafío encontrado, en el que se presenta la problemática de emparejar el espacio de referencia del robot con un tejido de epidermis en un marco de trabajo común, posterior a virtualizar una escena quirúrgica para posibilitar la interacción médico-robot de manera precisa, considerando las implicaciones espaciales y ambientales del escenario. De esta forma, se tiene la siguiente pregunta de investigación:

En el contexto de la sutura robótica, ¿qué condiciones debe poseer un sistema de localización de tejido epidérmico e instrumental quirúrgico para la automatización de un proceso de sutura y cómo evaluar el desempeño del mismo?

1.2. Trabajos relacionados

Teniendo en cuenta la necesidad de un sistema que sirva como guía para el cirujano, durante un procedimiento quirúrgico, por medio del cual pueda orientarse en su entorno con precisión milimétrica, permitiendo identificar la lesión a tratar y reconocer

¹Acrónimo del inglés LIDAR: Light Detection and Ranging o Laser Imaging Detection and Ranging, tecnología que permite determinar las distancia desde un emisor láser a un objeto o superficie utilizando un haz láser pulsado.

estructuras anatómicas vecinas que pueda dañar a su paso [7], se expone en la siguiente sección algunos trabajos de investigación que se han encontrado y tenido en cuenta para la realización del proyecto.

Pei-Lun Yan y otros [5], proponen un sistema de navegación quirúrgico basado en KRKM, (*Kernel-Reliability K-Means*), que es un algoritmo de agrupación para el seguimiento visual de objetivos. El propósito de este proyecto fue obtener la posición de los instrumentos quirúrgicos y del paciente, a partir de un sistema de visión compuesto por una cámara de profundidad estéreo Realsense 200, que consigue crear un espacio virtual 3D, siguiendo marcadores previamente fijados en ellos; lo cual converge en cierta medida al enfoque propuesto, en cuanto a la virtualización, pero contrasta en el uso de marcadores. Por otro lado, en esta investigación, la posición se visualiza a través de un método gráfico por computadora, lo cual coincide en un solo camino, robot-médico, con la orientación del estudio propuesto.

Como se puede observar a continuación, debido a la multiplicidad de sistemas de coordenadas, Ver figura 1.1, tanto para el paciente, como para la cámara realsense 200, la pantalla virtual 3D y los instrumentos quirúrgicos; se hace una vinculación de los sistemas de coordenadas a través de un algoritmo de fusión gradual con el cual lograron simplificar el proceso de posicionamiento general y específico del instrumental quirúrgico. Para esta investigación, el procedimiento en la obtención de las imágenes para el seguimiento no se obtiene a través de tomografías, siendo un factor a favor, debido a la radiación nociva que genera, lo cual puede afectar tanto al paciente como al cirujano. Los resultados obtenidos cuentan con un error de precisión de menos de 1 mm debido a los cambios de iluminación en el ambiente que afectan la cámara, así como errores en el algoritmo. Sin embargo, prevalecen desafíos como mejorar la precisión de la posición y el entorno experimental en los aspectos de hardware y software para satisfacer requerimientos médicos que tienden hacia procesos semi-autónomos.

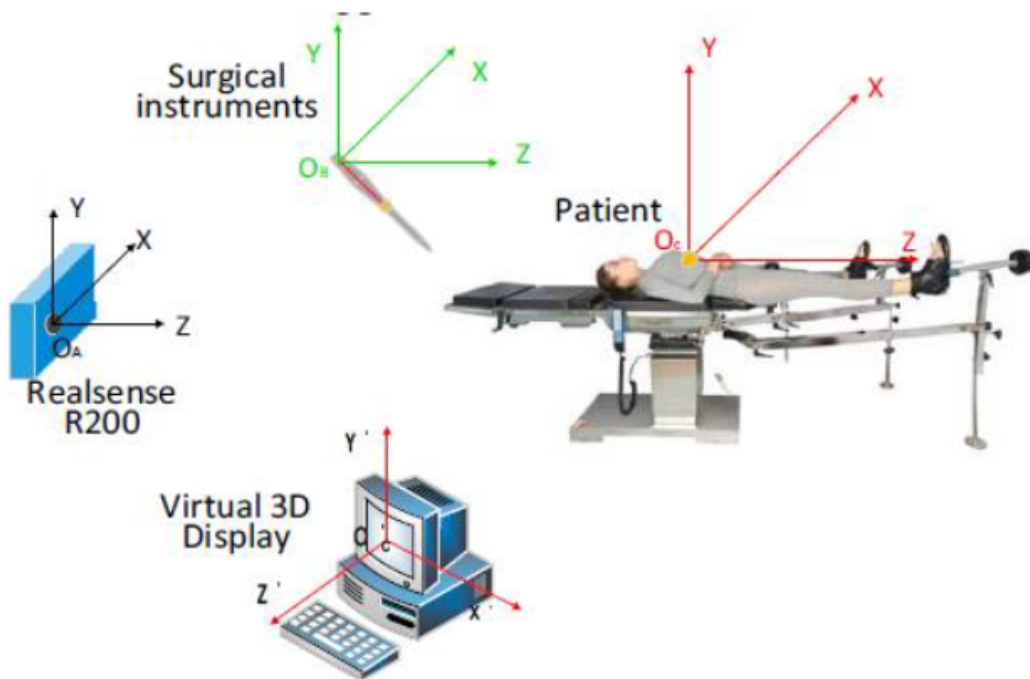


Figura 1.1: Multiplicidad de sistemas coordenados.
Fuente: [5]

Xiaojun Chen y otros [8], proponen desarrollar un sistema de navegación quirúrgica basado en realidad aumentada, con el objetivo de mejorar la seguridad y fiabilidad de la cirugía, abarcando la planificación quirúrgica preoperatoria, la reconstrucción 3D, el registro y el seguimiento intraoperatorio. En este caso, un cirujano usando las gafas de realidad virtual podrá obtener la imagen fusionada de estructuras anatómicas virtuales tales como vasos sanguíneos, nervios y tejidos blandos combinados con el escenario intraoperatorio del mundo real. Es importante recalcar que al hacer uso de diferentes sistemas de coordenadas se necesita una matriz de transformación para determinar la relación espacial entre el instrumento quirúrgico y el marco de referencia montado, véase figura 1.2. Para la evaluación del sistema realizan un experimento de verificación de precisión en tres objetos: Un sustrato de vidrio orgánico, una base de metal y un modelo craneomaxilofacial impreso en 3D; arrojando buenos resultados que cumplen con los requisitos clínicos. Resaltan que en la intervención quirúrgica los marcos de referencia con marcadores retro reflectantes, que tienen forma de esfera, se ubican en el paciente y en la pantalla para que esas posiciones espaciales puedan ser localizadas en tiempo real a través del dispositivo de seguimiento óptico Polaris Vicra². Destacan que algunas limitaciones de la navegación quirúrgica tradicional son superadas, como lo es que el cirujano ya no se ve obligado a cambiar entre la pantalla de la computadora y el escenario de la operación real. Además, mencionan que los resultados de los experimentos de verificación de precisión serían una buena alternativa para ser desarrollados en cirugía mínimamente invasiva.

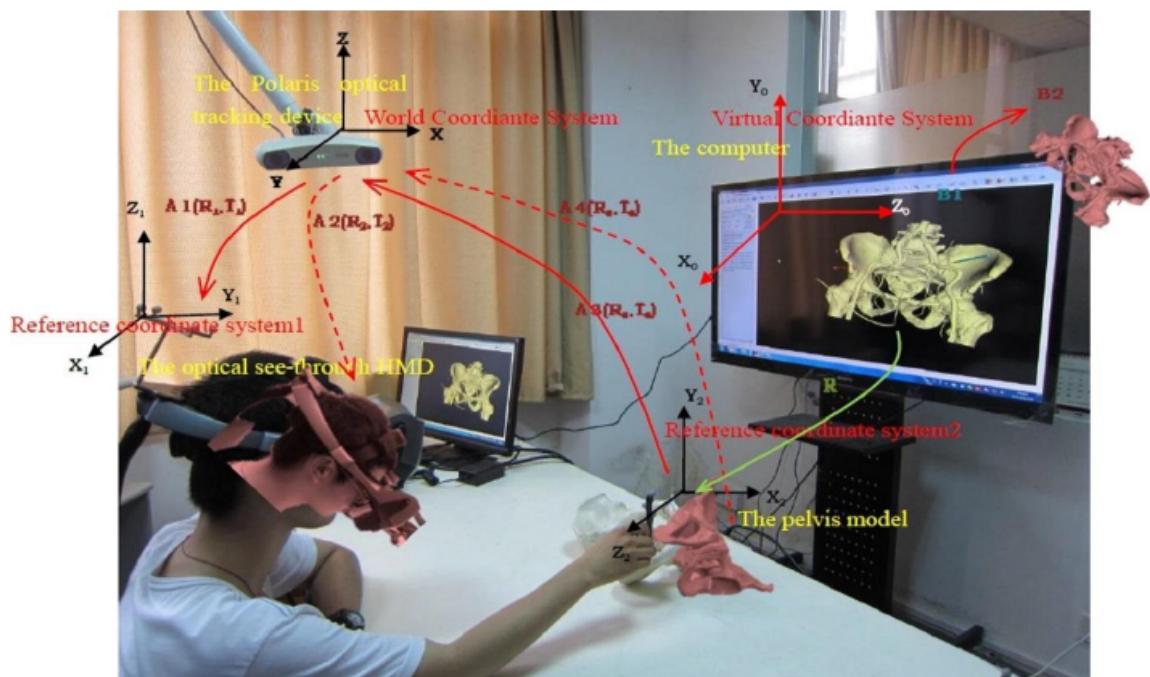


Figura 1.2: Sistema de navegación quirúrgica basado en realidad aumentada.

Fuente: [8]

Zhentian Zhou y otros [9], proponen un sistema de seguimiento óptico de bajo costo, el cual se basa en la teoría de visión binocular. El estudio se enfoca en el diseño de un sis-

²Polaris Vicra: Dispositivo de medición óptica que calcula la ubicación y orientación de un objeto o herramienta dentro de un sistema de coordenadas definido.

tema de seguimiento óptico que utiliza la cámara binocular llamada Bumblebee²³ para capturar la imagen de un sistema de marcadores emisores de luz, el cual está compuesto por 3 leds ópticos que se colocan en el instrumento quirúrgico y son sometidos a un procesamiento de imágenes, combinando este proceso con un algoritmo de seguimiento de espacio tridimensional; con lo cual se logra una alta precisión en el seguimiento de los instrumentos quirúrgicos. El estudio concluye que los resultados experimentales del sistema consiguen una alta estabilidad y precisión, permitiendo el desarrollo de una guía precisa para la navegación quirúrgica.

Por otra parte, Azad Shademan y otros [6] demuestran la cirugía autónoma supervisada in vivo de tejidos blandos en un entorno quirúrgico abierto, habilitado por un sistema de seguimiento visual en 3D que utiliza imágenes fluorescentes de infrarrojo cercano, y un algoritmo de sutura autónomo. Para ello mejoran el sistema compuesto por un robot autónomo de tejidos inteligentes (STAR), con ocho grados de libertad, el cual realiza varias tareas quirúrgicas, encontrándose equipado con tecnologías de imágenes inteligentes, con el propósito de determinar si un complejo procedimiento quirúrgico que actualmente solo se realiza por humanos podría ser realizado por un robot autónomo supervisado. En este trabajo se demuestra que el resultado de los procedimientos autónomos supervisados es superior a la cirugía realizada por cirujanos expertos; De igual forma, Azad Shadademan y colaboradores en [10] utilizan esta técnica de seguimiento visual ofreciendo la posibilidad de que el robot quirúrgico tenga una precisa colocación de la punta del instrumento en el tejido epidérmico. También, en [11], presentan a STAR, un sistema robótico que se guía por visión, el cual posee una herramienta de sutura laparoscópica accionada, haciendo posible la ejecución de suturas a partir de comandos basados en imágenes, además de lograr detectar ciertos marcadores mediante la técnica de imágenes fluorescente de infrarrojo cercano. Una de las falencias desarrolladas en este trabajo de investigación es la falta de un algoritmo que permita hacer una ubicación con sus respectivas coordenadas, dejando esta responsabilidad al cirujano, quien debe realizar la tarea manualmente. El estudio concluye que STAR es capaz de colocar suturas en un plano fantasma con precisión media de 0.5 mm con una desviación estándar de 0.2 mm, quedando dentro de los resultados esperados.

Damir Filko y otros [12], proponen un sistema para la detección, reconstrucción en 3D y segmentación de heridas crónicas usando el sensor Kinect-V2⁴, con el objetivo de realizar un estudio donde se caracteriza la lesión con una medición de buena exactitud, recurriendo a la utilización de diversos métodos para tal fin, ya que es un gran indicador del progreso en la curación. La medición de las heridas se divide en 2 enfoques principales: de contacto y sin contacto, el método de contacto requiere que las medidas tomadas sean en un directo contacto con la herida, para el método sin contacto se utiliza un dispositivo de grabación. Una de las ventajas del sistema de reconstrucción es la posibilidad de ver la herida en su totalidad desde una sola vista, permitiendo la localización en un área mayor y en superficies curvas. Todo el sistema ha sido desarrollado sobre C++ e implementado sobre una plataforma de computación en paralelo, lo que permite el funcionamiento en tiempo real. El estudio concluye que el sistema desarrollado es un preciso modelo tridimensional de alta resolución en color de

³Bumblebee2: Equipo de visión estéreo, el cual proporciona datos 3D de calidad y velocidad de procesamiento en tiempo real.

⁴Kinect-V2: Dispositivo que permite a los usuarios controlar e interactuar una consola mediante una interfaz natural de usuario, que reconoce gestos, comandos de voz, objetos e imágenes.

la herida segmentada, y sus propiedades medibles, incluyendo el perímetro, el área y el volumen, que puede utilizarse para determinar una terapia y un tratamiento correcto de las heridas crónicas.

Philip Edgcumbe y otros [13], proponen utilizar un proyector láser de captación en miniatura llamado Pico Lantern, cuyo funcionamiento se basa en la luz estructurada [18], para una orientación quirúrgica mínimamente invasiva desarrollada hacia la reconstrucción de la superficie, la realidad aumentada y la orientación en la cirugía laparoscópica. El enfoque de la luz estructurada sustituye a una cámara del sistema convencional de estereovisión por un dispositivo activo que proyecta un patrón codificado conocido en la escena. Mencionan que la reconstrucción de la superficie y las proyecciones de la realidad aumentada son más precisa debido a que el dispositivo se encuentra dentro del abdomen, posibilitando su libre movimiento en relación con el laparoscopio, por lo tanto, no se requiere una herramienta de seguimiento externo. El propósito es demostrar la viabilidad y la precisión de la reconstrucción de la superficie del tejido en 3D y la realidad aumentada usando el Pico Lantern. Adicionalmente, un experimento porcino in vivo muestra que puede utilizarse durante la cirugía para la reconstrucción de la superficie.

Artur Sagitov y otros [14], implementan procedimientos de sutura quirúrgica automática asistida robóticamente empleando el robot Kuka- intelligent industrial work assistant(iiwa)⁵ el cual incluye sensores de retroalimentación háptica que ayudarían en la interacción con los tejidos blandos. El propósito es presentar un marco para la detección automática de heridas y la planificación de la intervención en la automatización de las suturas. La investigación se enfoca en las suturas continuas e interrumpidas, teniendo en cuenta otros factores que afectan directamente el procedimiento. Sin embargo, los desafíos que persisten son la reconstrucción del perfil de la herida, así como ampliar la planificación a un espacio de tejido 3D utilizando técnicas de visión estereoscópica.

Yuichiro Hayashi y otros [15], proponen un sistema de navegación quirúrgica y su aplicación clínica a la gastrectomía laparoscópica para el cáncer gástrico, basados en el registro de puntos de referencia anatómicos en la superficie de la piel y la visualización de vistas virtuales laparoscópicas alineadas mediante un rastreador posicional 3D. Este sistema obtiene la información posicional y de orientación a través de un laparoscopio utilizando un sistema de rastreo óptico que usa marcadores pasivos reflectantes. El estudio concluye que hay una precisa identificación de la anatomía vascular como guía para la sujeción de los vasos antes de la gastrectomía total o parcial.

Durante la búsqueda de información se evidenció que existen muchas iniciativas en pro de contribuir a mejorar las cirugías robóticas y llevarlas a un enfoque de autonomía, es importante mencionar que el proyecto contrasta con esto debido a que se maneja un enfoque semiautónomo, siendo este el paso intermedio antes de conseguir tecnología totalmente autónoma, para cualquier procedimiento quirúrgico que se realice. Se destaca que los inconvenientes presentados en cada una de las tecnologías aplicadas son distintos y variados, creando una complejidad extra para el desarrollo de una solución definitiva que complemente hasta ahora las desarrolladas, ver tabla 1.1.

En síntesis, con diferentes tecnologías como la luz estructurada, la visión estereoscópica, el sensor Kinect, los marcadores infrarrojos, entre otros; pueden ser utilizados en el desarrollo del trabajo, por tanto, es parte fundamental conocer sus ventajas y desventajas para así lograr implementar la mejor solución. Como se ha mencionado

⁵Kuka-iiwa: Robot con capacidad sensitiva.

previamente, con la multiplicidad de retos en la detección y seguimiento de tejidos para un entorno quirúrgico, se hace necesario trabajar en un sistema para desarrollar una virtualización de una escena quirúrgica, ayudando en gran medida a la evolución de la robótica médica hacia la autonomía de la misma.

Tecnología	Segmentación	Seguimiento		Registro (fase de reconstrucción)	Proposito
		Calibración	Detección, puntos de referencia		
Triangulación Cámara estereo pasivo. Correspondencia de puntos clave del objeto en ambas cámaras	Aplica un filtro de borde novedoso para delinear el contorno del hilo de sutura Segmentación del contorno del hilo de sutura en tres capas	Matriz fundamental real en la punta se calcula utilizando los datos de calibración del tablero de ajedrez, método de Zhang.	Se utiliza un algoritmo de computación iterativa impulsada por la posición inicial dada por clics de mouse.	No	Seguir un hilo de Sutura y ubicarlo en un espacio tridimensional
Cámara de profundidad de tipo ToF (kinect v2)	Surfeles, basado en el método de crecimiento regional, adheriendo a este la consideración de la información de color de manera uniforme.	No	Histogramas concatenados de cada canal de color en el espacio de color HSV y algoritmo k-vecino más cercano	No	Detectar, segmentar, reconstruir y medir una herida
Cámara de seguimiento polaris (Estereo activo)	Para tejido duro y blando específicamente: Método combinado de crecimiento de región y umbral, método de crecimiento de región semiautomático. Si se presenta sobresegmentamiento, se usa modificación manual.	Calibración de pivote y el Kit de herramientas de cirugía guiada por imágenes (IGSTK, una biblioteca de software de código abierto C++)	Uso de marcadores retroreflectantes	El registro de puntos fiduciales y el registro de superficie	Investigativo
Cámara estereo con filtro de infrarrojo añadidos a cada lente, para eliminación de interferencia de luz ambiental	Se hace de manera Hardware utilizando filtro de infrarrojos. Sobre los puntos de luz infrarroja observada se aplica RG y GC.	Método de calibración de Zhang	Optimización de lentes infrarrojos opacos en el instrumento quirúrgico. Para extraer la coordenada de píxeles del centro de un marcador, se utiliza el método de crecimiento de la región (RG) para eliminar el punto singular y luego se adopta el método del centroide gris (GC) para obtener la coordenada de píxeles del centro del marcador.	algoritmo hace coincidir los marcadores en las imágenes de la cámara izquierda y derecha en función de la secuencia de sus posiciones espaciales	Combinar el seguimiento de marcadores de luz infrarroja y métodos de procesamiento de imágenes para la elaboración de un algoritmo de seguimiento (3-D), de alta precisión, para los instrumentos quirúrgicos.
Sistema estereo activo separado. Por un lado se tiene un estereoscopio y por otro un proyector de luz estructurada llamado Pico Lantern.	No	Calibración del proyector convencional, kit de herramientas de calibración de la cámara de Bouguet basada en el algoritmo de Zhang	Detección de las esquinas del tablero de ajedrez proyectadas y del tablero de ajedrez añadido al dispositivo picolatern utilizando el algoritmo de detección de la esquina del tablero de ajedrez (Bouguet, 1999)	La reconstrucción 3D se consigue recuperando la información proyectada por el Pico Lantern a través de un algoritmo de detección de esquinas de un tablero de ajedrez y haciendo una triangulación entre el Pico Lantern, el Estereoscopio y el objeto a mapear	Recuperar información de profundidad de superficie mediante el mapeo de órganos internos

Tabla 1.1: Investigaciones

Fuente: Elaboración propia

1.3. Objetivo

1.3.1. Objetivo General

- Proponer un sistema de localización de tejido epidérmico, que involucre elementos de una escena quirúrgica, para la automatización de una sutura.

1.3.2. Objetivos Específicos

- Adaptar un método de localización de cuerpos de superficie suave que emulen el tejido epidérmico utilizando la información de profundidad obtenida por un sistema de visión.
- Desarrollar un sistema de rastreo de elementos de escena quirúrgica.
- Evaluar el sistema de localización con un tejido emulado por una almohadilla de entrenamiento quirúrgico, considerando un marco de referencia espacial común.

Capítulo 2

Marco teórico

En este capítulo se presentan los diferentes conceptos teóricos que describen y componen un sistema de visión en la robótica quirúrgica, los cuales permitirán establecer los principios necesarios para cumplir con los objetivos propuestos de la presente investigación.

2.1. Visión por computadora en la robótica quirúrgica

La robótica quirúrgica es un sub-campo de la robótica médica que tiene como propósito permitir que el cirujano realice procedimientos quirúrgicos, soportados por la asistencia de un brazo robótico, lo que conlleva a hacer procedimientos más precisos además de posibilitar la teleoperación. En la actualidad, la cirugía asistida por robot (RAS), depende totalmente de los movimientos manuales que realiza el cirujano [16], sirviendo como una extensión de sus manos. Por lo cual, gran parte del desarrollo de esta técnica se ha dado en el marco de MIS¹. Sin embargo, hoy en día, gracias al creciente desarrollo de sistemas de aprendizaje autónomo, navegación y visión por computadora, se ha proyectado como siguiente paso para RAS, realizar procedimientos donde el robot no dependa totalmente del cirujano. En este sentido, las investigaciones se han enfocado en resolver procedimientos semiautónomos superficiales, como lo son cortar un hilo o suturar una herida externa, entre otros [12] [17] [18] [19].

Como se mencionó, la evolución en los sistemas de visión por computadora ha sido uno de los factores determinantes para acortar la brecha hacia la ejecución de procedimientos semiautónomos. La visión por computadora o visión artificial se puede entender como la ciencia de programar un computador para procesar imágenes o videos e incluso entenderlos [20]; también es definida como la transformación de datos desde un fotograma o videocámara en lo que puede ser una decisión o una nueva representación [21]. En este sentido, la visión por computadora se muestra como uno de los factores más relevantes hacia la automatización de procedimientos quirúrgicos, dado que permite la abstracción, entendimiento y toma de decisiones en un entorno espacial.

La figura 2.1, presenta las fases del proceso de funcionamiento de la visión por computadora en un diagrama general no concatenado, en el que el uso de las fases no depende

¹Acrónimo del inglés MIS: Minimally invasive surgery, esta técnica permite realizar cirugías a través de pequeñas incisiones, por las que se introduce un laparoscopio para la navegación interior y diminutos instrumentos quirúrgicos guiados para la ejecución del procedimiento.

entre sí, ni tampoco son obligatorias. En este sentido, la ejecución de cada fase solo se ve influenciada por el entorno en el que se desarrolla el sistema y su alcance [22].

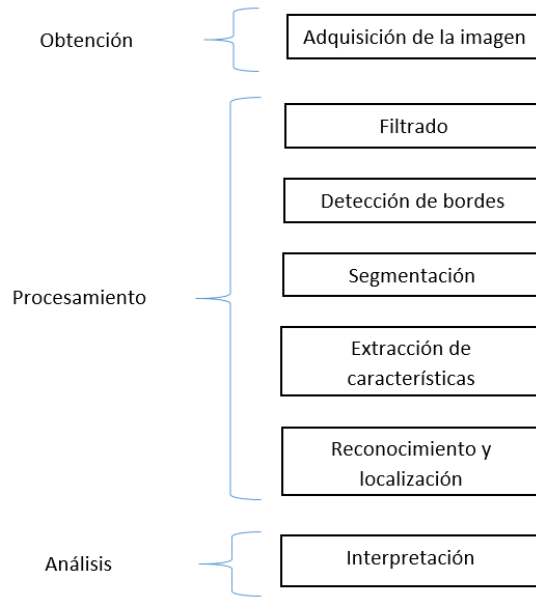


Figura 2.1: Fases del proceso de la visión artificial al determinar una imagen
Fuente: Elaboración propia

En el campo de la robótica quirúrgica, para la fase de adquisición de imagen, los datos que caracterizan la escena se obtienen a través de dispositivos ópticos que capturan información de la imagen en un plano bidimensional RGB² y tridimensional; datos que se utilizan para localizar espacialmente los instrumentos quirúrgicos y la zona a tratar con respecto al dispositivo de captura.

Continuando con las fases que hacen parte del procesamiento, se puede considerar la fase de filtrado como no obligatoria, pudiendo llegar a ser necesaria conforme al grado de disparidad en los píxeles y la definición requerida; con respecto a las fases de detección de bordes y segmentación, estas se hacen necesarias al determinar el contorno de la región a tratar, para luego separar la zona de interés del campo de visión general; la fase de extracción de características puede ir orientada al uso de marcadores³ o estar relacionada con procesos de machine learning, llegando a ser muy práctica, ya que extiende las posibilidades de autonomía en la detección y seguimiento de objetos sin marcadores; la fase de reconocimiento y localización, interpola las fases de adquisición, detección, seguimiento y extracción de características. Aquí, el procesamiento de la imagen sigue una ruta cuyo propósito tiene compactar los datos de profundidad, con la imagen RGB plana, considerando tanto el acotamiento de la región como el seguimiento de objetos, utilizando marcadores o algoritmos de aprendizaje autónomo.

Partiendo de lo anterior, la visión por computadora en el contexto de la cirugía robótica semiautónoma, se puede definir como una ciencia que tiene como propósito

²Acrónimo de RGB, en inglés red, green, blue; es un modelo de colores en el cual los descritos se suman de diversas formas para producir una amplia gama de tonalidades

³Uso de características físicas artificiales de gran contraste, como puntos fluorescentes, sobre los objetos que se quiere localizar o seguir.

comprender visualmente la escena quirúrgica acotada, a través de la obtención, procesamiento y análisis de información óptica. Por otro lado, si se piensa en la creación de un sistema de visión por computadora, es necesario estudiar cada uno de los elementos internos que lo compondrían, así como también tener en cuenta los diferentes elementos externos que interactúan con el sistema; elementos que están relacionados respectivamente a aspectos hardware-software y a condiciones ambientales del espacio de trabajo.

2.1.1. Elementos externos de un sistema de visión por computadora enfocado a RAS semiautónomo en un procedimiento de sutura epidérmica superficial

Entre las consideraciones externas de un sistema de visión, se encuentran las condiciones de humedad, temperatura y luminosidad, las dos primeras están vinculadas al buen rendimiento del hardware, y la tercera está asociada a la correcta captura del mapa de profundidad y la imagen RGB de la escena (Mapa de profundidad, imagen RGB), lo cual es sumamente importante, teniendo en cuenta las posibles sombras que puedan impedir un procesamiento de imagen favorable.

Humedad y temperatura: Partiendo del supuesto en el que el sistema se desarrolle bajo un ambiente quirúrgico, propiamente dicho un quirófano, la norma sugiere que el ambiente satisfactorio para llevar a cabo una cirugía, debe poseer una humedad relativa al 60 % y una temperatura entre 18 y 24 °C. Condiciones para las cuales las diferentes tecnologías de captura y equipos de procesamiento operan de buena forma [23].

Iluminación de la escena: Con respecto a la iluminación, la unidad de medición es el Lux. Según la norma, en un quirófano, específicamente en la mesa de operación, se establece un mínimo de 500 Lux hasta un máximo de, 100000 Lux, tope que se compararía a la luz emitida por el sol a medio día y cielo despejado. Como ya se había mencionado, la necesidad de planificar una buena iluminación tiene sentido al considerar las posibles sombras que dificultarían el correcto tratamiento de la escena poscaptura [24].

Aunque en el caso de un quirófano, el escenario está bastante iluminado, se pueden presentar sombras que intuitivamente el ojo humano podría evitar o encontrar forma alguna para establecer línea de visión directa. Sin embargo, en el caso de la cirugía robótica, implementar estos mecanismos podría ser una tarea complicada. Por ello, la forma más fácil de resolver este problema, en un principio, es buscar la mejor posición y ángulo de proyección de luz hacia la región observada, teniendo en cuenta que en un escenario superficial las sombras pueden originarse principalmente por oclusión de la fuente de luz con el brazo robótico móvil u otro objeto en movimiento en la escena.

Entre los métodos de iluminación más utilizados en aplicaciones de visión artificial se encuentran: La técnica de **iluminación frontal**, en la cual la cámara se ubica frente al objeto y en la dirección de la luz, siendo útil en superficies con pocos reflejos; la técnica **iluminación lateral**, en la cual la fuente de luz se ubica a un lado del objeto creando un ángulo de 90 grados entre la cámara y el objeto a iluminar; la técnica de **iluminación por campo**, en la que la fuente se ubica de forma lateral al objeto en cuestión, iluminando la escena en dirección perpendicular a la cámara; la técnica de **iluminación por contraste**, en la que se ubica al objeto entre la cámara y la fuente de iluminación, permitiendo reconocer la silueta del objeto por contraste; la técnica de **iluminación coaxial** o iluminación sobre el mismo eje, en la cual se emite una luz de

manera lateral sobre un espejo semitransparente provocando que la luz se desvíe en la misma dirección al eje de la cámara, su principal uso se da en la iluminación de objetos reflectantes; la técnica de **iluminación difusa**, en la cual la fuente de iluminación emplea una estructura para encapsular la luz o utiliza la iluminación con una longitud de onda determinada. Debido a esto, la cámara en particular debe contar con un filtro para esa longitud de onda [25]. Todos los tipos de iluminación se pueden observar en la figura 2.2

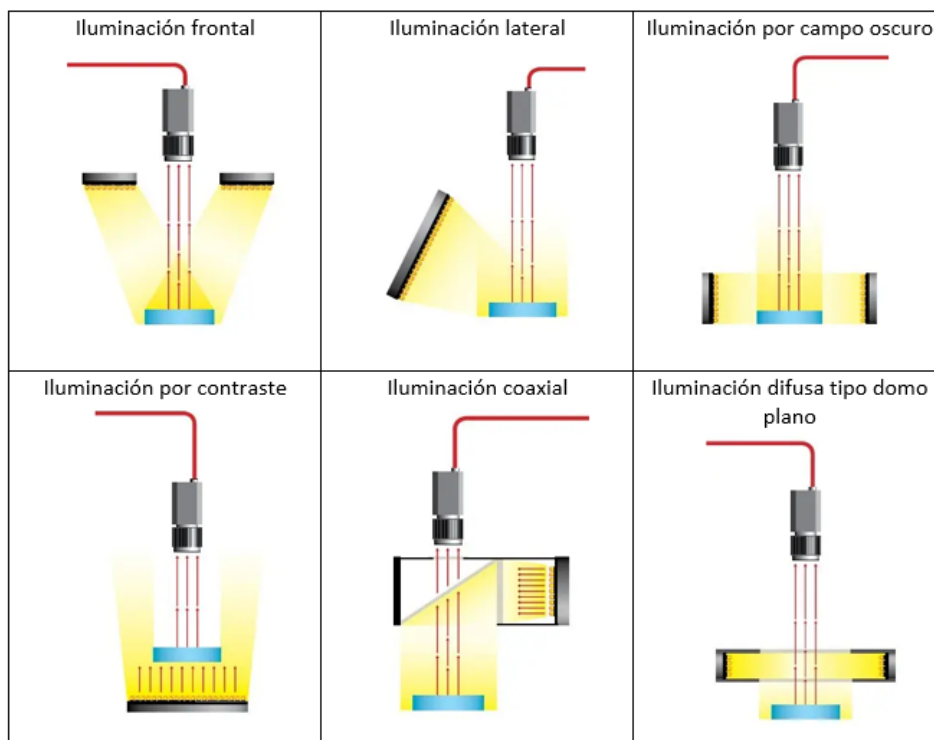


Figura 2.2: Técnicas de iluminación
Fuente: [25]

2.1.2. Elementos internos de un sistema de visión por computadora enfocado a RAS semiautónomo en un procedimiento de sutura epidérmica superficial

Por otro lado, como ya se había mencionado, la disposición de los elementos, hardware y software que componen un sistema de visión por computadora depende del alcance y propósito del proyecto. De forma específica, estos elementos se presentan en correspondencia al uso de las fases vistas en la figura 2.1. De esta manera, como se observa en la figura 2.3, la función de obtención vinculada a la fase de adquisición de la imagen surge del elemento de investigación que considera la técnica y tecnología hardware de captura 3D utilizada.

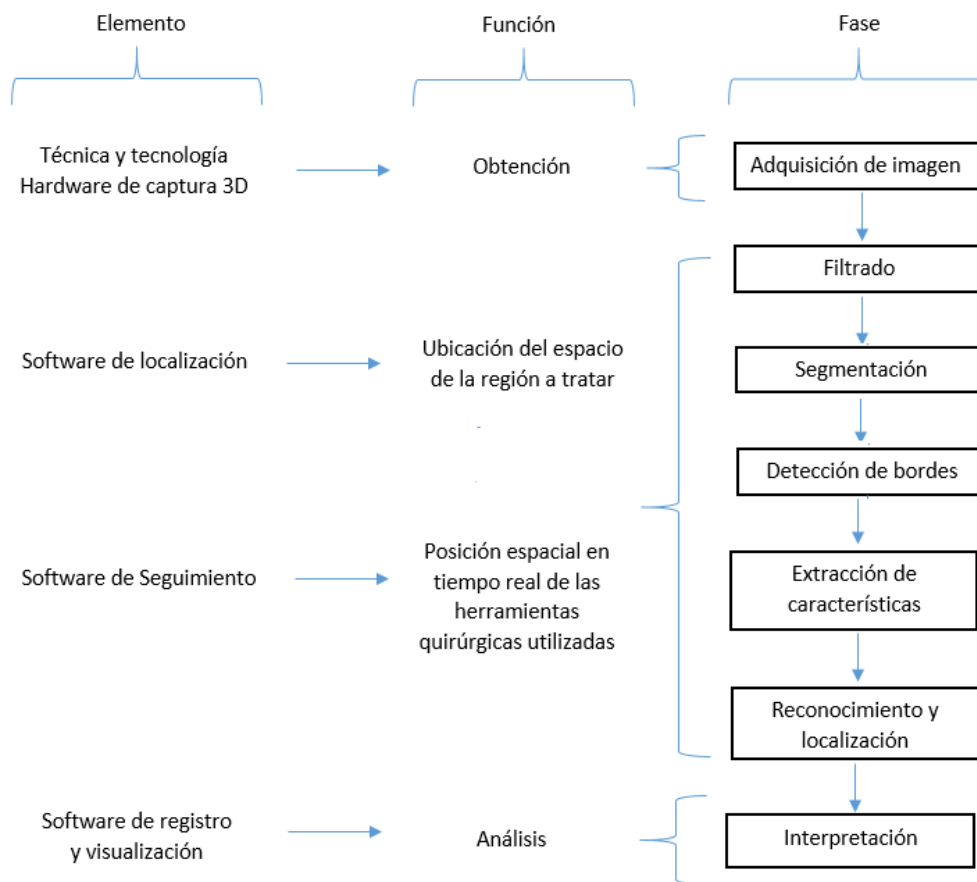


Figura 2.3: Elementos del funcionamiento de la visión artificial al determinar una imagen

En este sentido, para el caso concreto de un sistema de visión para intervención quirúrgica superficial, el sistema debe poder conocer la posición de las herramientas quirúrgicas (pinzas, tijeras) y el espacio de la región a tratar, por medio de información de profundidad capturada a través de cierta tecnología desarrollada con una técnica específica de abstracción espacial. A continuación se describirán las técnicas de reconstrucción 3D más importantes.

2.2. Técnicas de reconstrucción 3D

Como se observa en la figura 2.4, existe una división básica entre las técnicas de reconstrucción 3D, por un lado, se encuentran las técnicas que no requieren tener contacto físico con la superficie a escanear, y, por otro lado, las que requieren interactuar con el medio físico directamente. Los métodos basados en contacto por lo general utilizan un brazo robótico con una punta sensible al tacto, usándose principalmente en el control de procesos de fabricación. Por otro lado, las técnicas que no requieren contacto con la superficie, utilizan ondas electromagnéticas como ondas de radio, luz visible, infrarrojo y rayos x; estas técnicas se encuentran enfocadas en el desarrollo de sistemas de localización, posicionamiento, visualización clínica computarizada y navegación autónoma.

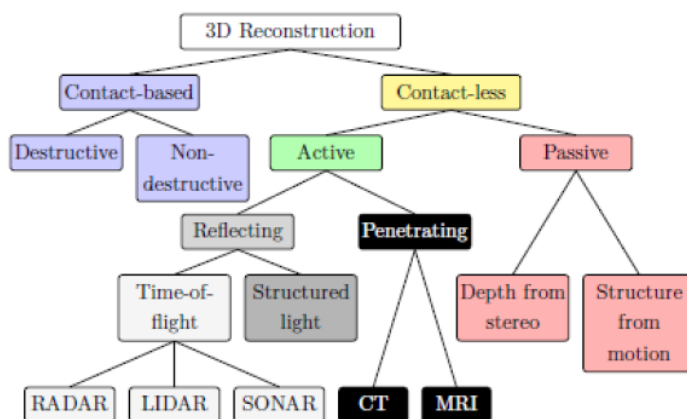


Figura 2.4: Tipos de escáneres para la reconstrucción 3D
Fuente: [26]

De igual forma, las técnicas sin contacto se dividen en técnicas activas y pasivas, las primeras se caracterizan por emitir algún tipo de señal electromagnética que se refleja, deforma o penetra sobre la superficie u objeto. Esta señal es posteriormente recuperada, procesada y analizada para determinar la geometría del objeto en cuestión. Entre estas técnicas se encuentran el tiempo de vuelo (TOF), luz estructurada, imagen por resonancia magnética (IRM) y por tomografía computada (TM) [26]. Por otra parte, las técnicas pasivas no emiten ninguna señal, estas utilizan la geometría de la escena para capturar la profundidad de un punto observado por medio de la triangulación de dos o más posiciones de observación, creadas con el movimiento de la cámara; proceso simplificado al utilizar dos cámaras⁴. A continuación, se describen con detalle algunas de las técnicas nombradas anteriormente.

Luz estructurada: Se utiliza en sistemas compuestos por un proyector y 1 o 2 cámaras, la técnica consiste en irradiar un patrón de luz conocido en el objeto o superficie a escanear, para que las cámaras registren y analicen la deformación del patrón por la geometría del objeto, creando una secuencia temporal con los diferentes valores de brillo. Una de las ventajas es su velocidad, ya que se puede escanear múltiples puntos eliminando el problema por deformación. Entre sus desventajas se encuentran la mala detección que tienen sobre las superficies reflectantes o transparentes, además no se

⁴Al utilizar dos cámaras simultáneamente se puede hacer una triangulación entre los lentes y el punto observado sin la necesidad de mover la herramienta de captura

recomienda utilizarlo para escenas con objetos de grandes dimensiones debido a que el sistema requiere conocer la localización exacta del proyector, por lo cual deberá permanecer fijo, digitalizando solo la parte iluminada. [26]

Tiempo de vuelo: Esta técnica calcula la distancia a la cual se encuentra un punto en el espacio, emitiendo un pulso de luz y calculando el tiempo que este invierte en ir desde el emisor hasta el punto y desde este hasta el receptor; la reconstrucción de una superficie se da conforme se recibe una secuencia a gran velocidad de estos puntos. Por consiguiente, la exactitud de los sistemas que utilizan esta técnica, depende de la precisión con la que se pueda medir el tiempo que tarda el pulso [26].

Visión estereoscópica: La técnica de visión estereoscópica modela espacialmente un punto o una superficie, a través de un procesamiento geométrico de los datos capturados, que se asemeja al proceso visual que realiza una persona cuando enfoca un objeto a través de sus ojos, los cuales captan desde posiciones ligeramente distintas un mismo objeto (lo que se conoce como disparidad ocular), brindando la percepción de profundidad o tercera dimensión. De esta manera, el proceso de estereovisión se da cuando el cerebro fusiona las dos imágenes recibidas en una sola. En el caso de un sistema de visión artificial que hace uso de visión estereoscópica para la reconstrucción espacial, la percepción de profundidad se da por medio de un procesamiento geométrico donde las imágenes capturadas desde dos distintos lentes terminan por compactarse en una sola imagen.

Visión estéreo activa: Esta técnica utiliza el mismo principio de la visión estéreo, con la característica de tener una fuente de luz infrarroja que emite un patrón conocido, el cual se deforma sobre los objetos y luego es reconocido por un sistema de cámaras dual. La proyección de luz estructurada sobre el escenario, mejora la capacidad del sistema de cámara estéreo para determinar profundidad, puesto que el patrón de infrarrojos estático aumenta la textura de los objetos en la escena, favoreciendo la abstracción de profundidad sobre elementos de baja textura. Además, brinda la posibilidad de captura de profundidad en un ambiente con nula luminosidad.

2.3. Descripción de las características de la cámara Realsense D435

En este apartado se presentan algunos conceptos generales sobre la toma de imágenes y se describen las características de funcionamiento configurables para la cámara RealsenseD435 y los ajustes hechos para mejorar la captura de profundidad cercana, para lo cual se consideró la exposición del sensor de imagen, el cambio en la disparidad de las cámaras estéreo y la resolución.

Exposición: Una de las características más importantes, para la correcta captura de una escena a través de una cámara, es la exposición. La cual determina en términos generales la cantidad de luz que entra al sensor. Esta característica está vinculada con tres parámetros: velocidad de obturación, apertura del diafragma y sensibilidad del sensor. A continuación se detalla cada uno de estos conceptos.

Velocidad de obturación: Por un lado, la velocidad de obturación medida en ms

indica el tiempo que el obturador permanece abierto. De esta manera, más tiempo significará mayor exposición que a su vez se traducirá en capturas con mayor percepción de luminosidad. Sin embargo, aumentar este parámetro implica prolongar el tiempo de toma de luz en el sensor, ocasionando que el sensor capture las variaciones de luz, conforme al desplazamiento de los objetos en el escenario, provocando en la imagen la percepción de movimiento.

Apertura del diafragma: Por otro lado, la apertura del diafragma indica el tamaño de la abertura por la cual pasa la luz desde el lente hasta el sensor. Por lo tanto, entre más grande sea tal abertura, mayor cantidad de luz llegará al sensor, logrando obtener mayor iluminación en la imagen. Además de esto, la apertura del diafragma es inversamente proporcional a la profundidad de campo, es decir, entre mayor sea la apertura del diafragma, menor profundidad de campo se podrá captar.

Sensibilidad ISO: La sensibilidad ISO, se presenta en el SDK de la cámara con el parámetro de ganancia, aumentar este valor incrementará la energía por pixel de la captura, con lo que la imagen tendrá una apariencia más luminosa, sin embargo, también aumentará el ruido en esta, siendo dañina en la mayoría de los casos, por esa razón se recomienda tener la sensibilidad ISO lo más bajo posible.

La figura 2.5, resume lo dicho, ubicando los tres factores descritos en una representación conocida como triángulo de exposición.



Figura 2.5: Triángulo de exposición
Fuente: [27]

La cámara de profundidad D435 ofrece una exposición automática; sin embargo, existe en el SDK la posibilidad de configurar la exposición variando la velocidad de obturación y sensibilidad ISO. Como ya se había mencionado, el incremento de la sensibilidad o ganancia del sensor ocasiona un aumento en el ruido de la imagen, por lo que es aconsejable trabajar con el mínimo valor permitido, teniendo en cuenta que factores externos pueden suplir la necesidad de luminosidad en la captura. Por otro lado, menor velocidad de obturación presentará imágenes más oscuras, pero más nítidas.

La cámara realsense viene por defecto con una distancia mínima de 16.08 cm; sin embargo, en las especificaciones del dispositivo de la tabla 3.1 se presenta como distancia mínima, un valor de 10.5 cm. Para conseguir tal valor se debe configurar de forma avanzada la cámara, variando un parámetro llamado *disparty-shift*, con el cual digitalmente se puede hacer un desplazamiento de la cámara derecha hacia el interior de la cámara izquierda, con lo que la correlación de las imágenes cambia su punto de triangulación a uno más cercano, logrando una distancia mínima de captura menor; lo que a su vez reduce de manera inversa y exponencial la distancia máxima de captura.

2.4. Modelos de visión por computadora

Los modelos de visión por computadora se pueden clasificar en tres ramas según su funcionamiento, en modelos de clasificación, detección y segmentación, como se observa en la figura 2.6. De esta manera, los modelos de clasificación toman una o varias imágenes y asigna una clase correspondiente a cada una de ellas; en este sentido, normalmente a una imagen le corresponde una sola clase. Los modelos con enfoque a detección de objetos, muestran las clases y las coordenadas donde se encuentra el objeto detectado. Por último, los modelos en segmentación buscan determinar los puntos de los píxeles correspondientes a cada clase segmentada.

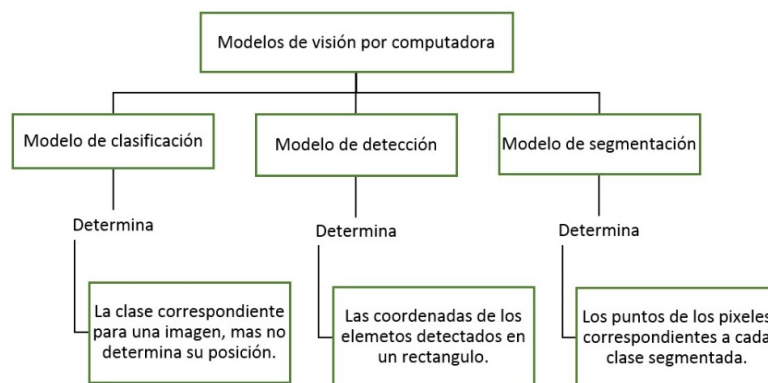


Figura 2.6: Clasificación modelos de visión por computadora por su funcionalidad
Fuente: Elaboración propia

2.5. Software de localización

Uno de los principales desafíos que surgen al momento de construir un sistema de visión por computadora, es el de implementar un método que permita localizar la región u objeto sobre el cual se va a trabajar. En el caso específico de un sistema de visión

artificial enfocado al proceso de automatización de sutura, el desarrollo del software de localización tiene como objetivo determinar la posición de los puntos que conforman el contorno de la herida, como región de interés; así como también la ubicación de los instrumentos que interactúan con la escena.

Acorde a lo anterior y como se observó en la figura 2.3, las fases vinculadas a la localización del contorno de la herida e instrumental quirúrgico son: filtrado de imagen, segmentación y extracción de características. Teniendo como eje central la segmentación, en la figura 2.7 se exponen los métodos más relevantes, los cuales están acompañados de diferentes algoritmos. De esta manera, se tienen algoritmos de baja complejidad como la umbralización, hasta algoritmos de segmentación basados en aprendizaje profundo. En el caso de los algoritmos de segmentación por umbralización, el umbral de segmentación se puede seleccionar utilizando diferentes métodos como el método de histogramas a color o el método de umbral adaptativo por Otsu.

2.5.1. Método de Otsu

Es un método para la selección del umbral más apropiado, el cual se obtiene tomando los puntos para los cuales la varianza es mínima dentro de cada segmento, pero máxima entre segmentos diferentes. De esta manera, se calcula el cociente entre ambas varianzas y se busca un valor umbral, en donde el cociente sea máximo [28].

Para la segmentación asistida o semiautomática, se tienen dos algoritmos principales: El algoritmo Watershed y el algoritmo Grabcut; aunque los dos se encuentran en el campo de segmentación asistida, la base de sus algoritmos es muy diferente.

2.5.2. Algoritmo Watershed

Por un lado, el algoritmo watershed utiliza el histograma de la imagen en escala de grises para representarla como una superficie topográfica [29], donde la intensidad de los niveles de gris indica picos o valles, en función de lo anterior el algoritmo ubica semillas en los puntos donde hay valles. Desde estos puntos o etiquetas el algoritmo va incrementando su región hasta llegar al límite de las otras regiones incrementadas, en los puntos de unión se construyen barreras. De esa manera, las barreras creadas en este proceso son la segmentación de la imagen.

2.5.3. Algoritmo Grabcut

Por otro lado, el algoritmo de grabcut basa su proceso de segmentación en representar la imagen como un grafo de flujo, y luego aplicar un algoritmo de corte mínimo en el grafo para separar la imagen de fondo de la región de interés [30].

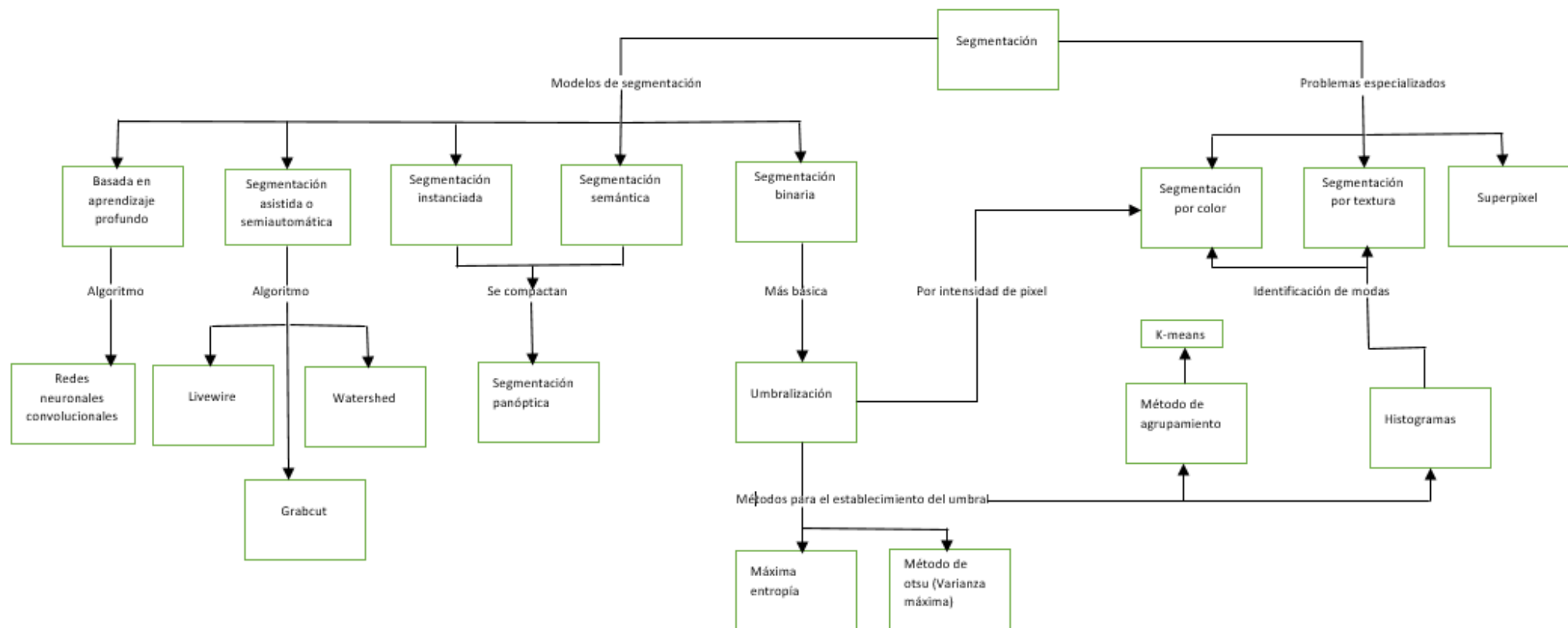


Figura 2.7: Modelos de segmentación
Fuente: Elaboración propia

2.6. Transformaciones homogéneas

2.6.1. Coordenadas homogéneas

Para representar un punto en el espacio se necesitan cuatro elementos, como se puede ver en la figura 2.8, 'x' y 'y' determinan las coordenadas en un plano 2D, mientras que 'z' determina la altura en un espacio 3D. Con el objetivo de crear la matriz de transformación de coordenadas homogéneas se debe añadir un elemento al vector de posición, este elemento se conoce como factor de escalado, y generalmente es unitario.

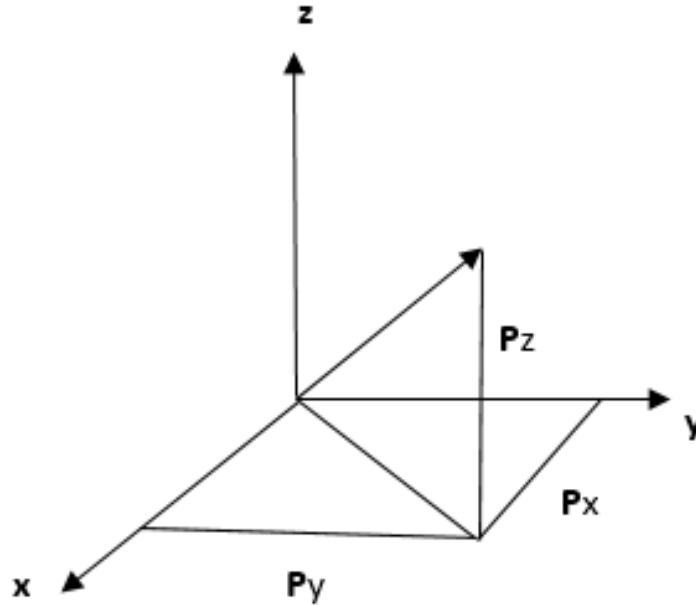


Figura 2.8: Representación de un punto en coordenadas homogéneas
Fuente: [31]

La matriz 2.1 corresponde a la representación de un punto en el espacio.

$$\mathbf{p} = [P_x \quad P_y \quad P_z \quad 1]^T \quad (2.1)$$

Para la representación de una dirección también son utilizados cuatro elementos, pero los primeros 3 elementos son vectores de dimensión 3×1 , donde cada uno de ellos representa la rotación del punto final en x, y, z, respecto a los ejes x, y, z, originales. Esta rotación se define por la matriz 2.2 [31].

$$\mathbf{u} = [u_x \quad u_y \quad u_z \quad 0]^T \quad (2.2)$$

Al expandir cada vector, se obtiene la matriz 2.3.

$$\mathbf{u} = \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \\ 0 & 0 & 0 \end{bmatrix} \quad (2.3)$$

Donde:

s_x : Rotación del eje x actual respecto al eje x anterior.

s_y : Rotación del eje x actual respecto al eje y anterior.

s_z : Rotación del eje x actual respecto al eje z anterior.

n_x : Rotación del eje y actual respecto al eje x anterior.

n_y : Rotación del eje y actual respecto al eje y anterior.

n_z : Rotación del eje y actual respecto al eje z anterior.

a_x : Rotación del eje z actual respecto al eje x anterior.

a_y : Rotación del eje z actual respecto al eje y anterior.

a_z : Rotación del eje z actual respecto al eje z anterior.

2.6.2. Transformación de coordenadas

De acuerdo a las representaciones anteriores para rotar y desplazar un cuerpo, en la figura 2.9 se muestra la transformación de coordenadas entre dos sistemas de referencia.

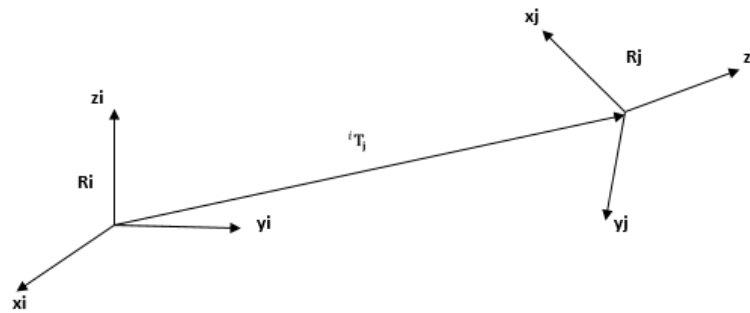


Figura 2.9: Transformación de coordenadas
Fuente: [31]

Esta transformación se define por la matriz T_j^i , la cual se expresa en la matriz 2.4:

$$T_j^i = \begin{bmatrix} s_j^i & n_j^i & a_j^i & P_j^i \end{bmatrix} \quad (2.4)$$

También se define así:

$$T_j^i = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Los vectores unitarios s_j^i, n_j^i, a_j^i son los vectores según los ejes x_j, y_j y z_j de la base R_j , expresados en la base R_i ; y que P_j^i es el vector que expresa el origen de la base R_j en la base R_i . Se tiene A_j^i , a la cual se le conoce como matriz de orientación o rotación [31], igual a 2.6:

$$A_j^i = \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix} \quad (2.6)$$

Y a P_j^i como vector de posición o translación, igual a 2.7:

$$P_j^i = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (2.7)$$

A continuación se muestra el significado de cada una de las sub-matrices en 2.8.:

$$T_j^i = \begin{bmatrix} Rotacion & Translacion \\ Perspectiva & Escalado \end{bmatrix} \quad (2.8)$$

Capítulo 3

Metodología de trabajo

En este capítulo se describe la metodología utilizada para el desarrollo de este trabajo, detallando los pasos y procedimientos que se siguieron en la propuesta de un sistema de localización de tejido epidérmico para la automatización de una sutura. Para ello, se trabajó sobre un modelo de desarrollo en cascada, puesto que se considera adecuado para abordar el problema desde el establecimiento claro de requisitos.

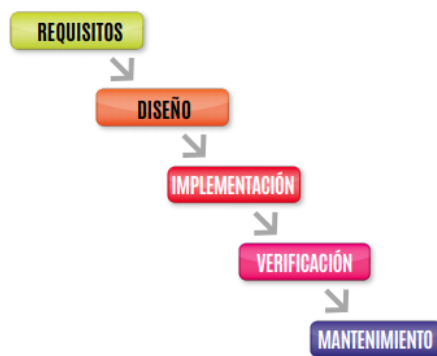


Figura 3.1: Desarrollo en cascada
Fuente: [32]

El modelo en cascada se divide en 5 fases: establecimiento de requisitos, diseño, implementación, verificación y mantenimiento [32]. Las cuales se realizan de manera ordenada, una tras otra. Razón por la cual, su complejidad radica en el buen establecimiento de los requisitos funcionales y no funcionales del proyecto. Siguiendo con lo anterior, como se observa en la figura 3.1, la fase inmediatamente posterior al establecimiento de requisitos es la de diseño, en la que establece la arquitectura completa del sistema y se describen a grandes rasgos las partes que conformaran el producto o servicio final.

De igual manera, se tienen las fases de implementación y verificación, en la primera se diseña el o los algoritmos y se ejecuta el software como conjunto o unidad de programas, para posteriormente pasar a la fase de integración y verificación de funcionamiento, en la cual se integran todas las partes del software y se hacen pruebas como un sistema completo.

Por último, se tiene la fase de mantenimiento, en la cual se pone en funcionamiento el sistema. A partir de este punto, el desarrollo se centra en la corrección de errores no descubiertos en las etapas anteriores, en mejorar el sistema y adaptar soluciones si hay nuevos requerimientos.

3.1. Establecimiento de los requisitos del sistema

3.1.1. Requisitos funcionales

Con el objetivo de implementar un sistema de localización de tejido epidérmico que involucre elementos de una escena quirúrgica para la automatización de una sutura, se establecieron los siguientes requerimientos funcionales:

El sistema debe poder localizar un cuerpo de superficie que emule el tejido epidérmico utilizando información de profundidad.

- Se debe establecer un dispositivo de captura de profundidad adecuado, con relación a los requisitos no funcionales.
- Se debe establecer la posición espacial de los elementos de la escena quirúrgica fijos, cámara y foco de luz.
- Se debe encontrar un algoritmo en un modelo de visión por computadora establecido, que se adapte a la necesidad de localizar puntos sobre el contorno de un objeto.
- Se debe disponer de un algoritmo que pese a oclusiones no parciales pueda reconstruir el contorno de una herida.

Se debe desarrollar un sistema de rastreo de elementos de escena quirúrgica.

- El sistema debe ser capaz de distinguir el elemento quirúrgico del resto de elementos de la escena.
- El sistema debe poder establecer la posición de la punta del objeto quirúrgico en el espacio.

Se debe evaluar el sistema de localización y seguimiento del instrumental quirúrgico con un tejido emulado, considerando un marco de referencia espacial común tanto para la herida como para el instrumental.

- Se debe establecer un plano de referencia común entre la herida y el instrumental quirúrgico.
- El sistema debe poder ubicar la herida y el instrumental quirúrgico en un mismo plano de referencia.
- Se debe poder visualizar el tejido epidérmico emulado estableciendo una correlación entre el plano virtual y el plano real.

3.1.2. Requisitos no funcionales

A continuación se describen los requisitos no funcionales:

- Debido a que el desarrollo del sistema se encuentra inmerso en una investigación más grande, el software debe tener la capacidad para ser modificado, efectiva y eficientemente dado a necesidades evolutivas correctivas o perfectivas [33].
- Debido a los diferentes periféricos que se pueden involucrar en la evolución del actual sistema de visión artificial, el software debe tener la capacidad de enviar y recibir información de manera uniforme y precisa.

- Debido a que el sistema tiene un enfoque de contacto físico directo con el ser humano, el sistema debe buscar ser preciso
- Debido a que el sistema actúa en tiempo real, se deben minimizar latencias, y buscar altas velocidades de respuesta.

3.2. Diseño del sistema

3.2.1. Evaluación y selección de los componentes físicos del sistema de visión

Considerando las características propias del escenario de trabajo, un quirófano, y con el objetivo de localizar una herida sobre un cuerpo de superficie suave que emule el tejido epidérmico, se evaluó como primera medida los componentes físicos de un sistema de visión, aptos para el desarrollo del proyecto, como lo son: El dispositivo de captura y su posición espacial, al igual que la ubicación, tipo de luz, intensidad y sentido de iluminación del foco,

3.2.1.1. Selección del dispositivo de captura

Para la selección del dispositivo de captura, se realizó un análisis, a partir de la investigación de las diferentes tecnologías y dispositivos de localización espacial utilizados o desarrollados en trabajos relacionados (Ver anexo C), consiguiendo hacer una caracterización de las ventajas y desventajas de cada tecnología según su respectiva técnica de abstracción:

De igual forma, en la tabla 3.1 se observan las características de funcionamiento que se tuvieron en cuenta para cada tecnología, las cuales son: Distancia mínima de captura, ángulo de visión, exactitud de la técnica de abstracción (fps, resolución), compatibilidad del sdk, precio y tamaño. A continuación se describe la importancia de cada una de estas características para el proyecto.

- Distancia mínima de profundidad: Teniendo en cuenta la relación inversa entre detalle y distancia y la necesidad de exactitud en la observación de los elementos y la región de trabajo, se pensó escoger una cámara que capturará la mayor cantidad de detalles sobre una región delimitada. En este sentido, la mínima distancia para capturar un buen mapa de profundidad.
- Ángulo de campo de visión: La tecnología debe contar con un buen ángulo de campo de visión relacionado con la distancia de profundidad mínima capturada, dado que la propuesta del sistema relaciona la localización de la región a suturar con la posición y seguimiento de los elementos quirúrgicos como pinzas y tijeras, sin cambiar la posición del dispositivo de captura.
- Compatibilidad del SDK de la tecnología: Dentro de la investigación se planteó escoger un dispositivo que pudiera soportar diferentes sistemas operativos como lenguajes de programación, además de que hubiera un soporte continuo del mismo en la actualidad.

DISPOSITIVO	KINECT 1	KINECT 2	CAMARA REALSENSE 415	CAMARA REALSENSE 435	CAMARA REALSENSE 435I	CAMARA ZED	CAMARA ZED 2
CARACTERÍSTICAS							
TECNICA	Luz estructurada	Tiempo de vuelo	Luz estructurada codificada por infrarrojos combinada con RGB estéreo(Estereoscopia activa)	Luz estructurada codificada por infrarrojos combinada con RGB estéreo(Estereoscopia activa)	Luz estructurada codificada por infrarrojos combinada con RGB estéreo(Estereoscopia activa)	Estereoscopia pasiva	Estereoscopia pasiva
RESOLUCION PROFUNDIDAD 3D	320x240 30 fps	512x424 30 fps	1280x720 90 fps	1280x720 90 fps	1280x720 90 fps	672x376 100 fps	672x376 100 fps
ANGULOS DEL CAMPO DE VISION DE PROFUNDIDAD	57 H; 43 V	70 H; 60 V	65 H; 40 V	86 H; 57 V	86 H; 57 V	90 H; 60 V	110 H; 70 V
DISTANCIA MINIMA DE USO	1,82 metros	1,37 metros	0,16 metros	0,105 metros	0,105 metros	0,3 metros	0,3 metros
RESOLUCION CAMARA RGB	640x480 30 fps 4:3	1920x1080 30 fps 16:9	1920x1080 30 fps	1920x1080 30 fps	1920x1080 30 fps	1920x1080 30 fps	1920x1080 30 fps
TAMAÑO	Grande	Grande	Pequeño	Pequeño	Pequeño	Mediano	Mediano
COSTO	60.00 \$	70.00 \$	159.00 \$	189.00 \$	209.00 \$	349.00 \$	449.00 \$
SDK ACTUALIZADO	No	No	Si	Si	Si	Si	Si

Tabla 3.1: Comparación de dispositivos
Fuente: Elaboración propia

- Tamaño: Si bien el tamaño podría ser algo indiferente, en este caso toma mucha relevancia porque la intención es lograr que los elementos que se encuentran en una sala de operaciones no interfieran con el trabajo de las personas.
- Resolución cámara RGB: Tener en cuenta que dependiendo de una resolución alta es posible obtener más detalles en la imagen y con ello lograr la calidad visual que se espera sea la adecuada en la ejecución del proyecto. (Dado que el sistema está pensado en tiempo real, es importante tener un buen nivel de fps, así como de resolución.)

De esta manera, se optó por seleccionar la cámara Realsense D435 como dispositivo de captura, presentando mejor adaptación al proyecto que los otros dispositivos analizados. Esto, debido en gran parte a la robustez que trae la técnica de abstracción estereoscópica activa sobre condiciones de iluminación alta, recordando que los niveles de iluminación en un quirófano superan los 1000 luxes; así como también sus características físicas y de operación: tamaño, distancia mínima de captura, compatibilidad y resolución de profundidad.

3.2.1.2. Posición espacial del dispositivo de captura

3.2.1.2.1. Ángulo de inclinación de la cámara con respecto al plano: Con el objetivo de determinar en que inclinación de la cámara se adapta mejor al proyecto, se varió la ubicación de la cámara con relación al ángulo que se forma entre la proyección de la captura y el plano donde se encuentra el objeto observado. Para observar fácilmente los cambios en la recepción de la imagen, se utilizó Realsense-viewer ¹. De esta manera la figura 3.2. presenta cuatro imágenes, con la posición de la cámara en un ángulo fijo de 45 grados respecto al plano donde se encuentra el phantom

¹Realsense-viewer: Interfaz de usuario creada por Intel para dinamizar de manera gráfica el comportamiento y las configuraciones que se pueden realizar a los dispositivos de esta marca

de sutura quirúrgico. Obteniendo: 1. Imagen de Color, 2. imagen de profundidad, 3. proyección de luz infrarroja del lente izquierdo y 4. proyección de luz infrarroja del lente derecho.

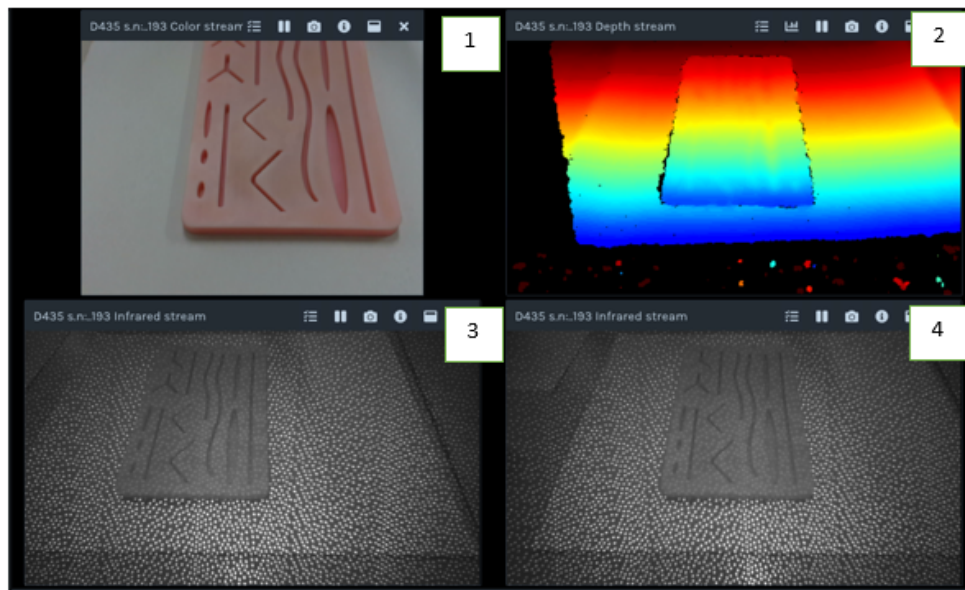


Figura 3.2: Imagen de realsense-viewer 45 grados: 1: Imagen de color; 2: Imagen de profundidad; 3: Infrarrojo izquierdo; 4: Infrarrojo derecho

Fuente: Elaboración propia

De igual forma, se realizó la toma de imágenes con la cámara en posición paralela al plano donde se encuentra el phantom, las imágenes en recepción se muestran en la figura 3.3.

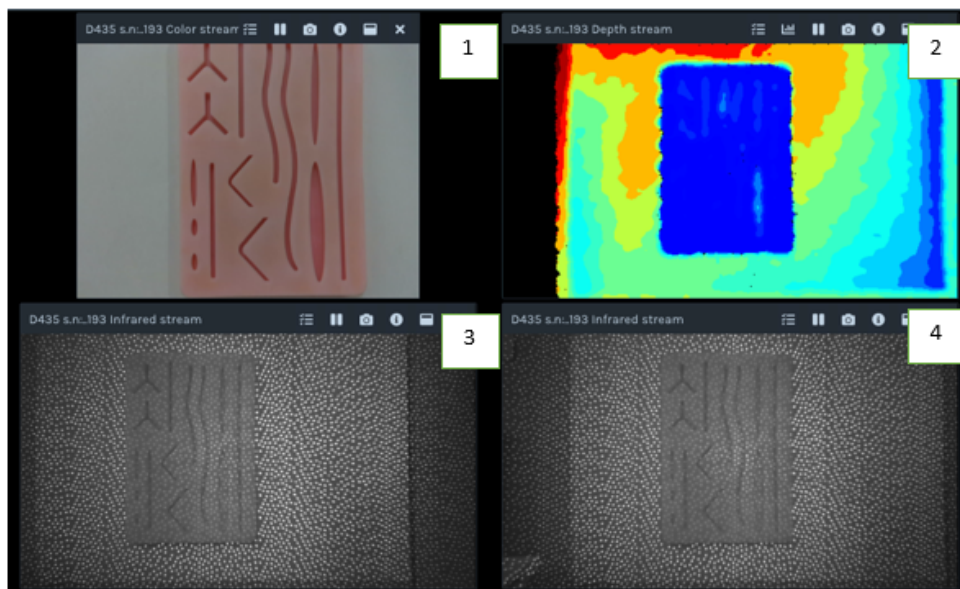


Figura 3.3: Imagen de realsense-viewer 90 grados: 1: Imagen de color; 2: Imagen de profundidad; 3: Infrarrojo izquierdo; 4: Infrarrojo derecho
Fuente: Elaboración propia

De esta manera, se puede observar, al comparar la imagen número 2 de las figuras 3.2 y 3.3, que la toma a 90 grados, presenta mayor precisión en la imagen de profundidad. Resultado que se relaciona implícitamente con las imágenes que se muestran en la proyección del infrarrojo (3 y 4). Esto, debido a que la formación de la imagen de profundidad resultante de la correlación de la cámara izquierda y derecha es reforzada por la proyección de puntos de luz infrarroja. Por esta razón, en el caso de ubicar la cámara a 45 grados, la proyección de los puntos descritos no se despliega de manera correcta, no marcando uniformemente los detalles que se quieren capturar del phantom, con lo cual se pierde precisión en profundidad luego del procesamiento de la imagen. Por otro lado, al ubicar la cámara de forma paralela al plano observado, se consigue el mejor despliegue de los puntos de luz infrarroja proyectados, como se puede observar en 3 y 4 de la figura 3.3. Por todo lo anterior y teniendo en cuenta la precisión que necesita un modelo de visión por computadora enfocado al contacto humano, se fijó la posición de la cámara en un ángulo de 90 grados con respecto al plano de referencia.

3.2.1.2.2. Distancia de la cámara con respecto al plano: Siguiendo con la descripción de la posición espacial de la cámara, se evaluó el dispositivo ubicado paralelo al plano de observación, en diferentes condiciones de luminosidad y distancias utilizando realsense-viewer. Lo cual permitió establecer rangos de distancia para la captura y posición espacial del dispositivo. Como también posibilitó determinar los niveles de intensidad lumínica propicios para ejecutar el método de localización. Para ello se varió la intensidad de luz, tomada en Luxes, y la posición del foco de proyección; así como también la distancia y la ubicación de la cámara en el espacio, obteniendo la tabla 3.2.

Luxes(lux) Distancia(cm)	500		1000		1500	
	Color	Depth	Color	Depth	Color	Depth
16.5						
17.1						
23.6						
28.6						
33						

Tabla 3.2: Rango de la cámara
Fuente: Elaboración propia

Con respecto a la tabla 3.2, se puede observar gran cambio en las imágenes de profundidad al variar la distancia. Para la primera distancia, 16.5 cm, el dispositivo de cámaras estéreo no logra correlacionar la imagen izquierda con la derecha, dando como resultado una imagen casi nula de profundidad. Caso contrario, para la segunda distancia, 17.1 cm; en la que el dispositivo puede correlacionar los datos que recibe de las dos cámaras, logrando la mejor percepción de profundidad. Para las siguientes distancias, 23.6, 28.6 y 33, la calidad de la imagen de profundidad disminuye, ya sea por ensanchamiento o por pérdida de nitidez.

Por otro lado, como era de esperarse, los detalles de la imagen a color disminuyen también en proporción a la distancia, se puede notar mayor percepción de iluminación en los cambios de lux a medida que esta aumenta, lo anterior debido a la variación del factor de exposición² digital automático que ejecuta la cámara. Así, para una condición de iluminación baja, el factor de exposición aumenta y para una iluminación alta el factor de exposición disminuye, mejorando la imagen. Por esta razón, aunque parezca mejor iluminada la imagen lejana a 1500 Lux, también tiene mayor ruido.

Debido a lo anterior y teniendo en cuenta la exactitud en la recepción de la imagen RGB y de profundidad, para el sistema de visión por computadora presentado en este proyecto se descartó ubicar a la cámara en posiciones que superen los 29 cm del plano observado. Por otro lado, conociendo que se puede modificar el parámetro de disparidad-shift de la cámara realsense para conseguir capturar objetos a una distancia más cercana, se delimitó el rango de distancia inferior de la cámara en 15cm, dejando 3-4 centímetros para el movimiento en z del instrumental. Con respecto a la intensidad de

²Factor que aumenta el nivel de intensidad de los píxeles en condiciones de baja iluminación añadiendo de igual forma ruido a la imagen y disminuyendo su calidad

luz, pensando en un ambiente quirúrgico³ se decidió trabajar en un rango de 500-2000 Lux.

3.2.2. Estructura física del experimento

3.2.2.1. Cámara de profundidad D435

Se considera uno de los elementos más importantes en el desarrollo del sistema, tanto sus características, como su utilidad se han descrito ya en este capítulo. Ver figura 3.4.



Figura 3.4: Cámara de profundidad D435
Fuente: Elaboración propia

Con la elección del dispositivo resuelta, y la implementación en conjunto de los elementos restantes como instrumental quirúrgico, iluminación y el phantom de la herida en el contexto de un entorno quirúrgico, fue necesario construir un prototipo de estructura para estos elementos. A continuación, se mencionarán los implementos utilizados a nivel hardware.

3.2.2.2. Pinza Laparoscópica

Se define como un instrumento diseñado para manipular, agarrar de una forma segura, el tejido epidérmico durante procedimientos laparoscópicos que sean posibles de realizar con dicho instrumento. Dentro de las principales funciones que realiza, se encuentra facilitar el acceso adecuado a una zona quirúrgica mediante la sujeción del tejido, para lograr obtener un campo visual claro del entorno. Con respecto a sus características físicas, tiene una longitud de 30 a 35 cm, pero es posible encontrarlos con longitudes más largas o cortas para diferentes escenarios. Estos instrumentos contienen una punta de agarre, la cual ha sido diseñada para evitar el menor trauma posible en cada movimiento. La punta de agarre puede tener un diseño diferente, ya sea lisa o una forma dentada como dientes de cocodrilo [34]. En la figura 3.6, se encuentra la pinza utilizada.

3.2.2.3. Luxómetro

Es un dispositivo empleado para medir la intensidad de la luz, con el que se puede obtener un estimado de los niveles de la cantidad de lúmenes por área en un ambiente controlado, siendo su unidad de medida el Lux.

³Ambiente quirúrgico: Para un ambiente quirúrgico la intensidad de luz varía según el procedimiento desde los 500 hasta los 10000 Lux



Figura 3.5: Pinza laparoscópica
Fuente: Elaboración propia



Figura 3.6: Luxómetro
Fuente: Elaboración propia

3.2.2.4. Almohadilla de sutura

Es una almohadilla hecha en silicona, diseñada específicamente para practicar suturas. La almohadilla es muy realista en la imitación de la piel humana, grasa y músculos. Como características posee una base antideslizante para evitar el movimiento cuando se esté realizando una sutura, además en la parte superior es algo pegajosa con el objetivo de que la piel se pegue más fácilmente al coser. En la figura 3.7, es mostrada la almohadilla de sutura.



Figura 3.7: Almohadilla de sutura
Fuente: Elaboración propia

3.2.2.5. Soporte móvil

El soporte móvil fue un diseño propio, inspirado en algunos ejemplos que se encontraron en la investigación. Para reducir costos, se optó por realizar un soporte en madera, ya que es un material económico, además de ser resistente. Como el soporte debía tener

movilidad y no ser estático, su diseño fue basado en articulaciones, lo cual dio un enfoque dinámico al posicionarlo a una distancia deseada, útil para hacer pruebas. Así mismo, es posible cambiarlo de ubicación si se desea, pues la sujeción de la base es una prensa, que permite instalarlo sobre cualquier superficie que contenga bordes. En la figura 3.8, se encuentra el soporte sujetado a una superficie, con la cámara instalada en su extremo.



Figura 3.8: Soporte móvil
Fuente: Elaboración propia

3.2.2.6. Lámpara

Con el objetivo de estandarizar las pruebas y que no afecte en gran medida la iluminación ambiente, para el experimento, se hizo uso de una fuente de iluminación enfocada. Por ese motivo se utiliza una lámpara que abarca de una manera adecuada todo el espacio de trabajo. En la figura 3.9, es mostrada la fuente de iluminación.



Figura 3.9: Lámpara
Fuente: Elaboración propia

A continuación, se muestra en la figura 3.10, el diseño esquemático de los componentes físicos del sistema.

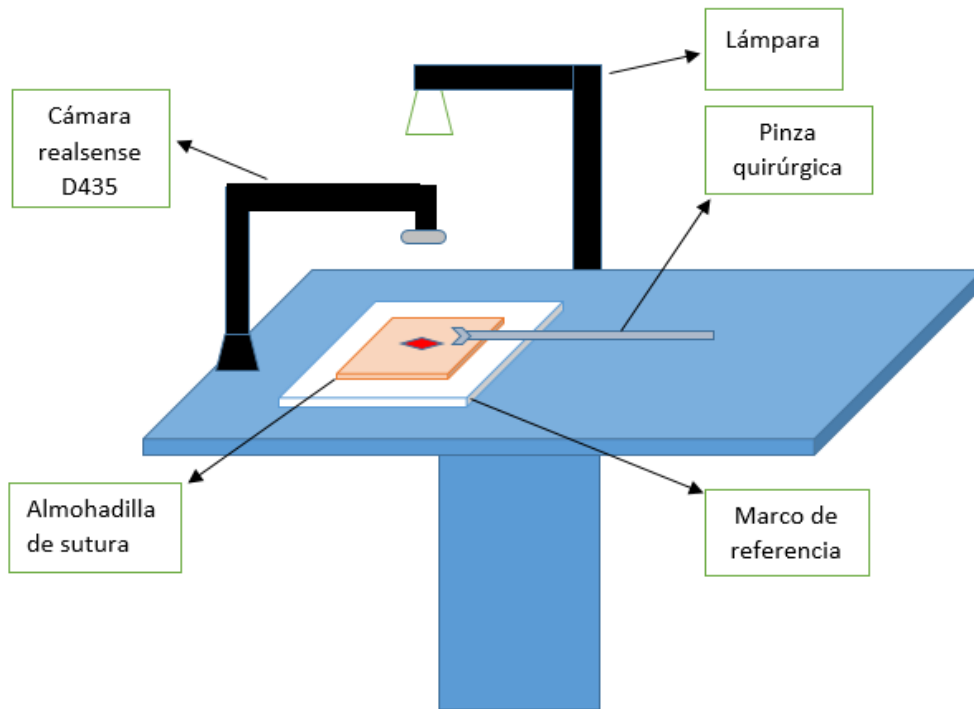


Figura 3.10: Diagrama general
Fuente: Elaboración propia

3.2.3. Análisis y selección del entorno y herramientas de programación para el sistema

Se realizó una investigación en búsqueda del entorno de programación más adecuado para el desarrollo del sistema, teniendo en cuenta factores como la conectividad, escalabilidad y la versatilidad del ambiente con el kit de desarrollo del dispositivo seleccionado.

Lenguajes de desarrollo lógico:

Para el desarrollo de programa principal se eligió como lenguaje de programación Python, que es un lenguaje de propósito general, el cual es popular principalmente debido a su simplicidad y legibilidad del código [35]. Entre sus características se encuentran las siguientes:

- Compatibilidad con todos los sistemas operativos (Linux, Windows, UNIX, macOS), lo que posibilita escribir y ejecutar algoritmos en cualquier distribución.
- Debido a que Python es gratuito y de código abierto, ha contribuido a generar una comunidad fuerte que está activa constantemente, por lo que es posible descargar código fuente, modificarlo y distribuirlo.
- Al ser un lenguaje interpretado y de alto nivel, la complejidad de su sintaxis es menor, comparada con otros lenguajes de programación como c++ o java.

De esta manera, al evidenciar las fortalezas de las herramientas de programación descritas anteriormente y utilizadas en distintas investigaciones afines a este proyecto [36] [20] [22], se decidió usar ROS, como entorno de interconexión; Python, como lenguaje principal para la lógica del desarrollo, Open CV, como librería para el procesamiento de imágenes; y Unity, para la interfaz de emulación del proyecto. Todo lo

anterior en adición a la compatibilidad que estas tienen con el kit de desarrollo de la cámara realsense.

3.2.3.1. Librería Open CV-Python



Figura 3.11: Logo de Open CV-Python
Fuente: [37]

Para el procesamiento de las imágenes, se decidió utilizar la librería Open CV sobre Python, la cual trae consigo diferentes algoritmos que facilitan el desarrollo de sistemas de visión por computadora. Esta librería combina las mejores cualidades de la API C++ de Open CV y el lenguaje Python. Hace uso de NumPy, que es una biblioteca altamente optimizada para operaciones numéricas con una sintaxis de estilo de MATLAB. Debido a esto, todas las estructuras matriciales de Open CV se pueden convertir en matrices NumPy, lo cual facilita la integración con otras bibliotecas que usan NumPy, como SciPy y Matplotlib, las cuales también fueron utilizadas en el proyecto. [35].

3.2.3.2. Robot Operating System (ROS)

El sistema operativo de robot, es un conjunto de bibliotecas de software y herramientas que ayudan a crear aplicaciones de robot. Desde controladores hasta algoritmos de última generación y potentes herramientas de desarrollo. Con respecto a las contribuciones de los paquetes en ROS, se dan bajo una gran variedad de licencias diferentes; además, los lenguajes de programación implementados son Python, C++ y lisp. Existen diferentes versiones que pueden llegar a ser incompatibles entre ellas [38].

3.2.3.2.1. RosBridge Es un componente que permite comunicar programas externos a ROS, como Unity, con nodos de ROS de forma directa. Ros Bridge establece un protocolo de comunicaciones que permite suscribirse o publicar en un tópico de ROS, desde los nodos creados en programas externos.

3.2.3.3. Unity 3D

Existen numerosas plataformas que permiten el desarrollo de simulaciones en tiempo real. Con el fin de simular el sistema planteado en un ambiente virtual, la herramienta escogida para el desarrollo de este trabajo de grado es Unity 3D, debido a que su curva de aprendizajes es baja, cuenta con una versión de licencia gratuita y ha sido ampliamente utilizado en el desarrollo de otras investigaciones orientadas hacia la misma temática. Unity 3D es un motor de videojuegos creado por Unity Technologies, donde es posible realizar el diseño, la creación y la representación de múltiples ambientes en 2D o 3D, para el desarrollo de simulaciones y juegos implementados en distintas plataformas como: Windows, Linux, OS, Android, Xbox. Cuenta con una comunidad activa que permite resolver dudas en poco tiempo y de igual manera es posible programar en diferentes lenguajes de programación. En la figura 3.12, se encuentra el logo que identifica a esta herramienta.

Otras herramientas utilizadas en el desarrollo fueron:



Figura 3.12: Logo de la herramienta Unity 3D
Fuente: [37]

- Notepad++: Es un editor de texto y de código fuente libre, el cual soporta varios lenguajes de programación. Incluye opciones avanzadas, útiles para usuarios avanzados como programadores. Se usó en la programación de los archivos en Linux para el programa principal.
- Visual studio code: Es donde se realiza la programación para los scripts de Unity a implementar.

3.2.4. Análisis y selección del modelo de visión por computadora para el sistema

Para la selección del modelo de visión por computadora utilizado en el sistema, se realizó una investigación relacionada con los modelos y algoritmos que se han implementado para clasificar, detectar, localizar y hacer seguimiento de puntos, planos y objetos. Con lo cual, se determinó que los modelos de segmentación y localización son los que mejor se adaptan con el desarrollo propuesto. De esta manera, bajo la lógica de determinar la posición de los puntos que conforman una herida en un cuerpo de superficie suave, se decidió utilizar un modelo de segmentación; mientras que para el instrumental quirúrgico, un modelo de localización.

Con relación a lo anterior, se evaluaron diferentes tipos de algoritmos de segmentación: por umbralización, basada en el método de otsu y en color; por crecimiento de regiones, watershed; y utilizando grafos, con grabcut; los cuales fueron probados en combinación de algoritmos morfológicos binarios como dilatación y erosión para evitar la oclusión. En la figura 3.13 se observa la segmentación de un cuerpo que simula una herida utilizando los algoritmos anteriormente descritos, evidenciando menor porcentaje de píxeles erróneos y, por lo tanto, mejor segmentación y robustez a la oclusión para el algoritmo de segmentación por color y cierre morfológico ⁴

De igual forma, se investigaron y analizaron diferentes alternativas para conseguir localizar el instrumental quirúrgico, bajo un modelo de detección. De esta manera, se obtuvieron dos alternativas: Por un lado, hacer la detección de objetos ubicando en ellos puntos característicos predefinidos, comúnmente llamados marcadores; por otro lado, se pensó hacer detección de objetos utilizando redes convolucionales a partir de una red neuronal entrenada.

Pese a que el uso de redes convolucionales para la clasificación de objetos por medio de una red neuronal entrenada, parecía ser la idea más llamativa, se concluyó no viable

⁴Es una transformación morfológica en la cual se hace dilatación y luego erosión. Resultando útil si se quiere cerrar pequeños agujeros dentro de los objetos de primer plano, producto de la superposición de elementos u objetos en un segundo plano.











Imagen original	Otsu con morfológico	Otsu sin morfológico	Color morfológico	Color sin morfológico
				
	5.079%	5.07%	3.848%	4.239%
Watershed sin morfológico	Watershed con morfológico	Grabcut	Grabcut con morfológico	Grabcut-Otsu
				
4.342%	3.850%	4.127%	3.983%	4.053%

Figura 3.13: Porcentajes de píxeles no correlacionados según la segmentación
Fuente: Elaboración propia

para las características de este proyecto, debido a la incompatibilidad con Python 2.7 (en el cual se ejecuta todo el middleware de ROS sobre Python) haciendo inviable el proyecto por factores como la interconectividad y mantenimiento de los cuales es garante ROS⁵. Debido a lo anterior, se decidió colocar objetos característicos por color predefinidos para localizar el instrumental quirúrgico.

3.2.5. Ubicación de la punta de la pinza, usando marcadores

Debido a la necesidad de ubicar la punta de la pinza, sin poner en ese lugar un marcador, que se vería fácilmente ocluido por cualquier procedimiento que se desarrolle en ese lugar; se diseñó y desarrolló un modelo geométrico que permite conocer la punta del instrumental a partir de la posición 'x', 'y' de los marcadores y las proyecciones de la sombra que forma el instrumental según la pendiente del segmento de recta entre los dos marcadores.

Como se había mencionado anteriormente, para la detección y el seguimiento del instrumental quirúrgico se ubicaron elementos característicos en el cuerpo del objeto. Los cuales fueron, marcadores de color, la figura 3.15 muestra una representación de la pinza, con los marcadores y el plano donde se encuentra la herida

De esta manera, se ubicaron dos marcadores de diferente color, un marcador de color azul con posición $A=(x_1, y_1)$ y un marcador de color verde con posición $B=(x_2, y_2)$, separados a una distancia de 2 cm entre ellos, ubicando el marcador más cercano a una distancia V_f de la punta del instrumental. Conociendo que la pendiente entre los dos puntos se expresa matemáticamente como (1) y que el punto ubicado en V_f se

⁵Pese a que existe una versión 2 de ROS: 'ROS2', que tiene Python 3.3 en su núcleo, en vez de Python 2.7, lo que posibilita trabajar con bibliotecas más recientes como TensorFlow; aún es muy inestable y por ello no se consideró su uso desde un principio

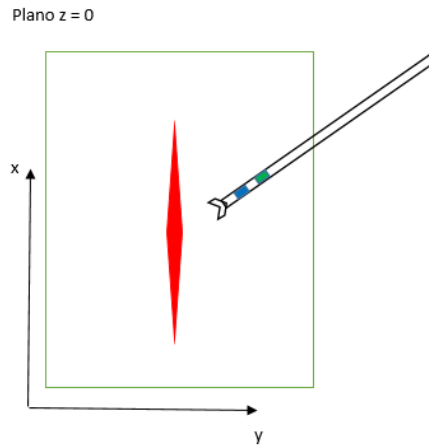


Figura 3.14: Representación geométrica de la pinza y la herida en un plano 2D
Fuente: Elaboración propia

encuentra en la recta $y = mx + b$ con AB como segmento de recta perteneciente a esta recta, se tiene que el punto V_{fx} en función del segmento de recta AB está dado por la ecuación de la recta (2)

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (1)$$

$$y = mx + mV_{fx} \quad (2)$$

Donde V_{fx} es el valor de V_f en x , añadido al instrumental. El cual se debe multiplicar por un factor escalar α , que varía en función de la pendiente del segmento de recta AB , obteniendo la ecuación (3)

$$y = mx + mV_{fx}\alpha \quad (3)$$

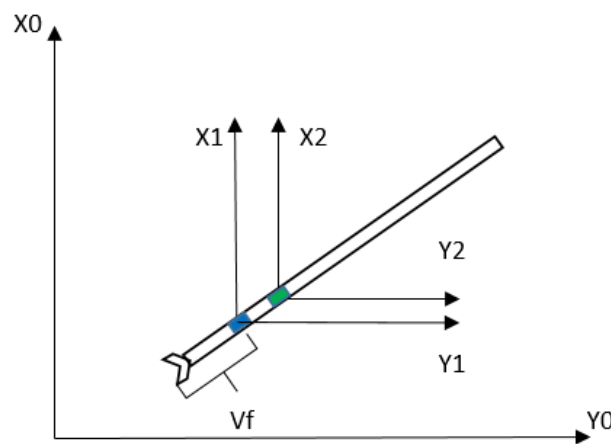


Figura 3.15: Ubicación de los marcadores en la pinza
Fuente: Elaboración propia

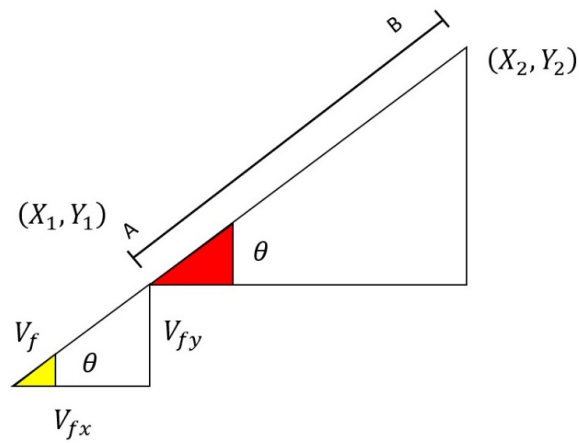


Figura 3.16: Proyección del valor V_f añadido sobre el segmento de recta AB
Fuente: Elaboración propia

Debido a que la pendiente del segmento de recta AB (ahora solo llamada pendiente), varía en función de la altura 'Z', se procede a modelar la ecuación en el espacio 3D. Para lo cual se tiene la figura 3.18. En la cuál para efectos de simplificación se trabajó con desplazamiento $y = 0$. De esta manera, se observa que para valores de pendiente de V_f elevados la sombra o proyección es pequeña y para valores de pendiente pequeños la proyección es grande. Entonces:

$$Z \uparrow \rightarrow V_f \downarrow \rightarrow 0(3)$$

$$Z \downarrow \rightarrow V_{fp} \uparrow \rightarrow V_{fx}(4)$$

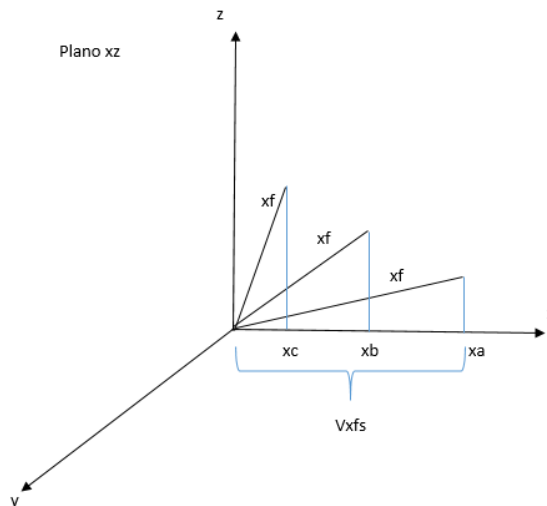


Figura 3.17
Fuente: Elaboración propia

Teniendo más claro que la variación de la proyección de V_f depende de la inclinación de la recta en la que está comprendido el punto V_f . Se procede a encontrar el valor escalar α que representa esta dependencia. Haciendo una relación de triángulos se encontró que el valor de α se puede expresar como la división entre la sombra de V_f en x, es decir, V_{fx} proyectada en el plano 'x', 'y', y la distancia V_f que para el caso de tener $y=0$, es la distancia del punto x, z en el plano x, z. Por lo tanto:

$$\alpha = \frac{x_2 - x_1}{Dist(x, z)} \quad (5)$$

Si se observa (5), se tiene que para valores de *alpha* igual a 1, la ecuación (3) es igual a la ecuación (2), lo cual es coherente con lo expresado en (4). Continuando con el análisis, se resuelve el problema anterior, para un vector con componente en *y* diferente de 0 como se observa en la figura 3.18. Con ello la ecuación que representa la variación de α se presenta en (5).

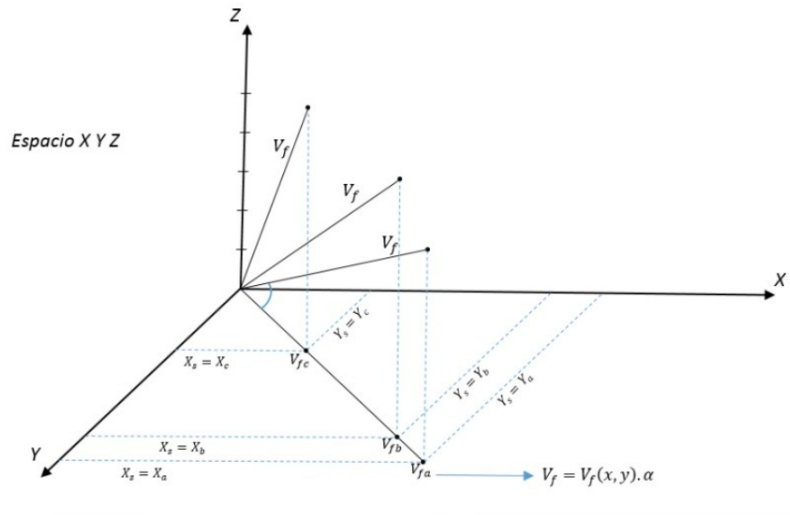


Figura 3.18
Fuente: Elaboración propia

$$\alpha = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{Dist(x, y, z)} \quad (6)$$

$$V_f = V_{f_{x,y}} \alpha \quad (7)$$

Ahora bien, hasta este punto se ha encontrado el valor escalar multiplicativo, que podría ser también llamado valor de corrección de distancia; pero no se ha encontrado su sentido de orientación en el plano x, y de manera concreta. Para ello, se encuentra la proyección V_f que se tiene en su componente 'x' e 'y'.

Entonces, las componentes en 'x' y en 'y' de V_f son respectivamente(8) y (9):

$$V_{fx} = V_f \cos \theta$$

$$V_{fy} = V_f \sin \theta$$

con:

$$\theta = \text{arccotan}(m) = \text{arccotan}\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$$

Con la elección de los elementos para la estructura física, así como de las herramientas software para el sistema, es necesario definir el ambiente virtual donde se realizara la simulación, para ello, a continuación se describe el proceso de creación del mismo.

3.2.6. Correlación del instrumental y los puntos de contorno de la herida en un mismo plano de referencia

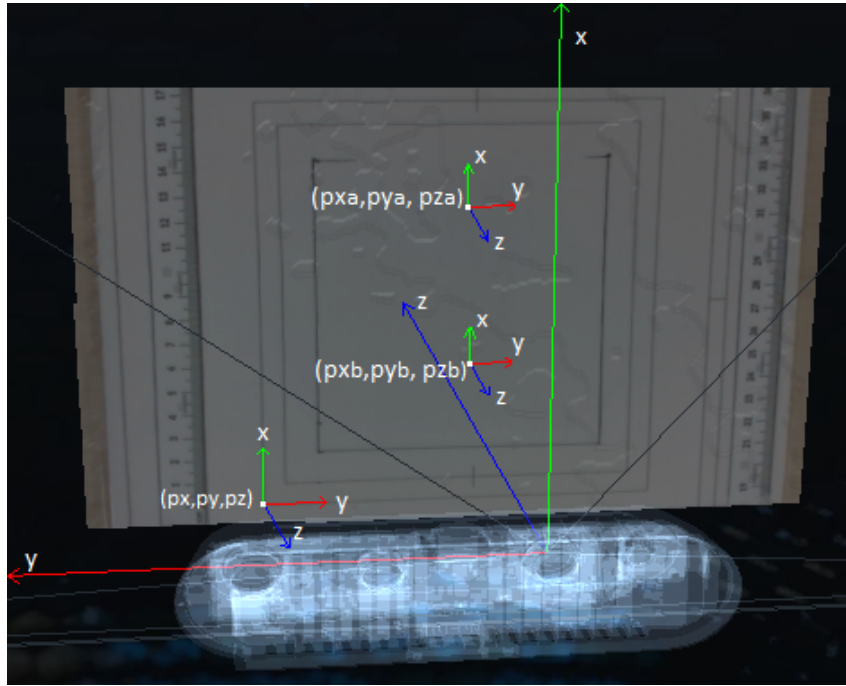


Figura 3.19: Esquemático de los planos de referencia
Fuente: Elaboración propia

La matriz homogénea correspondiente al caso estudiado se observa en 3.1. Donde P_x , P_y y P_z son las coordenadas del punto de referencia del sistema coordenado 'j' tomado a partir del sistema coordenado de la cámara 'i' (el vector de translación). Al multiplicar esta matriz por el vector de posición (p_x, p_y, p_z) , se obtiene la posición de p_x, p_y, p_z con respecto al punto de referencia mencionado. Para este caso se observa en 3.1 que tanto el eje z como el eje y rotaron 180 grados.

$$T_j^i = \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & -1 & 0 & P_y \\ 0 & 0 & -1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

3.2.7. Arquitectura del sistema

A continuación se presenta la arquitectura, hardware y software del sistema.

3.2.7.1. Arquitectura hardware del sistema

La parte hardware del sistema de visión por computadora propuesto, está conformada por la cámara Realsense d435, como el dispositivo de captura; por un ordenador lenovo, con procesador Corei5, de 8Gb de Ram, en el cual se hace el procesamiento de imagen de la información obtenida de la cámara R-D435; por un computador Asus, con procesador Corei5, 6Gb de Ram y tarjeta gráfica invidia930MX, en el cual se lleva a cabo la emulación de las imágenes procesadas; y por un router, al que se conectan los dos equipos bajo una red de área local, de forma inalámbrica, para el intercambio de mensajes. De esta manera, como se observa en la figura 3.20, el ordenador 'A' recoge la información de la cámara, la procesa y la envía al ordenador 'B', el cual se encuentra en la misma red de conexión local.

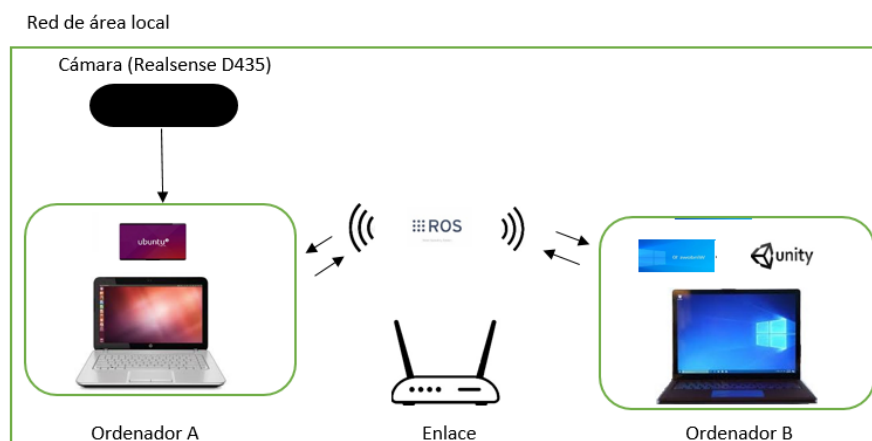


Figura 3.20: Arquitectura hardware

Fuente:Elaboración propia

3.2.7.2. Arquitectura Software del Sistema

La arquitectura general del software se divide en 5 módulos: Interfaz de control, adquisición de la imagen, procesamiento de la imagen, calibración depth-color, conexión ros bridge y visualización. Ver figura 3.21.

- Interfaz de control: La interfaz de control o interfaz gráfica de usuario (GUI por sus siglas en inglés) permite la interacción del usuario con el programa, por medio de esta GUI es posible seleccionar la región de interés a observar (región ROI), así como también el plano de referencia común a la región observada y al instrumental.
- Adquisición y procesamiento de imagen: El módulo de procesamiento de imágenes contiene los algoritmos necesarios para hacer detección a partir de los datos captados por el módulo de adquisición de imagen.
- Calibración depth-color: En este módulo se consigue localizar un punto en el espacio, por medio de una calibración entre la imagen de color y la profundidad.
- Conexión Ros bridge y visualización: En el módulo de conexión, se establece el ejecutable del programa como un nodo en ROS, para el envío y recepción de mensajes por un medio de un Topic en ROS. Datos que serán emulados en Unity.

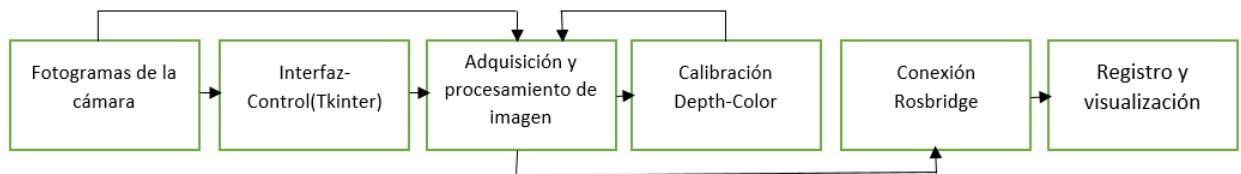


Figura 3.21: Arquitectura software
Fuente:Elaboración propia

3.3. Implementación del sistema

A continuación se presentan los módulos que componen el sistema de manera interconectada. Como se puede ver en la figura 3.22 estos se han clasificado, según su proceso, en módulos de entrada, procesamiento y salida.

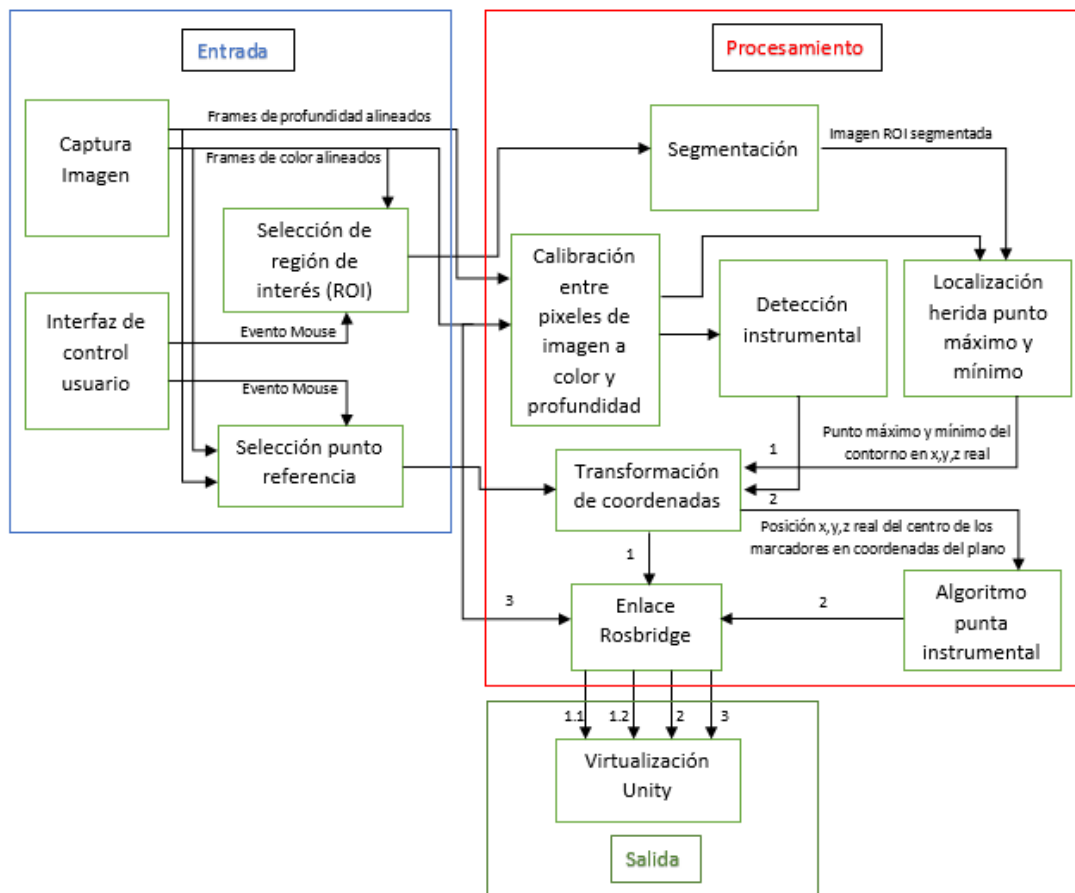


Figura 3.22: Módulos del sistema
Fuente: Elaboración propia

Sobre la entrada, se tienen tres módulos: Selección ROI, en el cual el usuario acota la región de interés que posteriormente se segmentará; selección del punto de referencia, en la cual el usuario ubica la posición espacial inicial que será relativa a la herida y al instrumental; y captura de imagen, que como ya se había mencionado, se utilizó la cámara Realsense D435, de la cual se obtiene información de imagen a color en formato RGB e información de profundidad por medio de un proceso interno de correlación entre las imágenes obtenidas con su sistema de cámaras estéreo. Entre los módulos de procesamiento se encuentran: El módulo de calibración, en el cual consigue relacionar cada píxel de la imagen a color con el valor de profundidad en ese punto, estimando una medida en unidades reales; el módulo de segmentación, en el que se realiza una umbralización por color sobre una región previamente acotada y luego se utiliza un algoritmo de cierre morfológico; el módulo de detección, en el que se obtienen las coordenadas de los marcadores previamente definidos en el instrumental quirúrgico; el módulo de función de transformación de coordenadas, en el que se realiza un proceso de translación y rotación de las coordenadas de la cámara iniciales, punto definido por el usuario; y el módulo de enlace Ros bridge, en el cual se hace del ejecutable principal un nodo de Ros, se crean publicadores y temas por los cuales se transmite la

información. Por último, en la salida se tiene el módulo de virtualización de la escena quirúrgica, en el cual se proyecta la imagen sobre la interfaz de Unity. A continuación se describirán cada uno de los bloques del sistema, teniendo en cuenta que cada uno de ellos se componen por una entrada, un proceso y una salida.

3.3.1. Esquema de la configuración cámara y captura de imagen de color y profundidad

Para poder capturar la imagen de color y profundidad se debe primero hacer un proceso de configuración, en el cual se establece el flujo de datos, sea de color o profundidad, el tamaño del frame, formato de codificación de imagen y la cantidad de frames por segundo, seguido de esto el sistema debe hacer una alineación entre la imagen a color y profundidad, como salida de este bloque se obtienen los frames de profundidad y de color alineados.

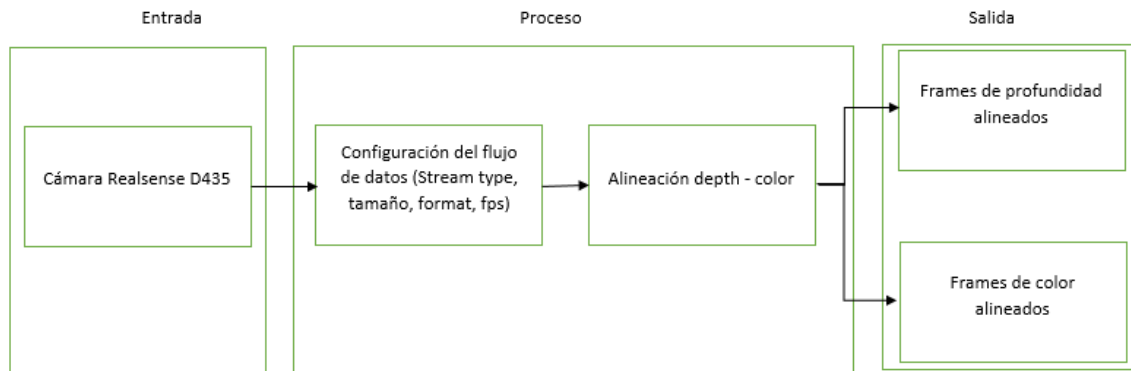


Figura 3.23: Esquema de configuración de la cámara
Fuente: Elaboración propia

3.3.2. Esquema de selección ROI

Para la selección ROI (Región de interés) se recibe como parámetros de entrada los frames de color alineados, de los cuales se toma la imagen a color alineada y un evento que ejecuta el usuario por Mouse, con el cual se crea un rectángulo. Como salida se tiene la región ROI.

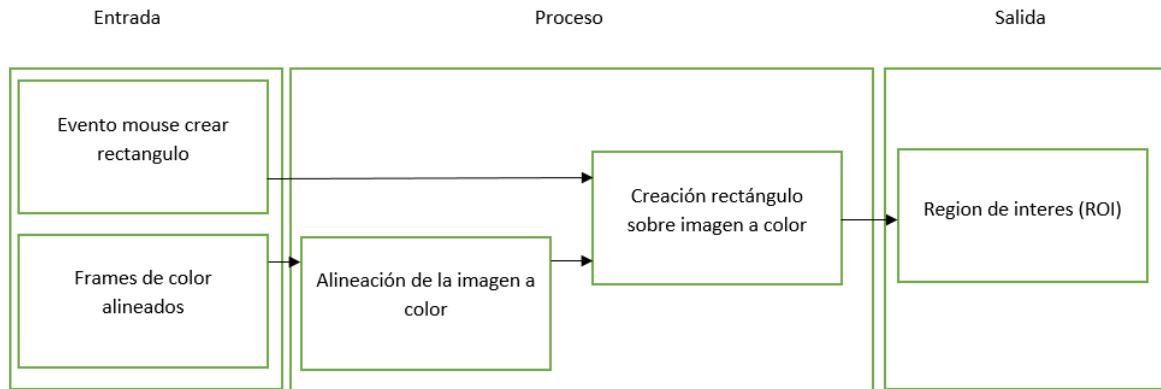


Figura 3.24: Esquema de selección ROI
Fuente: Elaboración propia

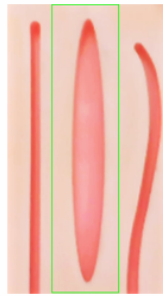


Figura 3.25: Región de interes seleccionada
Fuente: Elaboración propia

3.3.3. Esquema de la selección del punto de referencia

Como se quiere disponer de la posición de la herida y del instrumental quirúrgico referidos en un mismo plano, se hace necesario hacer una calibración de las coordenadas con respecto a un punto de referencia. Para ello, como entradas en este bloque, se tiene un evento por mouse, sobre la imagen de color alineada, con lo cual el usuario selecciona las coordenadas del punto de interes. De igual forma, se tiene como entrada los frames de color y profundidad alineados, con los cuales se realizará una calibración entre los píxeles de la imagen a color y profundidad. De esta manera, el último proceso que se realiza es el de calcular las coordenadas x , y , z reales. Como salida se guardan las coordenadas de ese punto, obteniendo un punto estático con coordenadas x , y , z reales.

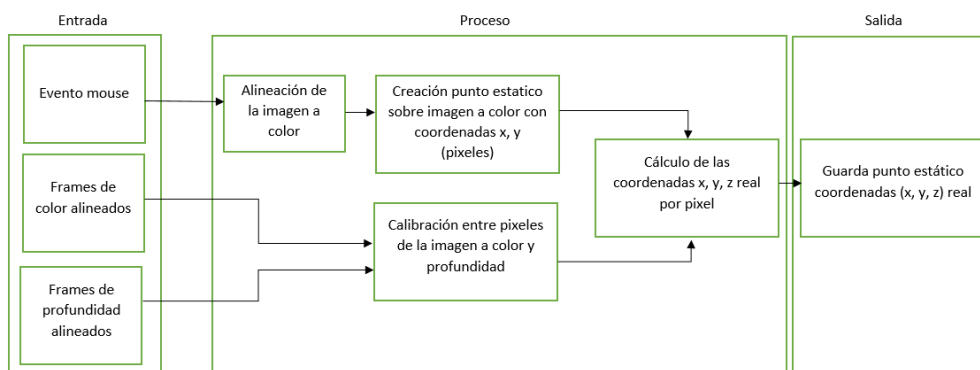


Figura 3.26: Esquema de selección del punto de referencia
Fuente: Elaboración propia

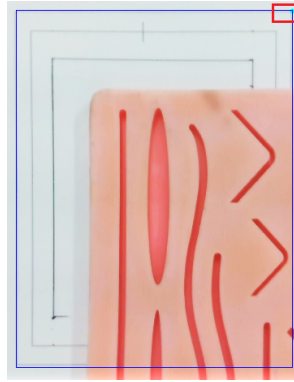


Figura 3.27: Punto de referencia seleccionado(cuadro rojo)
Fuente: Elaboración propia

3.3.4. Esquema de la segmentación

Para el módulo de segmentación se recibe la región de interes, entregada por el módulo selección ROI; y la imagen a color alineada, en el proceso de este bloque se realiza una umbralización por color acotada a la región ROI, luego se aplica un algoritmo de cierre morfológico y se obtiene la imagen acotada en la región de interes y segmentada.

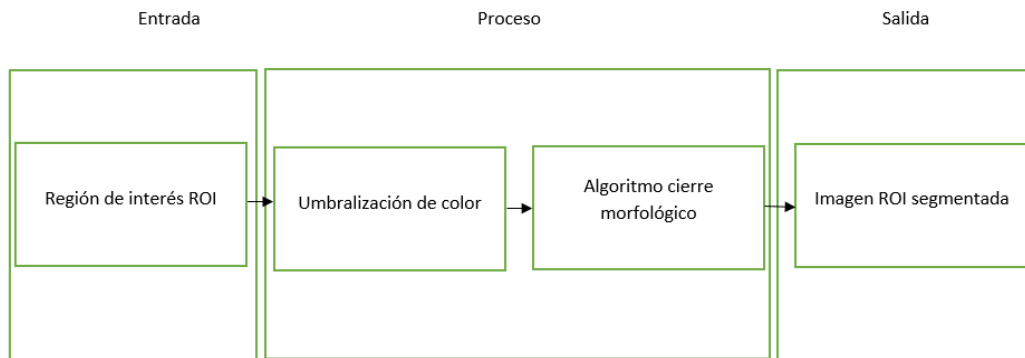


Figura 3.28: Esquema de la segmentación
Fuente: Elaboración propia

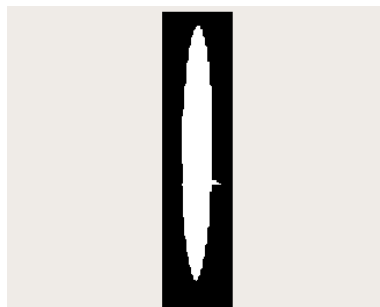


Figura 3.29: Imagen segmentada
Fuente: Elaboración propia

3.3.5. Esquema de la localización de la herida, punto máximo y mínimo

Para el módulo de localización de la herida, punto máximo y mínimo. Se recibe la imagen ROI segmentada del bloque anterior y la calibración entre píxeles de la imagen a color y profundidad. Con la imagen ROI segmentada se encuentran vectores de los N contornos, tomándose el más grande y seleccionando los puntos que tengan valor máximo y mínimo, con estos puntos y la calibración entre píxeles de la imagen a color y profundidad se calculan las coordenadas x, y, z reales por pixel, teniendo a la salida el punto máximo y mínimo de la herida en un espacio real x, y, z.

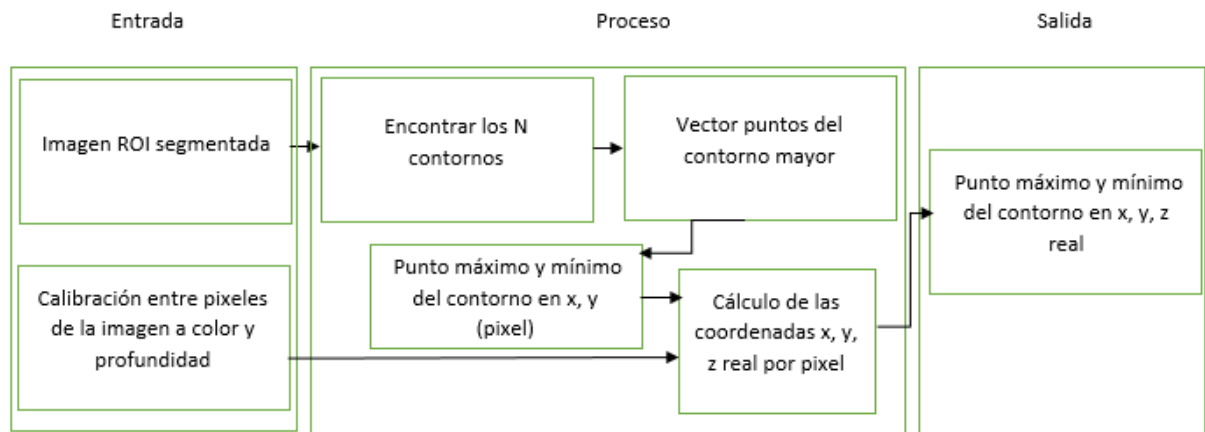


Figura 3.30: Esquema de la localización de la herida
Fuente: Elaboración propia



Figura 3.31: Puntos máximo y mínimo de la herida
Fuente: Elaboración propia

3.3.6. Esquema de la calibración entre píxeles de la imagen a color y profundidad

En este bloque ingresan los frames de color y profundidad alineados, se encuentran los parámetros intrínsecos de profundidad y color, luego se alinean estos parámetros entregando a la salida los parámetros intrínsecos de profundidad alineados al color y la imagen a color y profundidad alineada.

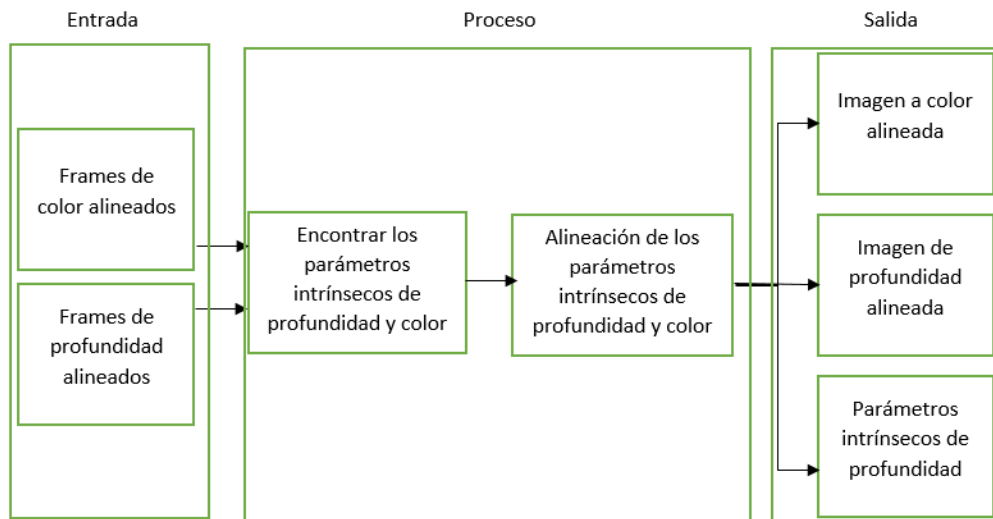


Figura 3.32: Esquema de la calibración entre píxeles de la imagen a color y profundidad

Fuente: Elaboración propia

3.3.7. Esquema del cálculo de las coordenadas x, y, z real por pixel

Para el cálculo de las coordenadas x, y, z reales por pixel, se debe procesar la imagen de color y profundidad alineada con los parámetros intrínsecos de profundidad encontrados en el bloque de calibración para los puntos x, y en píxeles. Como resultado se obtiene la posición x, y, z real del punto x, y en píxeles.

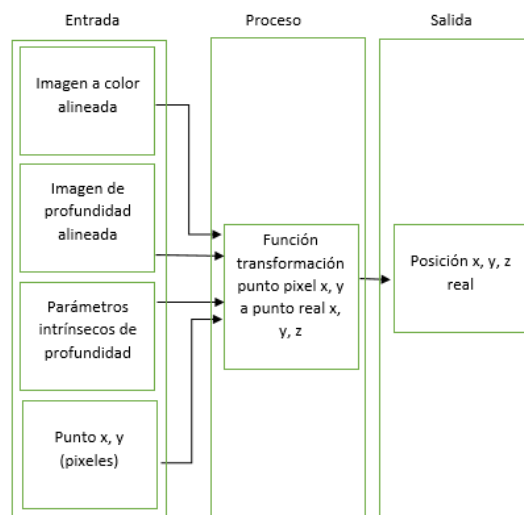


Figura 3.33: Esquema del cálculo de coordenadas x, y, z real por pixel

Fuente: Elaboración propia

3.3.8. Esquema de detección del instrumental quirúrgico

El bloque de detección de instrumental quirúrgico recibe como parámetros de entrada la calibración entre píxeles de la imagen a color y profundidad. En su procesamiento realiza una segmentación por color, encuentra los contornos y valida los puntos respectivos a seguir por área; luego, obtiene la posición x, y del centro de los contornos para

calcular las coordenadas x, y, z reales por pixel. A la salida se obtiene la posición x, y, z del centro de los contornos en el espacio real.

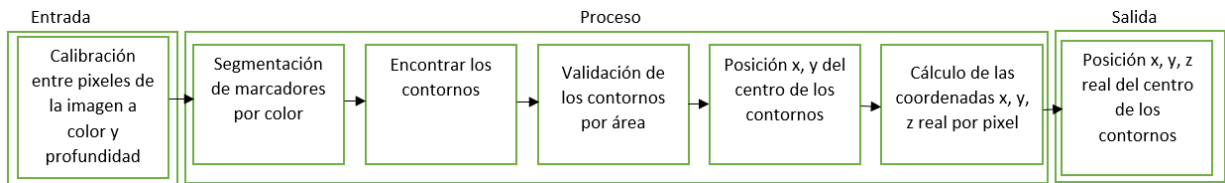


Figura 3.34: Esquema de la detección del instrumental
Fuente: Elaboración propia



Figura 3.35: Detección de la pinza quirúrgica
Fuente: Elaboración propia

3.3.9. Esquema de detección de la punta del instrumental

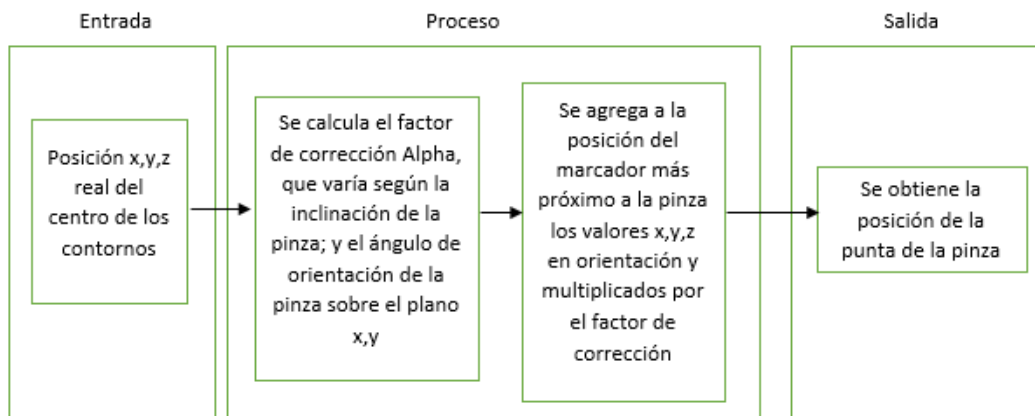


Figura 3.36: Esquema de la detección de la punta del instrumental
Fuente: Elaboración propia

3.3.10. Esquema de la función de transformación de coordenadas

Para la función de transformación de coordenadas se selecciona un punto de referencia con el cual se establecerá la posición de la pinza y de los puntos máximo y mínimo de

la pinza, haciendo una transformación matricial en translación y rotación. A la salida se obtiene la posición x, y, z con respecto al punto de referencia tanto de la herida como del instrumental.

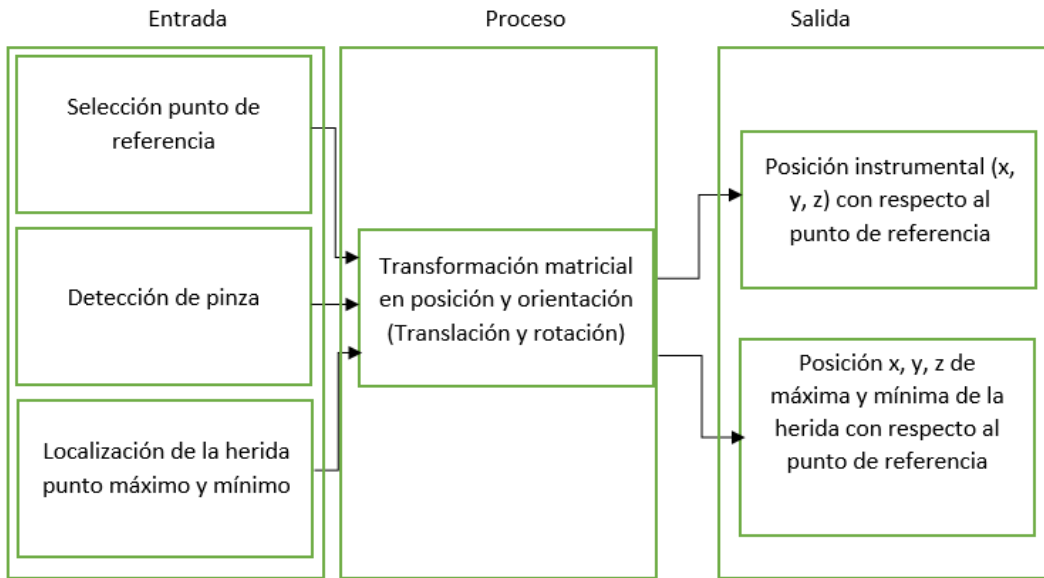


Figura 3.37: Esquema de la función de transformación de coordenadas
Fuente: Elaboración propia

3.3.11. Esquema del enlace RosBridge

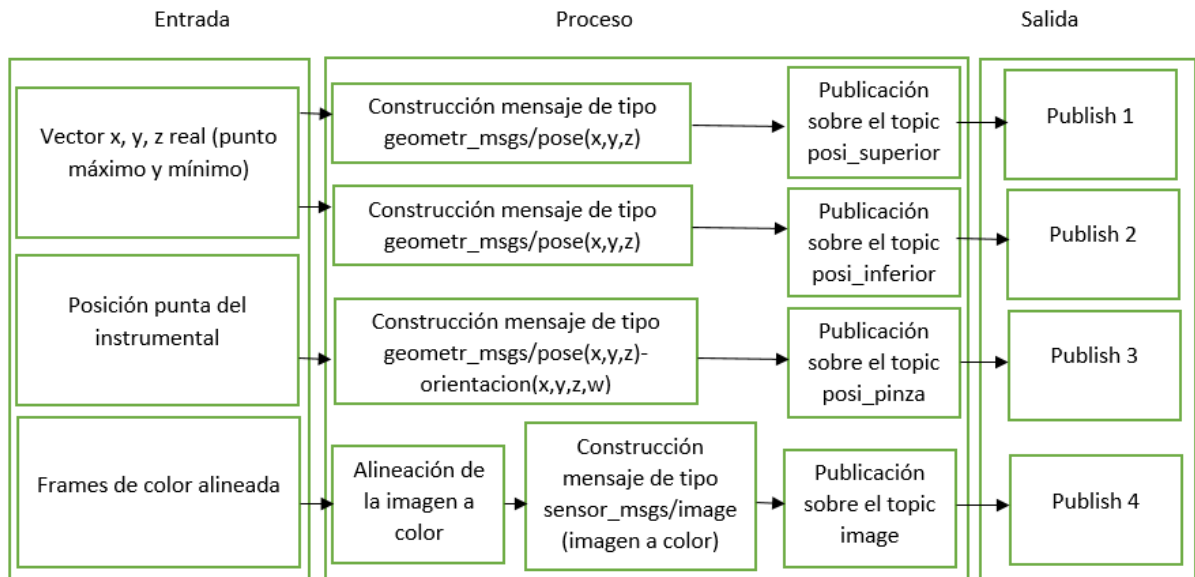


Figura 3.38: Esquema del enlace RosBridge
Fuente: Elaboración propia

3.3.12. Esquema de la virtualización de la escena quirúrgica

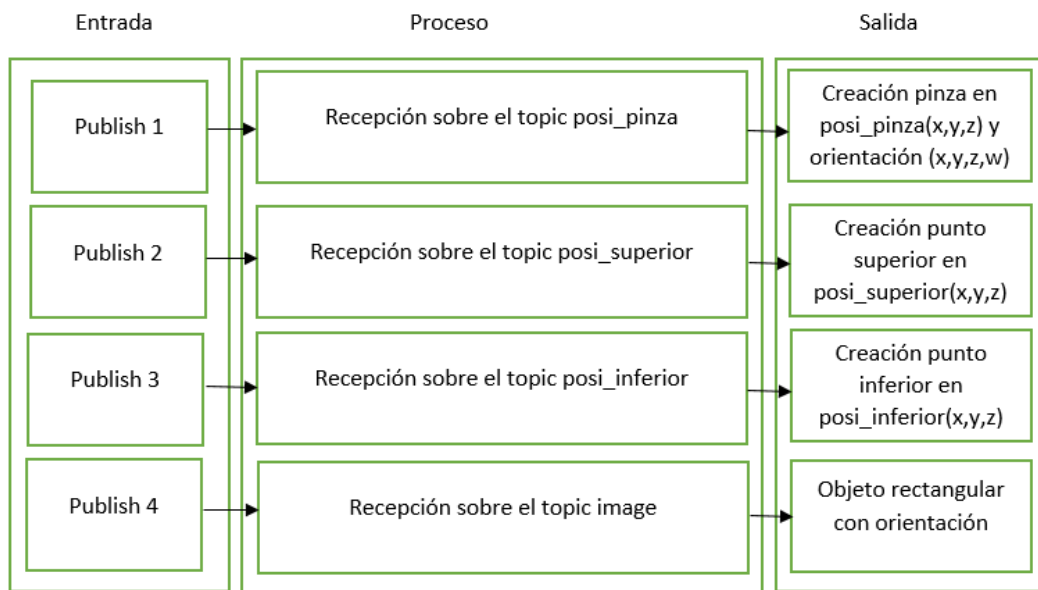


Figura 3.39: Esquema de virtualización
Fuente: Elaboración propia

Capítulo 4

Pruebas y Resultados

En este capítulo se presenta la evaluación realizada al sistema de localización de tejido epidérmico. Para ello, se realizaron diferentes pruebas orientadas a encontrar cuantitativamente que tan preciso en localización y seguimiento es el sistema, variando las condiciones espaciales y de luminosidad con respecto al valor verdadero medido manualmente. También, se realizó pruebas con respecto a su respuesta en correlación y precisión en un ambiente emulado.

En la primera prueba, se recopilaron muestras de 1000 datos, correspondientes a la posición x , y , z del punto máximo y mínimo de la herida, considerando diferentes variaciones. De esta manera, para cada muestra se varió la posición del dispositivo de captura, la intensidad de luz y la ubicación del foco. Con los datos obtenidos, se calculó la media y la desviación estándar con respecto a un valor esperado utilizando una cinta milimétrica ubicada en la imagen, lo cual facilitó la medida manual realizada.

La segunda prueba correspondió a la verificación del seguimiento de la pinza quirúrgica, en ella se obtuvieron imágenes de video en secuencia sobre la herida que permitieron observar que se mueve en profundidad y en el espacio.

Finalmente, la tercera prueba se enfocó en la implementación, en conjunto del posicionamiento de la herida y el rastreo de la pinza quirúrgica, en el ambiente virtual, con el objetivo de corroborar su funcionamiento.

4.0.1. Calibración de la cámara

Con el objetivo de poder cuantificar los datos, se hizo necesario implementar un método para correlacionar los datos entregados por el programa y los medibles físicamente. De esta manera, se realizó la implementación de un programa en Open CV, donde se delimitó una cuadrícula virtual para ubicar el plano de manera perpendicular, alineándose con una base de referencia que conserva igual proporción en sus medidas; en este caso se trata de una impresión en hoja de papel como se observa en la figura 4.1.

El objetivo de utilizar esta cuadrícula es posicionar la cámara de manera perpendicular al plano, haciendo coincidir la cuadrícula virtual diseñada en open CV con la cuadrícula real en papel. De esta manera, se realiza manualmente la correlación, colocando exactamente ambas cuadrículas en la misma posición. Esto permite tener centrada la herida con respecto a un punto en una de las esquinas de la plantilla, elegido previamente.

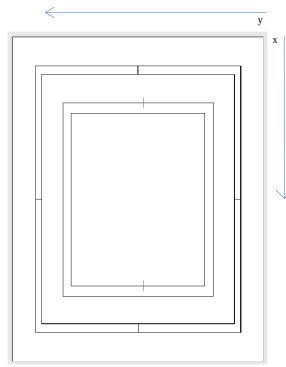


Figura 4.1: Plantilla usada para calibrar dispositivo de captura paralelo a un plano de referencia según su distancia

Fuente: Elaboración propia

4.1. Cálculo del valor esperado medido por cinta milimétrica

La siguiente imagen muestra como se encontraron los valores esperados para los puntos máximo y mínimo en 'x', 'y' y 'z'. Para ello, se utilizó una imagen con la máxima resolución RGB permitida por el dispositivo (1024x720), ubicando los puntos sobre una cinta milimétrica adherida a la plantilla de manera simplificada, pero efectiva. El valor medido en z sobre la almohadilla fue de 8 mm, mientras que la profundidad de las heridas fue de 4 mm. Para todos los análisis se tomó el valor esperado en 'z' como 8 mm, al nivel del contorno exterior de la herida.

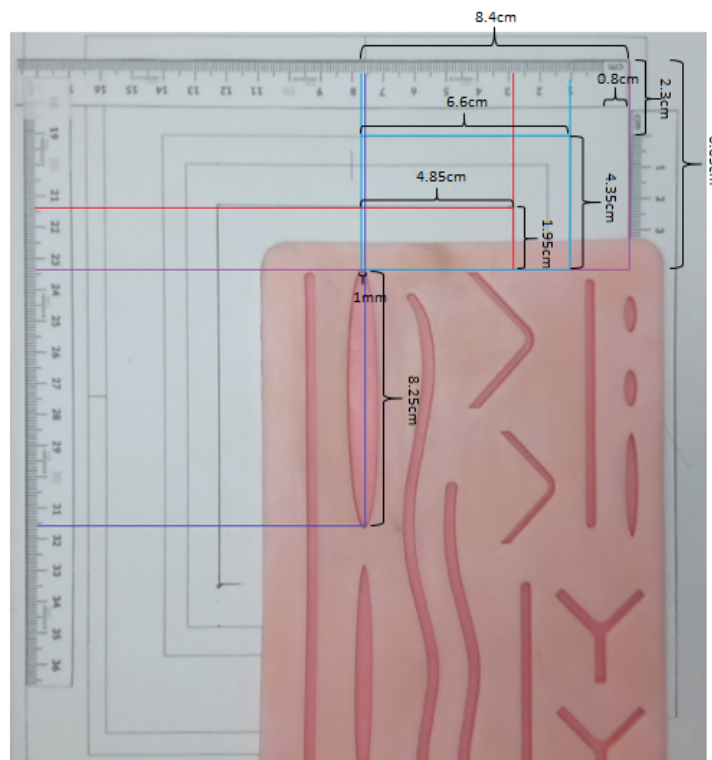


Figura 4.2: Medidas del valor esperado, gráfica con resolución 1024x720

Fuente: Elaboración propia

4.1.1. Prueba de precisión en función de la intensidad de luz y distancia de captura

Para la primera prueba se realizaron tres mediciones, cada una a diferente distancia e intensidad de luz medida por un luxómetro. Para ello, se ubicó cada uno de los elementos en la escena, obteniendo como resultado una muestra de 1000 datos para cada uno de los puntos observados (punto máximo y mínimo) en sus coordenadas 'x', 'y', y 'z'. Los parámetros de la prueba fueron los siguientes:

- Distancias: 17.1 cm, 23.6 cm, 28.6 cm
- Luxes: 500
- Sentido: vertical

A continuación, se observa en la figura 4.3, las pruebas realizadas ubicando el dispositivo en tres distancias distintas.

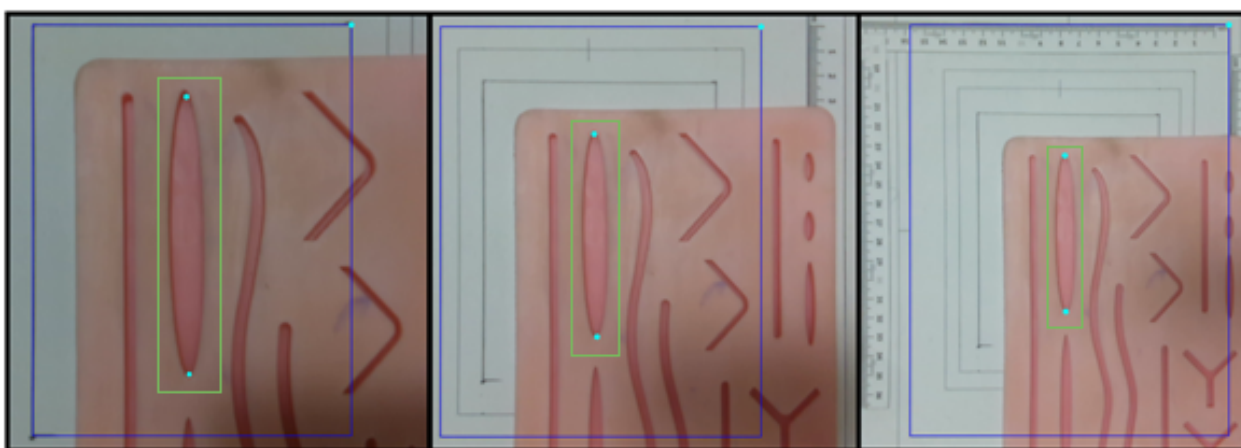


Figura 4.3: Distancias a la herida: 1: 17.1 cm; 2: 23.6 cm; 3: 28.6 cm

Fuente: Elaboración propia

Posición Distancia	Desviación estandar punto inferior			Desviación estandar punto superior		
	X	Y	Z	X	Y	Z
171mm	1.31946959	1.52643375	1.36124943	1.41031911	1.15325626	1.5601282
236mm	1.58524446	1.76635217	1.11579568	1.62265215	1.55884573	1.31491445
286mm	1.9851952	2.02731349	1.13137085	2.10309296	1.94164878	1.18995798

Tabla 4.1: Desviación estandar en mm, lux 500

Fuente: Elaboración propia

En la tabla 4.1 se observa una relación proporcional directa entre la distancia del dispositivo de captura y la desviación estándar para las coordenadas 'x' y 'y' del punto máximo y mínimo. Sin embargo, la varianza sobre 'z' presenta una proporción inversa, teniendo los valores menos dispersos para la mayor distancia de captura.

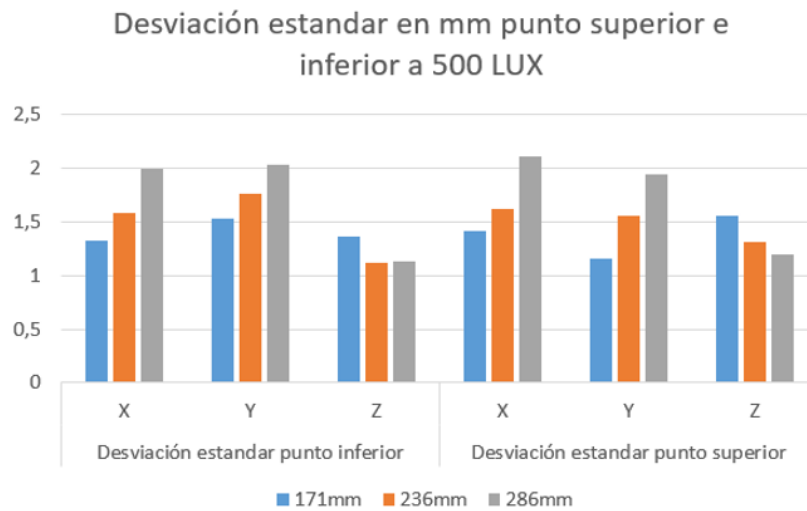


Figura 4.4: Desviación estándar a 500 lux
Fuente: Elaboración propia

Con respecto a la segunda medición, se efectuó de igual forma el procedimiento anterior, pero variando la intensidad de luz de 500 luxes a 1500 lux, los parámetros de esta prueba fueron los siguientes:

- Distancia: 17.1 cm, 23.6 cm, 28.6 cm
- Luxes: 1500
- Sentido: vertical

A continuación se observa en la figura 4.5, las pruebas realizadas a las diferentes distancias.

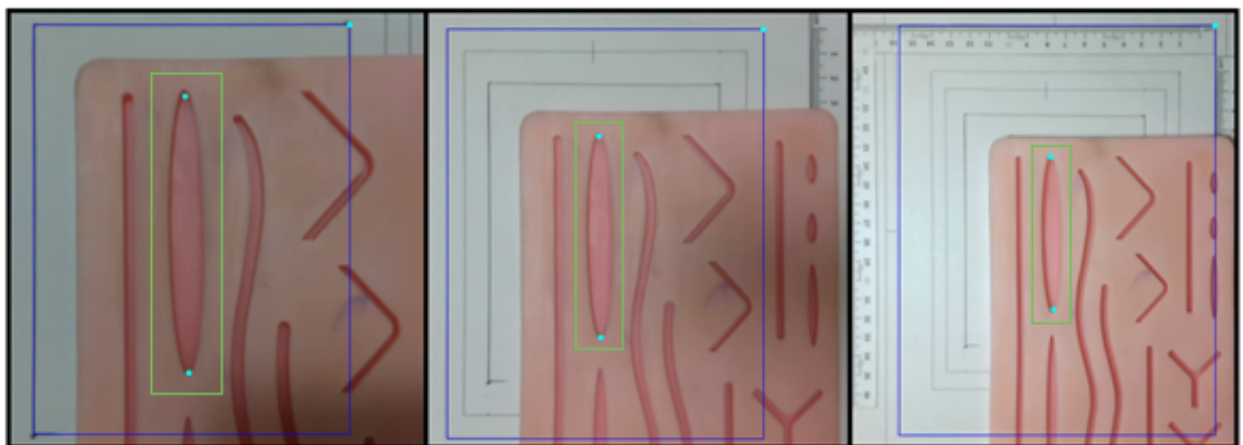


Figura 4.5: Distancias a la herida: 1: 17.1 cm; 2: 23.6 cm; 3: 28.6 cm
Fuente: Elaboración propia

Como se ve en la tabla 4.2, al igual que para 500 Lux, el valor de varianza de los datos, mantiene una proporción directa con respecto a la distancia en las coordenadas 'x' y 'y'. La dispersión en 'z' es de igual forma inversamente proporcional a la distancia.

Para la última prueba de este apartado, se varió el nivel de intensidad de luz a 2000 lux. Por lo tanto, los parámetros de esta prueba fueron los siguientes:

Posición Distancia	Desviación estandar punto inferior			Desviación estandar punto superior		
	X	Y	Z	X	Y	Z
171mm	1.07734241	1.24632794	1.89195666	1.15152073	1.08934231	2.10973932
236mm	1.29434668	1.44222051	1.50582203	1.32488993	1.27279221	1.75883484
286mm	1.52556001	1.65529454	1.34565969	1.71716821	1.48772757	1.70997076

Tabla 4.2: Desviación estandar en mm, lux 1500
Fuente: Elaboración propia

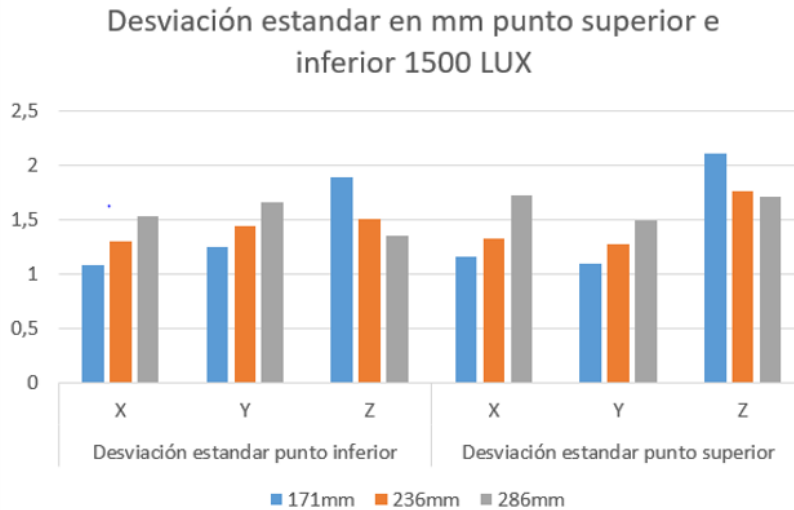


Figura 4.6: Desviación estándar a 1500 lux
Fuente: Elaboración propia

- Distancia: 17.1 cm, 23.6 cm, 28.6 cm
- Luxes: 2000
- Sentido: vertical

A continuación se observa en la figura 4.7, las pruebas realizadas con las diferentes distancias.

Posición Distancia	Desviación estandar punto inferior			Desviación estandar punto superior		
	X	Y	Z	X	Y	Z
171mm	0.90408567	1.05084033	0.6164414	0.97788036	0.81547532	0.78485667
236mm	1.08619161	1.21600698	0.6670832	1.12510683	1.10227038	0.96384646
286mm	1.36023334	1.28369284	0.68044103	1.4582326	1.37295302	0.96643675

Tabla 4.3: Desviación estándar en mm, Lux 2000
Fuente: Elaboración propia

Al igual que en los dos anteriores análisis, la dispersión de los datos a 2000 LUX se mantiene proporcional para las coordenadas 'x' y 'y' de los puntos. Sin embargo, en este caso, 'z' también es proporcional. De igual forma, se observa un valor de dispersión considerable menor en la coordenada z comparada con las mediciones a 500 y a 1500 lux.

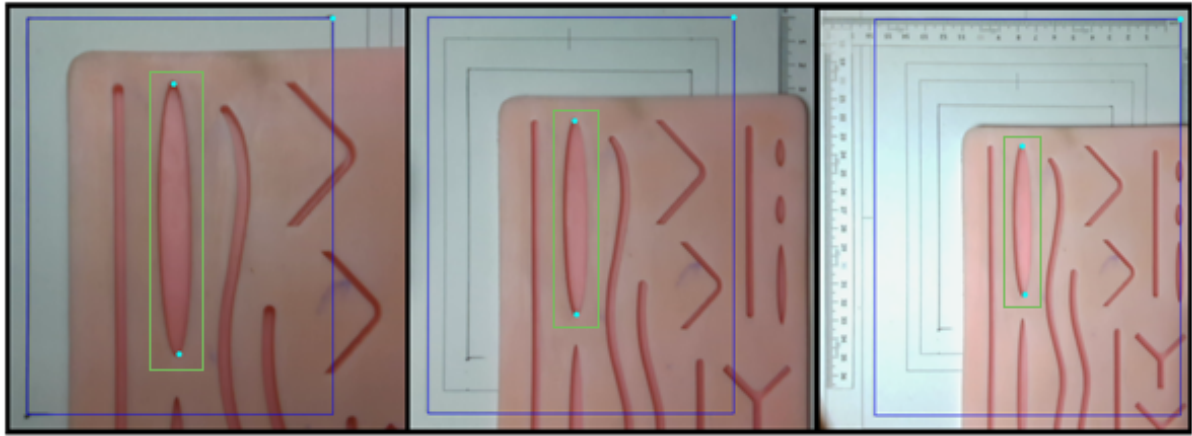


Figura 4.7: Distancias a la herida: 1: 17.1 cm; 2: 23.6 cm; 3: 28.6 cm
Fuente: Elaboración propia

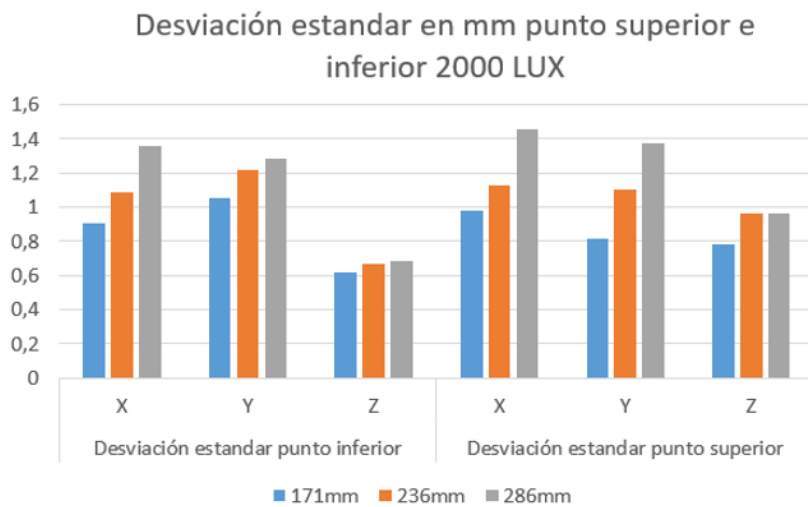


Figura 4.8: Desviación estándar a 2000 lux
Fuente: Elaboración propia

4.1.1.1. Prueba con la cámara en sentido horizontal

En la siguiente prueba, se realizó el cálculo de la dispersión de las posiciones de los puntos máximo y mínimo, variando la distancia, a una intensidad de luz fija. La realización de la prueba buscaba evaluar si el sistema era más preciso al hacer la correlación de un punto sobre el eje horizontal o sobre el eje vertical.

Parámetros de la medición.

- Distancia: 17.1 cm, 23.6 cm, 28.6 cm
- Luxes: 2000
- Sentido: Horizontal

A continuación, se observa en la figura 4.9, las pruebas realizadas a diferentes distancias.

La tabla 4.4, presenta los valores de dispersión ya observados en la tabla 4.3, para la coordenada en 'z', la cual presentó un gran contraste con los datos obtenidos al situar la cámara de manera horizontal a la herida.

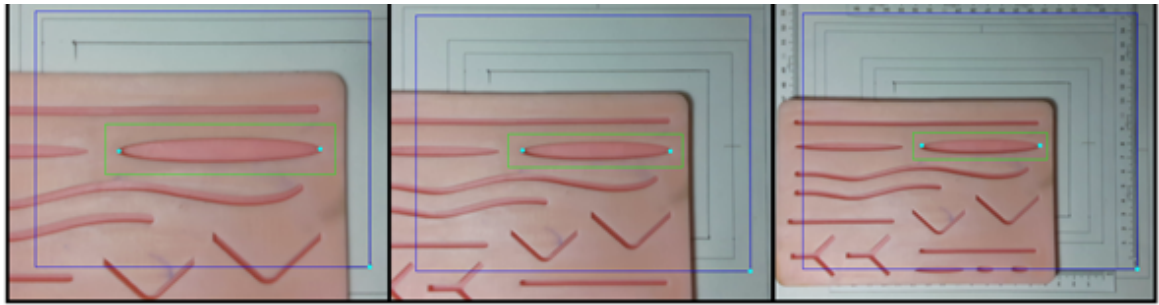


Figura 4.9: Distancias a la herida: 1: 17.1 cm; 2: 23.6 cm; 3: 28.6 cm
Fuente: Elaboración propia

Posición \ Distancia	Desviación estandar punto inferior		Desviación estandar punto superior	
	Vertical	Horizontal	Vertical	Horizontal
171mm	0.6164414	7.110239096	0.784856675	7.449798655
236mm	0.667083203	8.096635104	0.963846461	7.78174145
286mm	0.680441033	6.80940526	0.966436754	8.136983471

Tabla 4.4: Comparación en dispersión, cámara en vertical y horizontal; lux 2000
Fuente: Elaboración propia

Con relación a lo anterior, sobre los datos de profundidad capturados en 'z' se observa de manera muy significativa una varianza menor con la cámara ubicada en sentido vertical tanto para el punto superior como para el punto inferior.

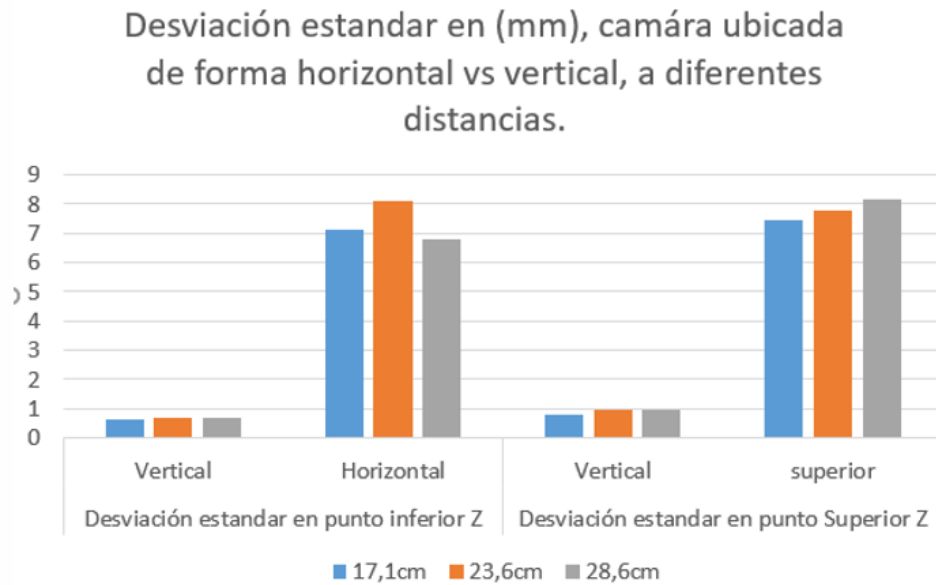


Figura 4.10: Desviación estándar
Fuente: Elaboración propia

4.1.1.2. Error absoluto en posición 'x', 'y' y 'z' del punto máximo y mínimo del contorno analizado

A continuación se presentan las tablas del valor promedio para los 1000 datos tomados por muestra y la tabla de error absoluto.

Posición Distancia	Error absoluto punto inferior			Error absoluto punto superior		
	X	Y	Z	X	Y	Z
171mm	0.659734795	0.763216876	0.68062	0.70515956	0.57662813	0.7800641
236mm	0.79262223	0.883176087	0.5579	0.81132608	0.77942286	0.65745722
286mm	0.992597602	1.013656747	0.56569	1.05154648	0.97082439	0.59497899

Tabla 4.5: Valor promedio obtenido de una muestra de 1000 datos, lux 500, sentido vertical

Fuente: Elaboración propia

Posición Distancia	Error absoluto punto inferior			Error absoluto punto superior		
	X	Y	Z	X	Y	Z
171mm	0.538671205	0.62316397	0.94598	0.57576037	0.54467115	1.05486966
236mm	0.647173341	0.721110255	0.75291	0.66244497	0.6363961	0.87941742
286mm	0.762780003	0.827647268	0.67283	0.85858411	0.74386379	0.85498538

Tabla 4.6: Valor promedio obtenido de una muestra de 1000 datos, lux 1500, sentido vertical

Fuente: Elaboración propia

Posición Distancia	Error absoluto punto inferior			Error absoluto punto superior		
	X	Y	Z	X	Y	Z
171mm	0.452042833	0.525420166	0.30822	0.48894018	0.40773766	0.39242834
236mm	0.543095803	0.608003492	0.33354	0.56255342	0.55113519	0.48192323
286mm	0.68011667	0.641846418	0.34022	0.7291163	0.68647651	0.48321838

Tabla 4.7: Valor promedio obtenido de una muestra de 1000 datos, lux 2000, sentido vertical

Fuente: Elaboración propia

4.1.1.3. Resultados y consideraciones a tener en cuenta en la puesta en marcha del sistema de localización de tejido epidérmico

Con las anteriores pruebas se evaluó la precisión del algoritmo encargado de segmentar la herida, se observó que la varianza de los datos se reduce proporcional a la cercanía del dispositivo de captura y el plano observado con respecto a las coordenadas 'x' y 'y'. Por otro lado, con respecto a las coordenadas en 'z' la varianza aumenta esto es debido a que, como se evidenció en el capítulo de diseño, la cámara capta mejor la profundidad en cercano, como el valor esperado de profundidad se fijó en 0.8 cm y la profundidad de la herida es cercana a los 0.4 cm muchos de los datos se tomaron en 0.4 cm y no en 0.8, lo cual no se consideraría un problema si se hace el debido procesamiento a los datos, por ejemplo, si se quiere, se podría ubicar la extracción de los datos del contorno desplazada algunos píxeles hacia afuera.

Con respecto a la ubicación de la cámara, se observó que trabaja mucho mejor en la captura de profundidad ubicada en sentido vertical a la herida, esto tiene que ver con su técnica de abstracción, la cual al ser por cámaras estéreo, logra mejor correlación de los elementos ubicados en el centro.

Sobre la intensidad y proyección de luz, según las recomendaciones, la cámara funciona muy bien en ambientes iluminados y distancias cercanas. Sin embargo, como se muestra en la tabla de error absoluto, tabla 4.6, para distancias cercanas a 1500 lux el error absoluto es mayor que a 500 luxes, tabla 4.5 y vuelve a ser bajo en 2000 lux, tabla 4.7; esto se puede deber a la proyección del foco de iluminación, cuando se ubica totalmente perpendicular la luz rebota sobre la almohadilla y disminuye la percepción de la cámara por luz cruzada, algo que no pasa para 2000 luxes debido a que para estas mediciones se ubicó una lámpara inclinada (para que no afectara el campo de visión de la cámara).

4.2. Prueba de localización y seguimiento de la punta del instrumental

Correspondiente a la prueba de localización y seguimiento del instrumental, se trabajó con el plano de referencia utilizado en las pruebas de precisión del algoritmo de localización de tejido epidérmico y una pinza para cirugía laparoscópica. De igual forma, se usó la almohadilla de sutura quirúrgica y la cámara realsense D435, en la figura 4.11 se muestran los elementos mencionados.

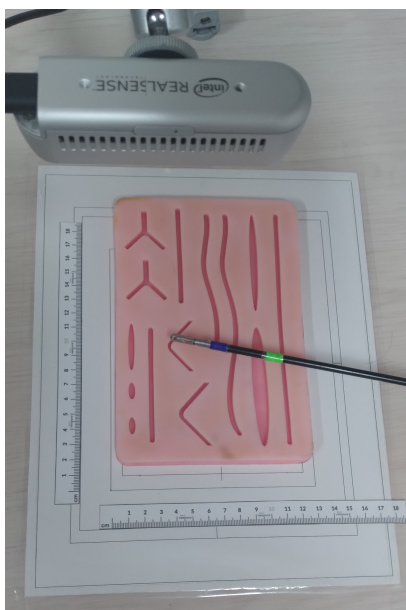


Figura 4.11: Ubicación de los elementos para el seguimiento de la pinza quirúrgica
Fuente: Elaboración propia

A continuación se describe el proceso: Con la calibración de la cámara con respecto al punto del plano de referencia, se ubica una escala de medición desde 1 cm hasta los 18 cm, ya que el phantom se encuentra dentro de ese rango y será posible determinar, en cada momento, la distancia real de la pinza con respecto al punto del plano de referencia. Se realizó una captura de frames de un video con el objetivo de validar el funcionamiento del seguimiento de la pinza quirúrgica, la cual tiene los marcadores que permiten realizar dicha acción.

A continuación se muestran los frames obtenidos. Para todos los frames se muestran numéricamente en la pantalla, los valores x , y , z en color verde, a la que actualmente se encuentra la pinza, también se tiene el plano de referencia en color azul y los puntos máximo y mínimo de la herida en azul celeste, los marcadores se identifican por su color azul y verde, y en el extremo de la pinza es generado un punto de color verde por el programa, el cual indica que a medida que la pinza se desplaza, se va realizando el seguimiento a ese punto. Ver figura 4.12 y 4.13.

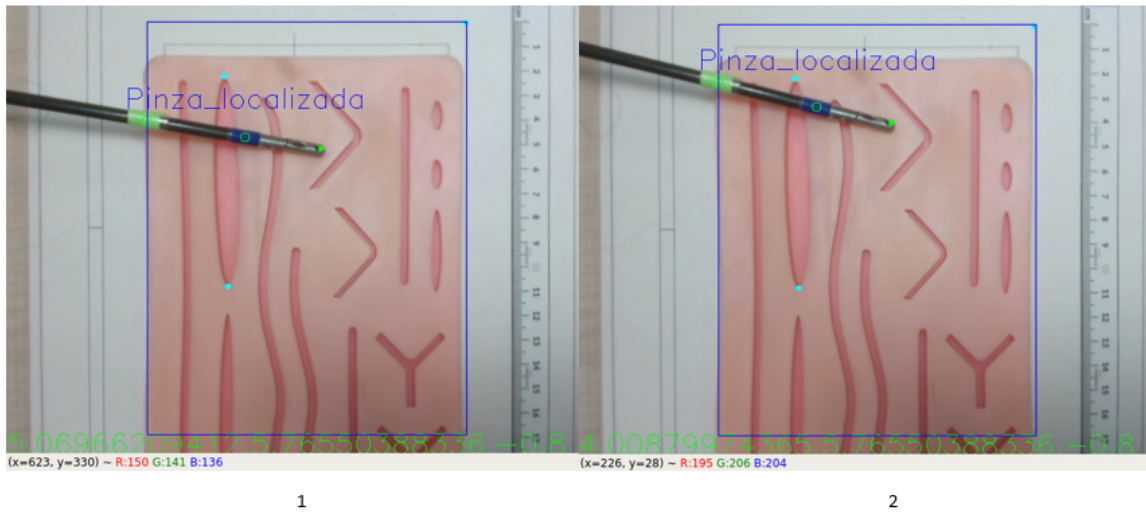


Figura 4.12: Seguimiento de la pinza quirúrgica: 1: frame 1; 2: frame 2
Fuente: Elaboración propia

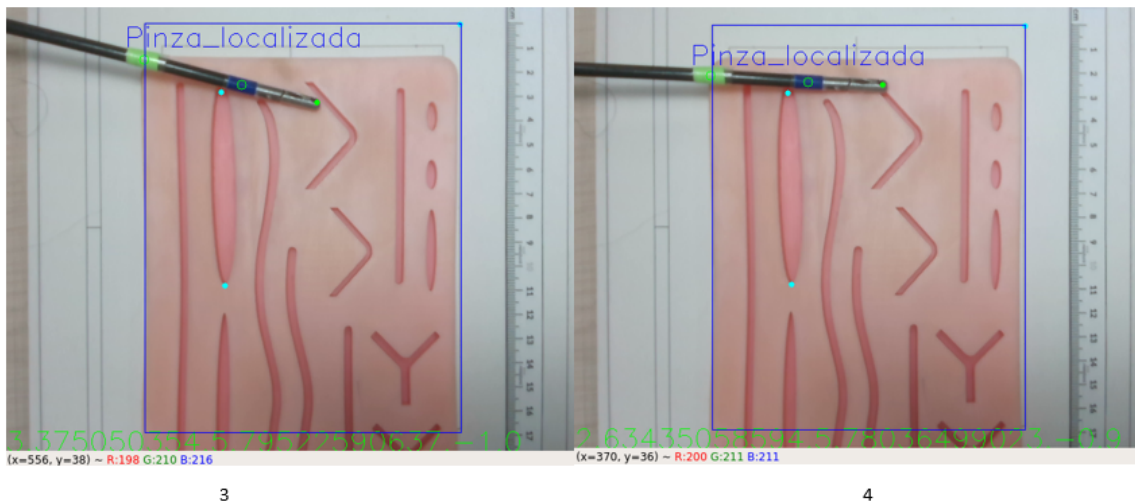


Figura 4.13: Seguimiento de la pinza quirúrgica: 3: frame 3; 4: frame 4
Fuente: Elaboración propia

Como primera prueba, para esta segunda parte se movió la pinza por el plano x,y de la almohadilla, con respecto a puntos medidos con anterioridad en el mundo real de manera manual, se comparó estos valores con los obtenidos por los valores obtenidos de la posición de la punta de la pinza dados por el programa, los resultados en terminos del error absoluto se muestran en la figura 4.14

Como segunda prueba, se compararon los datos de la distancia entre los puntos extremos de la herida y la ubicación de la punta del instrumental reales medidos de manera manual con respecto a los obtenidos por el programa, los resultados se observan en la figura 4.15

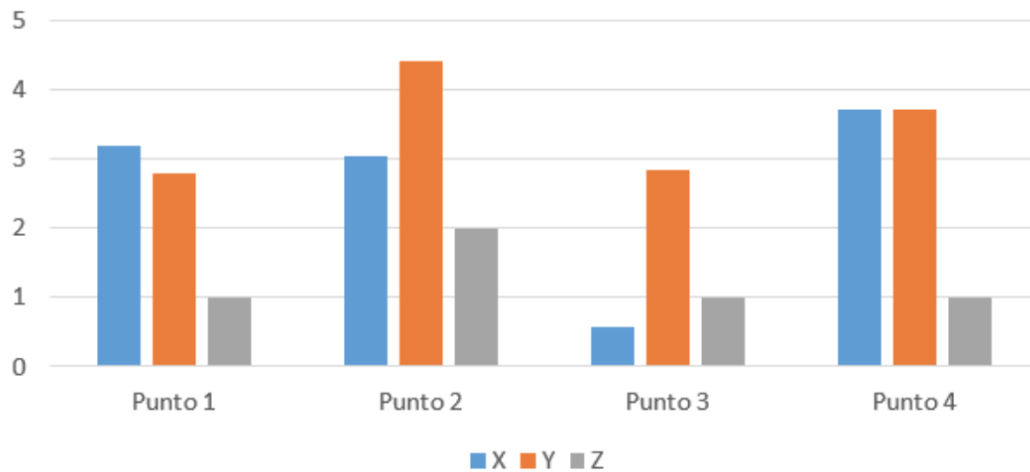


Figura 4.14: error absoluto de la posición de la punta en los puntos 1,2,3 y 4 en mm
Fuente: Elaboración propia

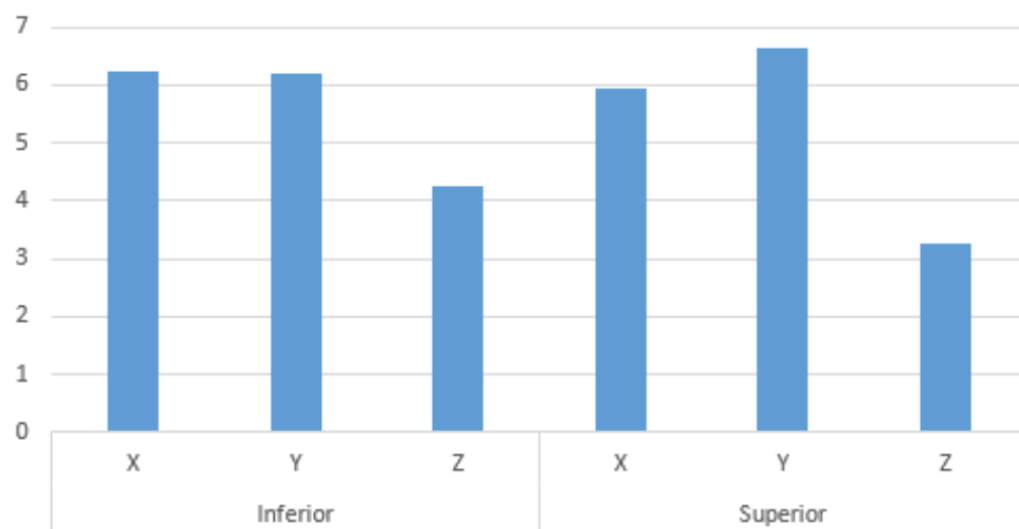


Figura 4.15: Error absoluto, distancia de la punta del instrumental al punto inferior y superior de la herida

Fuente: Elaboración propia

Con respecto a la figura 4.14 se observa que el error absoluto no supera los 5 mm en el peor de los casos y en el mejor de los casos es inferior a 3 mm. Datos que demuestran que existe un mayor nivel de dispersión con respecto a la localización de los puntos extremos de la herida evidenciados en la tabla 4.7. Por otro lado, los resultados obtenidos en la figura 4.15 muestran un incremento en el error, el cual se entiende, comprendiendo que tanto la localización de la herida como la localización de la punta del instrumental presentan un nivel de error de manera separada.

4.3. Prueba en conjunto, instrumental y herida

Para la realización de la tercera prueba, se utilizó Unity como entorno de emulación. Para ello, se emuló el phantom y la plantilla en tiempo real. De esta manera, el punto máximo y mínimo es representado por medio de esferas de color azul y verde que emulan la posición en x, y, z sobre el phantom también abstraído de la realidad con las coordenadas correspondientes. Por otro lado, para el seguimiento de la punta de la pinza se creó una esfera amarilla que emula el desplazamiento en tiempo real de la pinza, como se observa en la figura 4.16

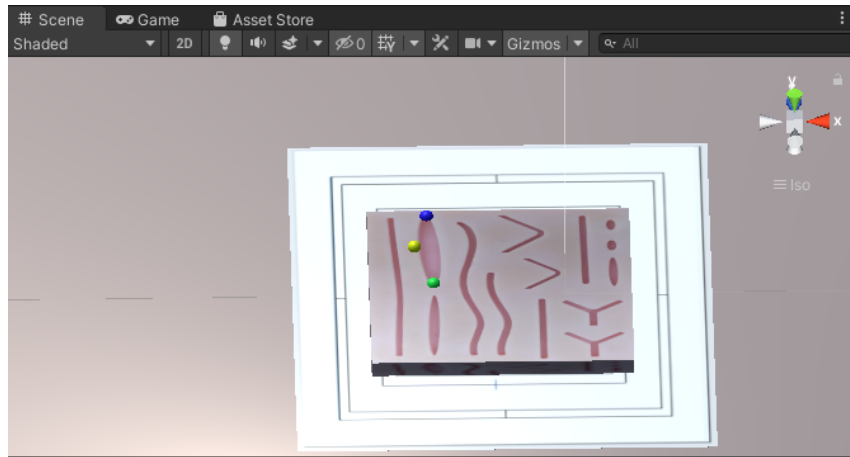


Figura 4.16: Simulación de la escena en Unity
Fuente: Elaboración propia

En las siguientes figuras se ubica una regla, también abstraída para observar la medición de la herida en el mundo virtual, la cual corresponde con la medida en el mundo real, siendo de 8,25 cm, como se observa en la figura 4.2. En la figura 4.17, 4.18, 4.19 se observa el desplazamiento de la punta del instrumental sobre el contorno de la herida sin afectar los puntos máximo y mínimo, pero al igual que en 4.13, en la figura 4.20 el punto máximo superior queda totalmente ocluido y pierde su alineación con respecto a la punta de la herida.

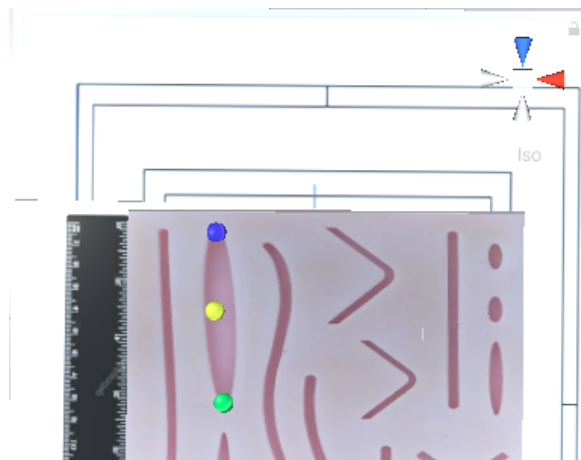


Figura 4.17: Ubicación del punto máximo, mínimo y recorrido de la pinza sobre la herida

Fuente: Elaboración propia

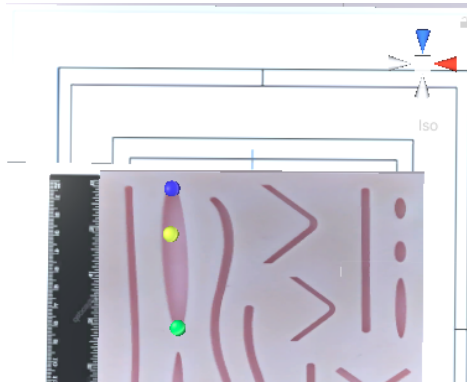


Figura 4.18: Ubicación del punto máximo, mínimo y recorrido de la pinza sobre la herida

Fuente: Elaboración propia

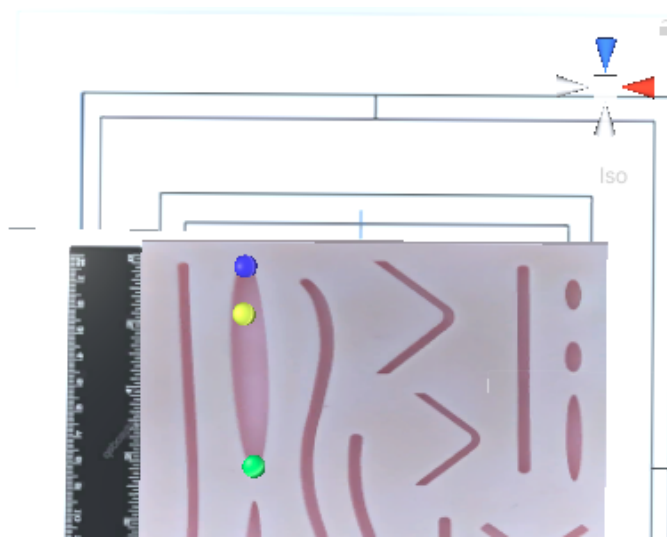


Figura 4.19: Ubicación del punto máximo, mínimo y recorrido de la pinza sobre la herida

Fuente: Elaboración propia

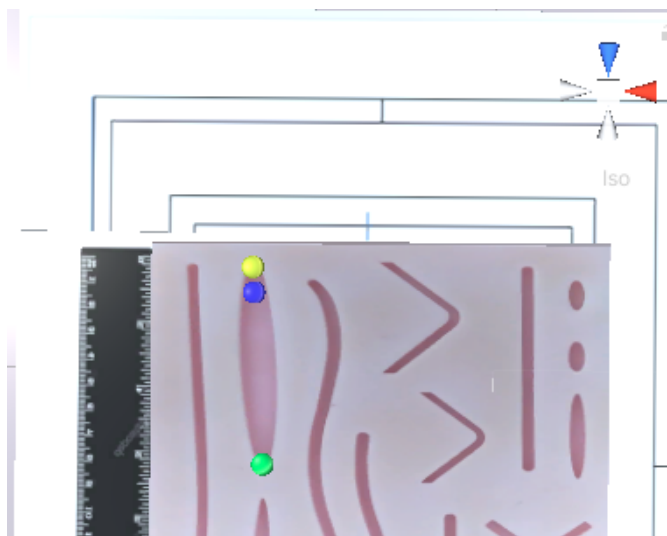


Figura 4.20: Ubicación del punto máximo, mínimo y recorrido de la pinza sobre la herida

Fuente: Elaboración propia

Capítulo 5

Conclusiones y trabajos futuros

En el siguiente capítulo se presentan las conclusiones y trabajos futuros del trabajo de grado titulado: Sistema de localización de tejido epidérmico para automatización de una sutura.

5.1. Conclusiones

Las conclusiones más relevantes del presente trabajo de grado se presentan a continuación:

- Esta investigación tuvo como objetivo proponer un sistema de localización de tejido epidérmico, que involucrara elementos de una escena quirúrgica, orientados a la automatización de sutura. En este sentido, se consiguió implementar un sistema de localización de tejido epidérmico por medio de una segmentación de color con un error absoluto en milímetros, lo cual presenta una buena precisión. De igual forma, se logró hacer seguimiento de una pinza laparoscópica, por medio de marcadores y ubicar la punta de la pinza sobre un marco de referencia común. Con respecto al seguimiento del instrumental, los resultados presentaron un error absoluto menor a 3mm para el mejor caso, por otro lado con respecto al error absoluto de la posición del instrumental y los puntos extremos de la herida en conjunto, los resultados no fueron tan buenos, obteniendo un error absoluto de casi 7 mm, por lo que se sugiere utilizar un método más robusto para ubicar el instrumental.
- En este trabajo se observó que las condiciones ambientales como la iluminación y espaciales como la ubicación del dispositivo de captura son fundamentales en el momento de construir un sistema de visión por computadora, es así que para las distancias más cercanas al plano de referencia se obtuvieron mejores resultados, así como también para los niveles más altos de iluminación.
- Como se observó, el sistema propuesto consigue relacionar el instrumental quirúrgico y el phantom de sutura en un marco de referencia común de manera emulada, situando el punto máximo y mínimo del contorno en una proporción evidenciada por una regla también emulada.

5.2. Trabajos Futuros

Los trabajos futuros que podrían realizarse teniendo como base el presente trabajo de grado son los mencionados a continuación:

- Desarrollar un sistema de localización de tejido epidérmico para automatización de una sutura utilizando algoritmos de segmentación con otras técnicas de inteligencia artificial.
- Utilizar más de un dispositivo de captura y observar su desempeño frente a la oclusión.
- Automatizar el proceso de posicionar la cámara de manera perpendicular al plano o encontrar la mejor posición de captura para esta.
- Desarrollar un sistema para otros tipos de heridas superficiales no uniformes, para su posterior localización, seguimiento y virtualización.

Bibliografía

- [1] J. G. Pereira Fraga, “Actualidad de la cirugía robótica,” *Revista Cubana de Cirugía*, vol. 56, no. 1, pp. 50–61, 2017.
- [2] A. Chanda, T. Ruchti, and V. Unnikrishnan, “Computational modeling of wound suture: A review,” *IEEE Reviews in Biomedical Engineering*, vol. PP, pp. 1–1, 02 2018.
- [3] R. Valero, Y. Ko, S. Chauhan, O. Schatloff, A. Sivaraman, R. Coelho, F. Ortega, K. Palmer, R. Sánchez-Salas, H. Davila, X. Cathelineau, and V. Patel, “Cirugía robótica: Historia e impacto en la enseñanza,” *Actas Urológicas Españolas*, vol. 35, pp. 540–545, 10 2011.
- [4] A. Casals, M. Frigola, and J. Amat, “La robótica, una valiosa herramienta en cirugía,” *Revista iberoamericana de automática e informática industrial (RIAI)*, ISSN 1697-7912, Vol. 6, N^o. 1, 2009, *pag. 5-19*, vol. 6, 01 2009.
- [5] Y. Pei-Lun, C. Hua, H. Yu-Qing, and W. Da-Zhi, “Surgical navigation system based on the visual object tracking algorithm,” 04 2018, pp. 160–164.
- [6] A. Shademan, R. Decker, J. Opfermann, S. Leonard, A. Krieger, and P. Kim, “Supervised autonomous robotic soft tissue surgery,” *Science Translational Medicine*, vol. 8, pp. 337ra64–337ra64, 05 2016.
- [7] J. Martínez-Salamanca, C. Ballesteros, A. Tewari, and J. Carballido, “Cirugía robótica y tecnología láser: una oportunidad para descubrir,” *Archivos Españoles de Urología (Ed. impresa)*, vol. 61, 11 2008.
- [8] X. Chen, L. Xu, Y. Wang, H. Wang, F. Wang, X. Zeng, W. Qiugen, and J. Egger, “Development of a surgical navigation system based on augmented reality using an optical see-through head-mounted display,” *Journal of biomedical informatics*, vol. 55, 04 2015.
- [9] Z. Zhou, B. Wu, J. Duan, X. Zhang, N. Zhang, and Z. Liang, “Optical surgical instrument tracking system based on the principle of stereo vision,” *Journal of Biomedical Optics*, vol. 22, p. 065005, 06 2017.
- [10] A. Shademan, M. Dumont, S. Leonard, A. Krieger, and P. Kim, “Feasibility of near-infrared markers for guiding surgical robots,” *Proc SPIE*, vol. 8840, 09 2013.
- [11] S. Leonard, K. Wu, Y. Kim, A. Krieger, and P. Kim, “Smart tissue anastomosis robot (star): A vision-guided robotics system for laparoscopic suturing,” *Biomedical Engineering, IEEE Transactions on*, vol. 61, pp. 1305–1317, 04 2014.

- [12] D. Filko, R. Cupec, and K. Nyarko, "Detection, reconstruction and segmentation of chronic wounds using kinect v2 sensor," *Procedia Computer Science*, vol. 90, pp. 151–156, 12 2016.
- [13] P. Edgcumbe, P. Pratt, G.-Z. Yang, C. Nguan, and R. Rohling, "Pico lantern: Surface reconstruction and augmented reality in laparoscopic surgery using a pick-up laser projector," *Medical Image Analysis*, vol. 25, 05 2015.
- [14] A. Sagitov, H. Li, and E. Magid, "Toward automated open wound suturing using haptic feedback: detecting wounds and planning the suture," *Proceedings of International Conference on Artificial Life and Robotics*, vol. 23, pp. 208–211, 02 2018.
- [15] Y. Hayashi, K. Misawa, M. Oda, D. Hawkes, and K. Mori, "Clinical application of a surgical navigation system based on virtual laparoscopy in laparoscopic gastrectomy for gastric cancer," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, 10 2015.
- [16] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. W. Kim, "Supervised autonomous robotic soft tissue surgery," *Science translational medicine*, vol. 8, no. 337, p. 337ra64, May 2016. [Online]. Available: <https://doi.org/10.1126/scitranslmed.aad9398>
- [17] B. Lu, H. K. Chu, K. Huang, and J. Lai, "Surgical suture thread detection and 3-d reconstruction using a model-free approach in a calibrated stereo visual system," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 792–803, 2020.
- [18] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. D. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms," pp. 1202–1209, 2015.
- [19] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization," pp. 4178–4185, 2016.
- [20] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to opencv," pp. 1725–1730, 2012.
- [21] G. Bradski and A. Daebler, "Learning opencv. computer vision with opencv library," pp. 222–264, 01 2008.
- [22] [U+FFFFD] Duran and M. Alejandra, "Análisis, diseño e implementación de un sistema de control de ingreso de vehículos basado en visión artificial y reconocimiento de placas en el parqueadero de la Universidad Politécnica Salesiana - Sede Cuenca," p. 144.
- [23] Anestesiados, "Normas de seguridad e higiene en el quirófano," *[En línea]*, Disponible en:<http://anestesiados.com/normas-de-seguridad-e-higiene-en-el-quirofano/>.
- [24] "Iluminación de hospitales según área de atención," *[En línea]*, Disponible en:<https://hospitecna.com/documentacion/iluminacion-de-hospitales-segun-areas-de-atencion/>.

- [25] INFAIMON, “Las 6 mejores tecnicas de iluminacion para la vision industrial,” [*En linea*], Disponible en: <https://blog.infaimon.com/tecnicas-de-iluminacion-sector-industrial/>.
- [26] F. Manteca Fernández *et al.*, “Generación de modelos 3d mediante luz estructurada,” Ph.D. dissertation, Tesis. Popayan: Universidad del Cauca, 2018.
- [27] “El triángulo de exposición: Conoce los tres elementos esenciales para dominar la luz en el modo manual,” [*En linea*], Disponible en: <https://www.dzoom.org.es/los-tres-elementos-que-afectan-a-la-exposicion-en-tus-fotos/>.
- [28] “El concepto de coordenadas homogéneas,” [*En linea*], Disponible en: <https://prateekvjoshi.com/2014/06/13/the-concept-of-homogeneous-coordinates/>.
- [29] “Segmentación de imagenes con el algoritmo watershed,” [*En linea*], Disponible en: <https://unipython.com/segmentacion-imagenes-algoritmo-watershed/>.
- [30] “3d grabcut: Una segmentación de volúmenes basada en la técnica grabcut utilizando la gpu,” [*En linea*], Disponible en: <https://picture.iczhiku.com/resource/paper/syiwpEhwwZIEfMVB.pdf>.
- [31] O. A. V. Albán, *Diseño y control de robots industriales: teoría y práctica*. Elaleph.com, 2010.
- [32] “Modelo waterfall o en cascada,” [*En linea*], Disponible en: https://www.dtagency.tech/cursos/metodologias_gestion_proyectos/tema1-ModeloWaterfall.pdf.
- [33] “Mantenibilidad,” [*En linea*], Disponible en: <https://iso25000.com/index.php/normas-iso-25000/iso-25010/26-mantenibilidad>.
- [34] C. de la salud, “Pinzas laparoscopicas,” [*En linea*], Disponible en: <https://www.catalogodelasalud.com/ficha-producto/Pinzas-laparoscopicas/>.
- [35] “Tutoriales de introducción a opencv-python,” [*En linea*], Disponible en: https://docs.opencv.org/4.x/d0/de3/tutorial_py_intro.html.
- [36] M. C. Chaparro Velasco, “Sistema de iluminación abdominal para cirugía laparoscópica basado en micro robots,” 2019.
- [37] Ubunlog, “Unity 5.3 finalmente llega a linux,” [*En linea*], Disponible en: <https://ubunlog.com/unity-5-3-finalmente-llega-a-linux/>.
- [38] “Ros - robot operating system,” [*En linea*], Disponible en: <https://www.ros.org/>.
- [39] N. Alvarez Casadiego, M. A. Segura Albarracín *et al.*, “Implementación de algoritmos filtro de kalman y filtro de partículas para localización y mapeo simultáneo aplicado a un robot móvil en ambientes interiores con variaciones de iluminación,” 2020.
- [40] S. Giancola, M. Valenti, and R. Sala, *State-of-the-Art Devices Comparison*, 06 2018, pp. 29–39.
- [41] Intelrealsense, “Camara de profundidad intelrealsense d415,” [*En linea*], Disponible en: <https://www.intelrealsense.com/depth-camera-d415/>.

- [42] C.-Y. Chiu, M. Thelwell, T. Senior, S. Choppin, J. Hart, and J. Wheat, “Comparison of depth cameras for three-dimensional reconstruction in medicine,” *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 233, p. 095441191985992, 06 2019.
- [43] Intelrealsense, “Camara de profundidad intelrealsense d435,” [*En linea*], Disponible en: <https://www.intelrealsense.com/depth-camera-d435/>.
- [44] —, “Camara de profundidad intelrealsense d435i,” [*En linea*], Disponible en: <https://www.intelrealsense.com/depth-camera-d435i/>.
- [45] STEREO LABS, “Zed,” [*En linea*], Disponible en: <https://www.stereolabs.com/zed/>.
- [46] —, “Zed2,” [*En linea*], Disponible en: <https://www.stereolabs.com/zed-2/>.

Anexos

Anexos A

Conexión Windows-Linux por medio de Rosbridge

A.0.1. Configuración Servidor:

Para realizar una correcta conexión de ros bridge, se siguen los siguientes pasos en la configuración del servidor.

- El primer paso es descargar de la página en línea el archivo correspondiente:
http://wiki.ros.org/rosbridge_suite/Tutorials//
- Luego se escriben los siguientes comandos en el terminal.
 - `sudo apt-get install ros-<rosdistro>-rosbridge-suite`
 - `source/opt/ros/<rosdistro>/setup.bash`
 - `roslaunch rosbridge_server rosbridge_websocket.launch`
- Se debe tener en cuenta que el archivo launch contiene la información necesaria para lograr hacer una conexión, estableciendo el puerto por el cual se enviarán y recibirán sockets. Ver figura A.1

```
<launch>
  <include file="$(find rosbridge_server)/launch/rosbridge_websocket.launch" >
    <arg name="port" value="8080"/>
  </include>
</launch>
```

Figura A.1: Archivo launch

Fuente: Elaboración propia

- Como siguiente paso se procedió a descargar la carpeta `file_server`, que contiene archivos que permiten que el equipo pueda conectarse en conexión local con otro, sirviendo como servidor en un ambiente ROS:
 - Se descarga el archivo de la siguiente página web https://github.com/tonydle/ur3_unity_sim/tree/master/ros
 - Luego de descargar el archivo, se debe ubicar en el espacio de desarrollo ROS, usualmente conocido como `catkin_ws`, en la carpeta `src`. Lo anterior con el fin de construir el paquete ya guardado en la carpeta de archivos fuente `SRC`. Para ello, en una terminal se debe ubicar `/catkin_ws` y escribir el siguiente comando: `catkin_make`

- Para finalizar esta secuencia de pasos se debe actualizar haciendo `/cdcatkin_ws; sourcedevel/setup.bash`
- Prueba de establecimiento de conexión:
 - Para probar la conexión se ejecuta en la terminal el siguiente comando: `roslaunchrosbridge_serverrosbridge_websocket.launch`. Esto permitirá lanzar y abrir el servidor y posteriormente se comienza a escuchar por medio de sockets en el puerto seleccionado.
 - El siguiente paso es ejecutar el comando `roslaunchfile_serverros_sharp_communication.launch`. Para comprender que sucede de manera gráfica se utiliza el comando `rqt_graph` ingresándolo por consola. Como se puede ver en figura A.2, se han activado los nodos para escuchar por medio de ROS.

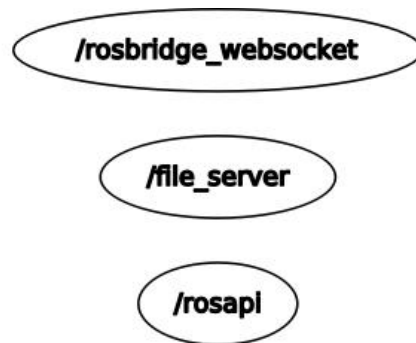


Figura A.2: Conexión ROS
Fuente: Elaboración propia

A.0.2. Configuración Cliente:

A continuación, se presentan los pasos que se han de seguir para la configuración del cliente.

- Se procede a abrir el programa principal en Unity, que se encuentra en el sistema operativo Windows y funcionara como cliente, con el objetivo de realizar la respectiva conexión al servidor ubicado en el sistema operativo Linux. Ver figura A.3



Figura A.3: Interfaz Unity
Fuente: Elaboración propia

- Teniendo la ventana de proyecto abierto en Windows, se localiza en el programa de Unity la ventana de jerarquía y es seleccionado el objeto ROS Connector. Ver figura A.4

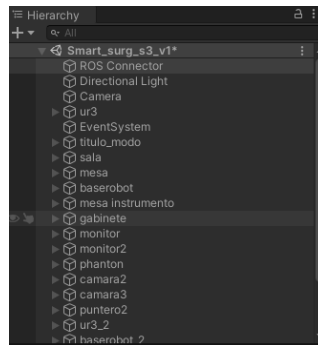


Figura A.4: Objeto Rosconnector
Fuente: Elaboración propia

- En el programa de Unity se ubica la ventana de inspector para visualizar todas las propiedades del objeto seleccionado.
- Ya seleccionado el objeto en la ventana de inspector se busca la propiedad llamada `RosBridgeServerUrl` donde se coloca la ip del computador desde donde se están enviando los datos, en este caso desde un computador que tiene sistema operativo Linux y funciona como servidor. Ver figura A.5

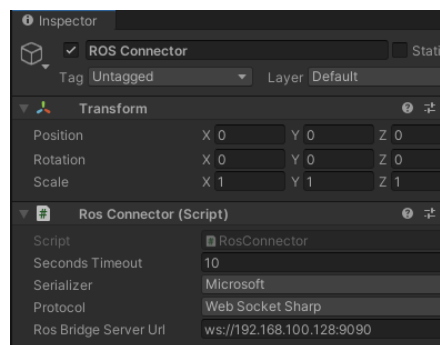


Figura A.5: URL de Linux
Fuente: Elaboración propia

- Para comprobar que se ha realizado una conexión exitosa, ejecutamos el comando `rqt_graph` en el servidor y se obtiene la imagen A.6 de la conexión establecida.

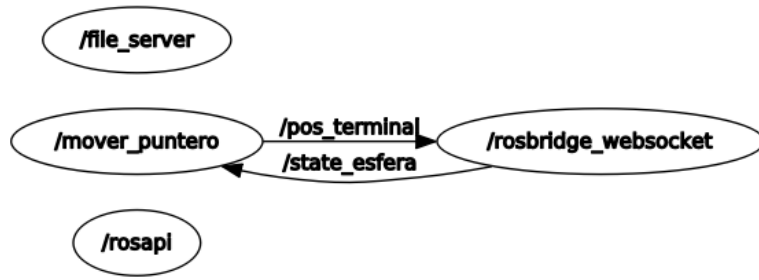


Figura A.6: Conexión exitosa
Fuente: Elaboración propia

Anexos B

Manual de usuario de la interfaz

Para iniciar el programa principal, ejecutamos en el ordenador que contiene al sistema operativo Linux, el archivo main.py, con ello inicialmente se despliega en pantalla la interfaz gráfica creada, que contiene 3 opciones a seleccionar. Ver figura B.1.

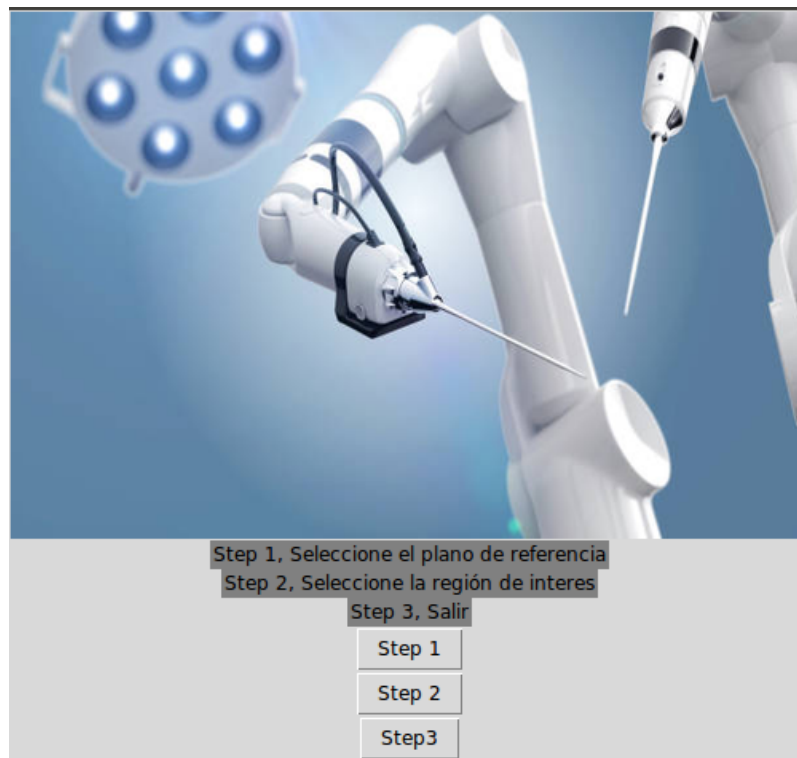


Figura B.1: Inicio de la interfaz
Fuente: Elaboración propia

- La primera opción permite seleccionar con el mouse el punto de referencia, con el cual se realizara la calibración. Esta acción iniciará la cámara y mostrará en una nueva ventana, la imagen visualizada por la cámara en tiempo real, con los elementos dispuestos en la escena de interés como lo son la herida sobre el marco de referencia común y la pinza quirúrgica.
- En la segunda opción se seleccionó la región ROI que contenga la herida. Se delimitó una región de interés, haciendo uso del mouse, para ello se presionó clic derecho sobre cualquier punto en específico en la imagen y se deslizó hasta otro punto cualquiera, con la intención de crear la región de interes sobre la herida, además, inmediatamente se reconocen los puntos máximo y mínimo de la herida.

Si la región de interés se crea mal, es posible presionar la tecla espaciadora, y volver a tener la opción de escoger una región, sin tener que finalizar el programa y volver a ejecutarlo desde consola.

- Por último, en la tercera opción se finaliza el programa completamente y se cierran todas las ventanas abiertas.

Anexos C

Dispositivos de captura 3D

Con el objetivo de realizar un adecuado estudio, donde se identificarán las características de diferentes cámaras que ayudarían en la solución del trabajo planteado, se procede a mencionar los siguientes dispositivos de captura 3D.

Kinect V1: Es un dispositivo que cuenta con una cámara RGB, un sensor de profundidad, un micrófono de múltiples matrices y un procesador con el que ejecuta su propio software de kinect. La captura de imágenes a color se lleva a cabo a través de la cámara RGB. El sensor de profundidad está formado por un generador de láser de infrarrojos y un sensor CMOS monocromo de luz infrarroja. De esta manera se hace posible que la cámara capture video con información en 3 dimensiones bajo cualquier condición de iluminación. [39]



Figura C.1: Kinect V1
Fuente: [40]

Kinect V2: La Kinect V2 es un dispositivo que trabaja con la técnica de tiempo de vuelo (TOF). La Kinect V2 cuenta con una cámara RGB full HD, calibrada con un sensor de profundidad, permitiendo proporcionar mapas de profundidad de colores y nubes de puntos a aproximadamente 30 Hz. [40]

Cámara de profundidad D415: Posee un campo de visión estándar, dando como resultado una mayor resolución de profundidad para objetos más pequeños o cuando es necesario hacer mediciones más precisas. Además, cuenta con sensores de persiana enrollable y unas lentes más pequeñas, logrando una cámara de profundidad de menor costo, pero de alta capacidad. Entre sus características se encuentran un alcance máximo de 10 metros y mínimo de 0,16 metros, influyendo en la precisión, la calibración, la escena y las condiciones de iluminación. Es importante mencionar que esta cámara



Figura C.2: Kinect V2
Fuente: [40]

es más adecuada para escanear objetos a distancias más largas. Entre sus aplicaciones se encuentran la autenticación facial y el escaneo 3D o captura volumétrica. [41]



Figura C.3: Cámara de profundidad D415
Fuente: [40]

Cámara de profundidad D435: Ofrece un campo de visión más amplio comparado con otras cámaras de su tipo, lo que le permitirá minimizar la distancia de captura, logrando mejorar la precisión y confiabilidad [42]. Posee un obturador global en el sensor de profundidad, siendo idealmente utilizada para aplicaciones de movimiento rápido, además de tener alta sensibilidad cuando hay poca iluminación. Es un referente para las aplicaciones como la navegación, robótica o la realidad virtual o aumentada y el reconocimiento de objetos. Cuenta con alcance de hasta 10 metros y una distancia de profundidad mínima de hasta 0,105 metros [43].



Figura C.4: Cámara de profundidad D435
Fuente: [40]

Cámara de profundidad D435i: La cámara de profundidad Intel RealSense D435i

combina las sólidas capacidades de detección de profundidad de la D435 con la adición de una unidad de medición inercial (IMU), lo cual permite que una aplicación refine su conciencia de profundidad en cualquier situación en la que se mueva la cámara. El uso de una IMU facilita el registro y la calibración para los casos de uso del sistema de escaneo portátil, además es importante en campos como los drones, la realidad virtual y realidad aumentada. [44]

Cámara ZED: Es una cámara estéreo pasiva que reproduce la forma en que funciona la visión humana. Usando sus dos ojos, el ZED crea un mapa tridimensional de la escena comparando el desplazamiento de píxeles entre las imágenes izquierda y derecha. La cámara estéreo ZED es el primer sensor que introduce la percepción de profundidad de largo alcance en interiores y exteriores, junto con capacidades de seguimiento de movimiento en 3D, lo que permite nuevas aplicaciones en muchas industrias: drones, robótica, realidad aumentada, realidad virtual, efectos visuales y más [45]. Una de las desventajas de las cámaras ZED es que utiliza únicamente las texturas naturales de la escena para inferir profundidad, por lo que podría no reconstruir adecuadamente si la textura es baja, provocando errores en la estimación de profundidad de hasta 1 metro [40].



Figura C.5: Cámara ZED
Fuente: [40]

Cámara ZED2: Es la primera cámara que utiliza redes neuronales para reproducir la visión humana, llevando la percepción estéreo a un nuevo nivel, combina detección de profundidad avanzada con inteligencia artificial, lo que permite desarrollar inteligencia espacial de vanguardia y reproducir virtualmente el mundo físico, así como la detección de objetos en un contexto espacial. Ha sido diseñada para las aplicaciones más desafiantes, desde navegación y mapeo autónomo, hasta realidad aumentada y análisis 3D. [46]

Anexos D

Emulación de la escena quirúrgica en Unity

Es necesario abrir dos terminales en Linux para realizar la respectiva conexión con Unity. En la figura D.1, se muestra el orden de ejecución.

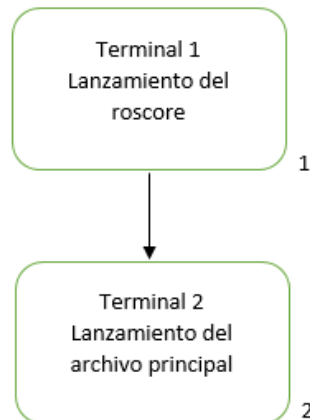


Figura D.1: Etapas de la ejecución: 1: lanzamiento del roscore; 2: lanzamiento del archivo principal

Fuente: Elaboración propia

Como primer paso, es necesario abrir una nueva consola en Linux y ejecutar el comando `roscore`, el cual habilita a ROS para funcionar y permite que los datos sean enviados, de lo contrario no será posible, ver figura D.1.

```
root@osare-Lenovo-C560:~# roscore
... logging to /root/.ros/log/8e058fd2-c5ab-11ec-beee-28e34760
3896/roslaunch-osare-Lenovo-C560-5410.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://osare-Lenovo-C560:34705/
ros_comm version 1.12.17

SUMMARY
=====

PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.17

NODES

auto-starting new master
process[master]: started with pid [5422]
ROS_MASTER_URI=http://osare-Lenovo-C560:11311/

setting /run_id to 8e058fd2-c5ab-11ec-beee-28e347603896
process[rosout-1]: started with pid [5435]
started core service [/rosout]
```

Figura D.2: Inicialización del roscore
Fuente: Elaboración propia

Luego se puede ejecutar en otra consola, el archivo con extensión .py que permite ejecutar el algoritmo principal. El resultado se observa en la figura ??.

```
root@osare-Lenovo-C560:~# roscore
... logging to /root/.ros/log/8e058fd2-c5ab-11ec-beee-28e34760
3896/roslaunch-osare-Lenovo-C560-5410.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://osare-Lenovo-C560:34705/
ros_comm version 1.12.17

SUMMARY
=====

PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.17

NODES

auto-starting new master
process[master]: started with pid [5422]
ROS_MASTER_URI=http://osare-Lenovo-C560:11311/

setting /run_id to 8e058fd2-c5ab-11ec-beee-28e347603896
process[rosout-1]: started with pid [5435]
started core service [/rosout]
```

Figura D.3: Ejecución del archivo .py
Fuente: Elaboración propia

Después es necesario abrir en el Unity hub, en Windows 10, el programa creado para el ambiente quirúrgico llamado conexioncamara, ver figura D.4.

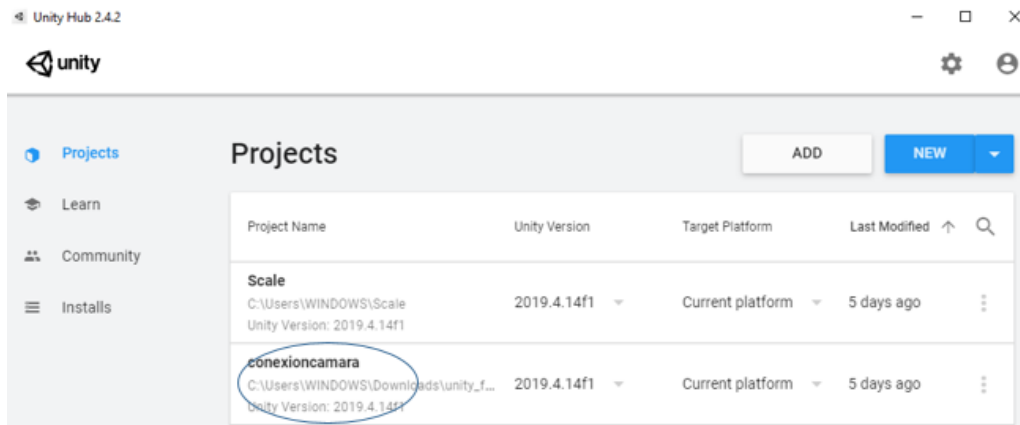


Figura D.4: Inicialización del programa en Unity
Fuente: Elaboración propia

Las respectivas configuraciones realizadas, para comunicar Unity, con ROS, a través de Ros Bridge, se muestran en el anexo A. A continuación, en la figura D.5, se muestra el entorno quirúrgico y los elementos que lo componen.

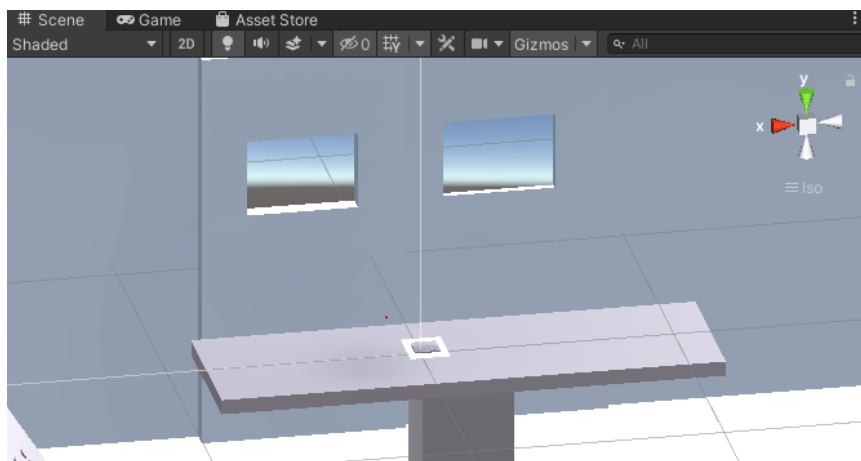


Figura D.5: Escena quirúrgica
Fuente: Elaboración propia

En la terminal de Linux se puede visualizar cuando se ha conectado un cliente al servidor de Unity, ver figura D.6.

```

2022-04-26 16:59:18-0500 [-] Starting factory <autobahn.twisted.websocket.
WebSocketServerFactory object at 0x7fcdcdc81710>
2022-04-26 16:59:18-0500 [-] [INFO] [1651010358.646244]: Rosbridge WebSock
et server started at ws://0.0.0.0:9090
2022-04-26 17:38:36-0500 [-] [INFO] [1651012716.669355]: Client connected.
1 clients total.

```

Figura D.6: Cliente conectado
Fuente: Elaboración propia

Además, se visualiza, a través de que temas se realiza el envío de información, en este caso corresponde a los temas `posición_marcador_pinza` (para el seguimiento de la pinza), `posición_mínima` (para el punto inferior de la herida), `posición_máxima` (para el punto superior de la herida), ver figura D.7.

```
2022-04-26 17:39:47-0500 [-] [INFO] [1651012787.302497]: Client connected.  
1 clients total.  
2022-04-26 17:39:48-0500 [-] [INFO] [1651012788.248913]: [Client 1] Subscr  
ibed to /ur3/follow_joint_traj  
2022-04-26 17:39:48-0500 [-] [INFO] [1651012788.254261]: [Client 1] Subscr  
ibed to /ur3/estop  
2022-04-26 17:39:48-0500 [-] [INFO] [1651012788.259736]: [Client 1] Subscr  
ibed to /ur3/joint_jog  
2022-04-26 17:39:48-0500 [-] [INFO] [1651012788.270731]: [Client 1] Subscr  
ibed to posi_esfer  
2022-04-26 17:39:48-0500 [-] [INFO] [1651012788.278858]: [Client 1] Subscr  
ibed to /pos_terminal  
2022-04-26 17:39:48-0500 [-] [INFO] [1651012788.289602]: [Client 1] Subscr  
ibed to /ur3_iz/follow_joint_traj_iz  
2022-04-26 17:39:48-0500 [-] [INFO] [1651012788.294922]: [Client 1] Subscr  
ibed to posi_esfera  
2022-04-26 17:39:48-0500 [-] [INFO] [1651012788.300206]: [Client 1] Subscr  
ibed to posicion_minimo  
2022-04-26 17:39:48-0500 [-] [INFO] [1651012788.305216]: [Client 1] Subscr  
ibed to posicion_maximo  
2022-04-26 17:39:48-0500 [-] [INFO] [1651012788.316137]: [Client 1] Subscr  
ibed to contornos
```

Figura D.7: Temas
Fuente: Elaboración propia