



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL**

UNIDAD ZACATENCO
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
SECCIÓN BIOELECTRÓNICA

Propuesta de reconstrucción virtual en 3D del posicionamiento de los aplicadores de RF en la terapia por hipertermia; aplicado en la planeación del tratamiento terapéutico de tumores óseos.

Tesis que presenta

Daniel Gerardo Serrano Díaz

para obtener el Grado de

Maestro en Ciencias

en la Especialidad de

Ingeniería Eléctrica

Directores de la Tesis:

Dr. Lorenzo Leija Salas

Dra. Citlalli Jessica Trujillo Romero

Ciudad de México

Octubre 2021

Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado durante la realización de este proyecto de tesis.

Agradezco al Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN) por brindarme la oportunidad de estudiar una maestría.

Agradezco al Dr. Lorenzo Leija y al Dr. Arturo Vera Hernández por brindarme su confianza y abrirme las puertas al mundo de la investigación, así como darme la oportunidad de trabajar con ustedes a lo largo de mi estancia como estudiante de maestría.

Agradezco a la Dra. Citlalli Jéssica Trujillo Romero por su apoyo, asesoría y dirección a lo largo del desarrollo de este trabajo de tesis, así como su disposición para la terminación de este.

Agradezco a mi compañero de laboratorio Rafael por todos sus consejos y comentarios a lo largo del desarrollo de este trabajo.

Dedicatoria

A mis padres Joel Serrano León y Guillermina Díaz Oliva, quienes siempre me han apoyado y aconsejado en las decisiones que he realizado a lo largo de mi vida.

A mi hermana Lilian por estar siempre a mi lado para apoyarme y escucharme.

Índice general

Índice de figuras.....	IX
Índice de tablas.....	XVI
Resumen	XVIII
Abstract.....	XX
1.Introducción	1
2.Antecedentes	4
2.1 Cáncer de hueso.....	4
2.2 Clasificación de tumores de hueso	4
2.3 Tratamientos contra el cáncer de hueso	5
2.3.2 Quimioterapia	6
2.3.3 Radioterapia	7
2.4 Termoterapia oncológica.....	7
2.4.1 Ablación térmica	8
2.4.2 Hipertermia oncológica	8
2.4.2.1 Hipertermia local.....	8
2.4.2.2 Hipertermia regional	9
2.4.2.3 Hipertermia de cuerpo completo.....	10
2.5 Tipos de aplicadores.....	10
2.6 Planeación de los tratamientos de hipertermia (HTP)	11
2.6.1 Generación de modelos anatómicos: segmentación de tejidos.....	12
2.6.2 Obtención del modelo o geometría del aplicador(es).....	12
2.6.2 Modelado electromagnético (EM)	13
2.6.3 Modelado térmico	14

2.6.4 Técnicas de optimización del tratamiento	14
2.7 Reconstrucción 3D.....	15
2.7.1 Campos de aplicación de la reconstrucción 3D a partir de imágenes	15
2.7.2 Métodos de reconstrucción activos y pasivos	15
2.7.3 Algoritmos de conversión 2D a 3D.....	16
2.7.4 Algoritmo <i>Structure from Motion</i> (SfM).....	21
2.7.5 Etapas de la búsqueda de correspondencia del <i>SfM</i>	21
2.7.6 Etapas de la reconstrucción incremental del <i>SfM</i>	22
3.Estado del arte.....	24
3.1 Métodos de reconstrucción 3D.....	24
3.2 Planeación de un tratamiento de hipertermia	26
3.3 Reconstrucción 3D de aplicadores para la planeación de un tratamiento de hipertermia.....	27
3.4 Localización de objetos en un espacio 3D	28
4.Planteamiento del problema.....	29
5.Objetivo general	30
5.1 Objetivos particulares	30
6.Solución propuesta	32
7.Metodología	36
7.1 Evaluación de la factibilidad de uso del aplicador “guía de onda” para la implementación del tratamiento de tumores en la interfaz músculo-hueso.....	36
7.1.1 Estudio paramétrico	38
7.1.2 Modelo electromagnético (EM)	39
7.1.3 Modelo térmico	39
7.1.4 Parámetros de evaluación	40

7.2 Etapa 1: Obtención de las coordenadas del diseño CAD de un aplicador	40
7.2.1 Diseño CAD de un aplicador	41
7.2.2 Obtención de las coordenadas (x, y, z & θ) de los vértices del aplicador ...	41
7.2.2.1 Fases del diagrama de flujo para la obtención de las coordenadas del diseño CAD de un aplicador	42
7.3 Etapa 2: Obtención de las coordenadas del diseño CAD de tres aplicadores ..	47
7.3.1 Diseño CAD de un arreglo de 3 aplicadores	47
7.3.2 Obtención de las coordenadas (x, y, z & θ) de los aplicadores	48
7.3.3 Exportación de las coordenadas (x, y, z) de los vértices de los aplicadores al <i>software</i> de modelado	49
7.4 Etapa 3: Reconstrucción 3D de la emulación de un tratamiento de hipertermia	50
7.4.1 Construcción de los aplicadores propuestos [11]	51
7.4.2 Ubicación de los aplicadores sobre el maniquí	52
7.4.3 Toma de video de los elementos de un tratamiento de hipertermia	53
7.4.4 Reconstrucción 3D utilizando el algoritmo <i>Structure from Motion (SfM)</i> implementado en el <i>software COLMAP</i> [13]	55
7.4.5 Edición de la reconstrucción 3D.....	57
7.4.6 Obtención de la posición (x, y, z & θ) de los vértices de los aplicadores reconstruidos	60
8.Resultados	65
8.1 Evaluación de la factibilidad de uso del aplicador “guía de onda” para la implementación del tratamiento de tumores en la interfaz músculo-hueso.....	65
8.2 Etapa 1: Obtención de las coordenadas del diseño CAD de un aplicador	70
8.2.1 Diseño CAD de un aplicador	70
8.2.2 Obtención de las coordenadas (x, y, z & θ) de sus vértices	71
8.2.2.1 Despliegue del archivo <i>.stl</i> en <i>Matlab</i> y eliminación de vértices duplicados.	71

8.2.2.2	Detección de los vértices del aplicador y sus coordenadas (x, y, z).....	72
8.2.2.3	Selección de un punto arbitrario en el espacio 3D y reubicación de los vértices.....	73
8.2.2.4	Reordenación de los vértices del aplicador.....	75
8.2.2.5	Cálculo del ángulo θ del aplicador	75
8.3	Etapa 2: Obtención de las coordenadas del diseño CAD de tres aplicadores ..	77
8.3.1	Diseño CAD de un arreglo de 3 aplicadores	77
8.3.2	Obtención de las coordenadas (x, y, z & θ) de los aplicadores	79
8.3.2.1	Despliegue del archivo .stl y eliminación de los vértices duplicados. ...	79
8.3.2.2	Detección de los vértices de los aplicadores y sus coordenadas (x, y, z)	80
8.3.2.3	Selección de un punto arbitrario en el espacio 3D y reubicación de los vértices.....	81
8.3.2.4	Reordenación de los vértices del arreglo de aplicadores	83
8.3.2.5	Cálculo del ángulo θ de los aplicadores.....	84
8.3.2.6	Exportación de las coordenadas (x, y, z) de los vértices de los aplicadores al software de modelado	86
8.4	Etapa 3: Reconstrucción 3D de la emulación de un tratamiento de hipertermia	89
8.4.1	Construcción de los aplicadores propuestos [11]	89
8.4.2	Ubicación de los aplicadores sobre el maniquí	90
8.4.3	Toma de video de los aplicadores	91
8.4.4	Reconstrucción 3D utilizando el algoritmo <i>SfM</i> implementado en el <i>software COLMAP</i> [13]	92
8.4.5	Edición de la reconstrucción 3D.....	95
8.4.5.1	Eliminación de vértices duplicados, eliminación de elementos aislados y eliminación de elementos no deseados	95
8.4.5.2	Escalamiento	96

8.4.5.3 Reducción en el número de vértices del modelo.....	97
8.4.6 Obtención de la posición (x , y , z & θ) de los vértices de los aplicadores reconstruidos.....	98
8.4.6.1 Despliegue del modelo ideal y detección de los vértices superiores de los aplicadores.....	99
8.4.6.2 Emparejamiento de los vértices superiores entre el modelo ideal y el modelo completo.....	100
8.4.6.3 Estimación de los vértices de la cara inferior de los aplicadores.....	101
8.4.6.4 Selección del punto de referencia y reubicación de los vértices.....	102
8.4.6.5 Calculo de los ángulos θ de los aplicadores.....	104
8.4.6.6 Exportación de las coordenadas (x , y , z).....	104
9. Discusión.....	109
10. Conclusiones y perspectivas.....	113
11. Resultados académicos.....	116
Referencias.....	117
Apéndice I.....	122
Apendice II.....	135

Índice de figuras

Fig. 1. Etapas de la planeación del tratamiento de hipertermia (HTP). Etapa 1: obtención del modelo anatómico. Etapa 2: obtención del o modelo o geometría del o de los aplicadores. Etapa 3: generación de una configuración adecuada de los aplicadores. Etapa 4: resolución de los modelos electromagnéticos. Etapa 5: resolución de los modelos térmicos.	3
Fig. 2. Aplicadores de energía EM para hipertermia; a) guía de onda (LCA = <i>Lucite Cone Applicator</i>); b) aplicador <i>current sheet</i> ; c) aplicador <i>microstrip</i>	11
Fig. 3. Flujo de trabajo para la implementación de un sistema de planeación del tratamiento de paciente-específico hipertermia. Inicia con la obtención del modelo anatómico del paciente, después se obtienen los modelos de los aplicadores a utilizar, posteriormente se genera una configuración de los aplicadores de acuerdo con la localización del tumor. Se resuelven los modelos electromagnéticos (EM) y térmicos para finalmente generar una optimización de la ubicación y/o configuración de los aplicadores.	12
Fig. 4. <i>Pipeline</i> del SfM incremental [13].	21
Fig. 5. Sistema propuesto para la obtención de las coordenadas de los aplicadores en un tratamiento de hipertermia.....	32
Fig. 6. Flujo de trabajo de la Etapa 1. Inicia con la elaboración un diseño CAD del aplicador propuesto por C. Trujillo et al [11]. Continúa con la obtención las coordenadas del diseño. Después se realiza la reubicación de los vértices en función de un punto de referencia arbitrario. Seguido de la reordenación de estos vértices reubicados para finalmente calcular el ángulo θ	33
Fig. 7. Flujo de trabajo de la Etapa 2. En esencia esta etapa es una extensión de la Etapa 1, implementando un arreglo de tres aplicadores y un cilindro con medidas	

antropométricas además de la adición de la exportación de las coordenadas (x, y, z) de los vértices de los aplicadores al <i>software</i> de modelado.	34
Fig. 8. Flujo de trabajo de la Etapa 3. Este flujo de añade una serie de pasos (recuadros naranjas) para la obtención de la reconstrucción 3D de los elementos involucrados en la emulación de un tratamiento de hipertermia (maniquí y 2 aplicadores) e implementa los pasos realizados en las Etapas 1 y 2 (recuadros azules) para la obtención de las coordenadas de los vértices de los aplicadores.	35
Fig. 9. Modelo 2D propuesto para evaluar la efectividad de la guía de onda en el tratamiento de tumores en la interfaz músculo-hueso. a) guía de onda, b) bolus de agua, c) piel, d) grasa, e) músculo y f) hueso.....	37
Fig. 10. Diseño original del aplicador (guía de onda) C. Trujillo et al. [11].	41
Fig. 11. Diagrama de flujo para la obtención de las coordenadas del diseño CAD de un aplicador.	42
Fig. 12. Orden demostrativo de los vértices de la geometría del aplicador en un programa CAD.	43
Fig. 13. Ejemplo de la suma de las coordenadas de los vértices para su correcto acomodo.	45
Fig. 14. Aplicador de referencia (azul y rojo), aplicador desplegado reubicado (gris y negro), distancias de referencia (verde).	46
Fig. 15. Diseño CAD del arreglo de 3 aplicadores colocados sobre un cilindro con medidas antropométricas.	48
Fig. 16. Exportación de las coordenadas de los vértices de los aplicadores al software de modelado, a) vértices de la geometría del aplicador, b) Pestaña de ajustes del hexaedro en el software del elemento finito.	50

Fig. 17. Reconstrucción 3D de un tratamiento de hipertermia, a) aplicadores (guías de onda) montados sobre el maniquí (paciente), b) reconstrucción 3D de los aplicadores y el maniquí utilizando el algoritmo de <i>SfM</i>	51
Fig. 18. Planos del diseño CAD de los aplicadores utilizados para su manufactura en una cortadora láser.	52
Fig. 19. Maniquí utilizado para emular a un paciente y colocar los aplicadores para emular un tratamiento de hipertermia.....	53
Fig. 20. Interfaz del programa <i>Freeware Free Video to JPG Converter</i> utilizado para extraer las imágenes (<i>frames</i>) del video del maniquí y los aplicadores.	55
Fig. 21. Ventana de configuraciones de la opción de reconstrucción automática en COLMAP [13]......	56
Fig. 22. Diagrama de flujo para la edición de la reconstrucción 3D en Meshlab. Inicia con la eliminación de vértices duplicados, de elementos aislados y de elementos no deseados, después sigue con el escalamiento. En este punto se divide en la obtención del modelo completo y el modelo ideal (cuya diferencia es la reducción en el número de vértices) para la obtención de dos archivos en formato .stl.....	58
Fig. 23. Flujo de trabajo para la obtención de las coordenadas de los aplicadores en la reconstrucción 3D obtenida con el algoritmo <i>SfM</i> . Inicia con el despliegue del modelo ideal y la detección de los vértices superiores para continuar con el emparejamiento de estos con los vértices del modelo completo. Después los vértices encontrados se reordenan para dar paso a la estimación de los vértices inferiores. Posteriormente los vértices (inferiores y superiores) se reubican en función de un punto de referencia para dar paso al cálculo del ángulo θ de cada uno de los aplicadores y finalizar con la exportación de las coordenadas (x, y, z) al <i>software</i> de modelado.	60
Fig. 24. Identificación de la cara superior de un aplicador.....	61

Fig. 25. Vectores utilizados para estimar los vértices inferiores.	63
Fig. 26. Distribuciones de campo eléctrico logradas al utilizar distintos espesores de bolus de agua, a) campo eléctrico normal en la guía de ondas con un bolus de agua destilada de 5 mm de espesor, espesores máximos de grasa (30 mm) y músculo (50 mm), b) campo eléctrico normal en la guía de ondas con un bolus de agua destilada de 20 mm de espesor, espesores máximos de grasa (30 mm) y músculo (50 mm). Una potencia de 50 W en ambos casos.	65
Fig. 27. SWR a diferentes espesores de grasa (eje x) y músculo (indicado en las etiquetas).	66
Fig. 28. Resultados del estudio paramétrico obtenidos con diferentes espesores en el bolus de agua, a) distribución de temperatura con bolus de agua destilada de 5 mm de espesor, b) distribución de temperatura con bolus de agua destilada de 20 mm de espesor.	67
Fig. 29. Resultados del estudio paramétrico obtenidas con diferentes espesores en el bolus de agua a una temperatura de 42°C. a) distribución de temperatura con un bolus de agua destilada de 5 mm de espesor, b) distribución de temperatura con un bolus de agua destilada de 20 mm de espesor.	67
Fig. 30. Perfiles de temperatura registrados en los tejidos con 1 mm de espesor en grasa y valores variables en el espesor del músculo (1 mm, 9 mm, 17 mm, 25 mm, 33 mm y 50 mm), con una potencia de 50 W.	68
Fig. 31. Perfil de temperatura registrado en los tejidos con 9 mm de espesor de músculo y espesor de grasa variable. Aplicando 50 W y 100 W de potencia.	69
.....	70
Figura 32. Temperaturas máximas obtenidas al utilizar espesores constantes de grasa (1 mm) y músculo (9 mm), a dos potencias de funcionamiento (50 W y 100 W). Teniendo un tiempo de aplicación de 60 min y 90 min.	70

Fig. 33. Vistas del diseño CAD del aplicador ideal, a) vista frontal, b) vista lateral, c) vista superior, d) vista trimetrica.....	71
Fig. 34. Aplicador ideal desplegado en Matlab.	72
Fig. 35. Despliegue de los vértices detectados y los vértices originales.	73
Fig. 36. Despliegue del aplicador, los vértices detectados y el punto seleccionado por el usuario (nuevo origen).....	74
Fig. 37. Despliegue de los vértices del aplicador reubicados en función del punto seleccionado (origen del plano) por el usuario.	74
Fig. 38. Coordenadas de cada uno de los vértices del aplicador (8 vértices) después de su reubicación (<i>coord_reubicadas</i>) y después de su reordenación (<i>coordenadas_exportacion</i>).....	75
Fig. 39. Vértices reubicados del aplicador y vértices del aplicador de referencia para al cálculo del ángulo θ	76
Fig. 40. Vistas del diseño CAD del arreglo de 3 aplicadores ubicados sobre un cilindro que emula una pierna humana (medidas antropométricas), a) vista superior, b) vista frontal, c) vista lateral, d) vista trimétrica.	78
Fig. 41. Arreglo de aplicadores desplegado en <i>Matlab</i>	80
Fig. 42. Despliegue de los vértices encontrados y los vértices originales del arreglo de aplicadores.....	81
Fig. 43. Despliegue del arreglo de aplicadores, los vértices encontrados y el punto seleccionado por el usuario (nuevo origen).....	82
Fig. 44. Despliegue de los vértices de los aplicadores reubicados en función del punto seleccionado (origen del plano) por el usuario.	82

Fig. 45. Coordenadas de cada uno de los vértices de cada aplicador después de su reubicación (<i>coor_reubicadas</i>) y después de su reordenación (<i>coordenadas_exportacion</i>).....	83
Fig. 46. Variable theta desplegada en <i>Matlab</i>	84
Fig. 47. Comparación de los vértices de uno de los aplicadores laterales y los vértices del aplicador de referencia.	85
Fig. 48. Comparación de los vértices del aplicador central y los vértices del aplicador de referencia.	85
Fig. 49. Comparación entre los vértices reubicados en <i>Matlab</i> (a) y los vértices desplegados en el <i>software</i> de modelado (b).	87
Fig. 50. Vértices seleccionados para la medición de distancia entre los aplicadores (plano xz).	88
Fig. 51. Aplicador (guía de onda) no funcional construido con MDF.....	90
Fig. 52. Aplicadores colocados sobre el maniquí para emular un tratamiento de hipertermia.	91
Fig. 53. Configuración utilizada para realizar la reconstrucción 3D	93
Fig. 54. Reconstrucción 3D de los aplicadores y el maniquí obtenida en COLMAP. .	95
Fig. 55. Reconstrucción 3D eliminando vértices duplicados y elementos aislados.	96
Fig. 56. Reconstrucción 3D eliminando elementos no deseados.	96
Fig. 57. Reconstrucción 3D escalada, tomando como referencia las uniones entre las caras de los aplicadores.	97
Fig. 58. Vistas del modelo ideal de la reconstrucción 3D de la emulación tratamiento de hipertermia, a) vista frontal, b) vista lateral, c) vista superior, d) vista trimétrica. ...	98

Fig. 59. Despliegue del modelo ideal de los aplicadores reconstruidos.	99
Fig. 60. Identificación de los vértices superiores de los aplicadores reconstruidos..	100
Fig. 61. Vértices emparejados en el modelo completo.	101
Fig. 62. Despliegue de los vértices inferiores (vértices estimados).....	102
Fig. 63. Punto de referencia seleccionado por el usuario en el espacio tridimensional de la reconstrucción 3D.	103
Fig. 64. Vértices reubicados de acuerdo con el punto seleccionado (nuevo origen).	103
Fig. 65. Ángulos de los aplicadores para cada uno de los planos.	104
Fig. 66. Comparación entre los vértices reubicados de la reconstrucción 3D en <i>Matlab</i> (a) y los vértices desplegados en el <i>software</i> de modelado (b).....	105
Fig. 67 Reconstrucción 3D en el <i>software</i> de modelado de los aplicadores fotografiados.	106
Fig. 68. Vértices utilizados para medir la distancia entre los aplicadores, a) vértices seleccionados en la reconstrucción realizada en el software de modelado, b) acercamiento a los vértices seleccionados, c) coordenadas de los vértices seleccionados	107

Índice de tablas

Tabla 1. Algoritmos de conversión 2D a 3D [23] [24].....	17
Tabla 2. Comparación de los algoritmos de conversión [23][24].	20
Tabla 3. Estado del arte de métodos de reconstrucción 3D	24
Tabla 4. Estado del arte de la planeación de un tratamiento de hipertermia	26
Tabla 5. Estado del arte de la reconstrucción 3D de aplicadores para la planeación de un tratamiento de hipertermia.....	27
Tabla 6. Estado del arte de obtención de la posición de un objeto en un espacio 3D.....	28
Tabla 7. Propiedades térmicas y dieléctricas de los medios implicados en el modelo 2D a una frecuencia de 250 MHz [32] [33] [34] [35] [36] [37].....	37
Tabla 8. Espesores de los medios utilizados en el estudio paramétrico [7][34][35][38][39].	38
Tabla 9. Distancia entre los vértices de una guía de onda [11].	43
Tabla 10. Propiedades generales de la cámara integrada en el celular motorola G7 plus [41].	53
Tabla 11. Configuración para la grabación del video.....	54
Tabla 12. Ángulos de los vértices del aplicador reubicado con respecto a los vértices del aplicador de referencia en los diferentes planos.....	76
Tabla 13. Ángulos de los vértices del aplicador central con respecto a los vértices del aplicador de referencia en los diferentes planos.	86
Tabla 14. Distancia entre el diseño CAD del arreglo de aplicadores a partir de dos vértices.	89

Tabla 15. Datos de los videos grabados de distintos objetos para su futura reconstrucción 3D.	92
Tabla 16. Datos de los videos grabados de distintos objetos para su futura reconstrucción 3D.	93
Tabla 17. Distancia entre los aplicadores reconstruidos a partir de dos vértices.	108

Resumen

Los tratamientos contra el cáncer óseo están limitados a la cirugía, la radioterapia y la quimioterapia o a una combinación de ellas. Dentro de los tratamientos emergentes para combatir este tipo de cáncer se encuentra la hipertermia oncológica; en dicha terapia, el tumor se expone a temperaturas entre 40°C y 45°C por tiempos de aplicación entre 60 minutos y 90 minutos. En la práctica clínica, la calidad de un tratamiento de hipertermia se evalúa generalmente por sondas de medición de temperatura colocadas superficial e intersticialmente. Estas sondas de temperatura no son suficientes para generar un mapa tridimensional de las temperaturas alcanzadas en los tejidos. Las distribuciones térmicas volumétricas se pueden predecir mediante modelos computacionales. Por lo tanto, en este proyecto se propone la reconstrucción virtual de un ambiente 3D que emule la ubicación de las guías de onda (aplicadores de RF) en un tratamiento de hipertermia. Es decir, se obtendrán las coordenadas $(x, y, z \text{ \& } \theta)$ de los aplicadores utilizados en un tratamiento de hipertermia a través de una reconstrucción 3D basada en fotografías bidimensionales. Esta información será usada como referencia para un trabajo futuro cuyo objetivo principal radica en la obtención de un sistema de planeación de tratamientos para tumores en la interfaz músculo-hueso. Este proyecto se basa en el uso de una guía de onda diseñada para tratamientos de hipertermia superficial (C. Trujillo *et. al.* 2011). Por lo tanto, se evaluó su fiabilidad de forma preliminar para su uso en el tratamiento de tumores ubicados en la interfaz músculo-hueso. Una vez comprobada su fiabilidad de forma preliminar, este proyecto de tesis se dividió en 3 etapas; la primera de ellas consistió en obtener las coordenadas $(x, y, z \text{ \& } \theta)$ de un solo aplicador en un espacio 3D; en la segunda etapa se obtuvieron las coordenadas $(x, y, z \text{ \& } \theta)$ de un arreglo de aplicadores colocados sobre un cilindro con medidas antropométricas (para emular una pierna humana). Además, se implementó la reconstrucción de la posición de los aplicadores en un *software* de modelado. Finalmente, en la etapa 3 se realizó la reconstrucción 3D de un sistema de dos aplicadores colocados sobre un maniquí para emular un tratamiento de hipertermia. Utilizando esta reconstrucción; se obtuvieron las coordenadas $(x, y, z \text{ \& } \theta)$ de los aplicadores. Además, utilizando estas coordenadas, la posición de los

aplicadores fue reconstruida en el *software* de modelado. Como producto final se obtuvieron las coordenadas $(x, y, z \ \& \ \theta)$ de los vértices de dos aplicadores reconstruidos virtualmente a través de una serie de fotografías o imágenes proporcionadas por la grabación de un video de ellos. Las coordenadas (x, y, z) son las únicas utilizadas para generar una reconstrucción 3D en un *software* de modelado de acuerdo con la forma en que se ingresan las coordenadas de cada uno de los aplicadores en dicho *software*. La orientación de los aplicadores se da de forma automática, haciendo que la obtención de la coordenada θ sea por el momento únicamente informativa.

Abstract

Treatments for bone cancer are limited to surgery, radiotherapy, and chemotherapy or a combination of them. Among the emerging treatments to combat this type of cancer the oncological hyperthermia is found; in this therapy, the tumor is exposed to temperatures between 40°C and 45°C for time applications around 60-90 minutes. In the clinical practice, the quality of a hyperthermia treatment is usually assessed by temperature probes placed either superficially or interstitially. These temperature probes are not enough to generate a three-dimensional map of the temperatures reached in the tissues. Volumetric thermal distributions can be predicted by computational models. Therefore, in this project the virtual reconstruction of a 3D environment that emulates the location of the waveguides (RF applicators) in a hyperthermia therapy was proposed. The coordinates (x , y , z & θ) of the applicators used in a hyperthermia treatment will be obtained through a 3D reconstruction based on two-dimensional photographs. This information will be used as a reference for a future work with main objective is planning the treatment of tumors located at the muscle-bone interface. This project is based on the use of a waveguide designed for superficial hyperthermia treatments (C. Trujillo *et. al.* 2011). Therefore, its feasibility to treat tumors located at the muscle-bone interface was preliminary evaluated. Once its feasibility was verified in a preliminary way, this thesis project was divided into 3 stages; the first one was focused in the obtention of the coordinates (x , y , z & θ) of a single applicator in a 3D space; in the second stage the coordinates (x , y , z & θ) of an arrangement of applicators located over a cylinder with anthropometric measurements (to emulate a human leg) were obtained. Moreover, the reconstruction of the applicator's location was *implemented* in a modeling *software*. Finally, in stage 3, the 3D reconstruction of a system of two applicators located over a mannequin to emulate a hyperthermia treatment, was carried out. By using this 3D reconstruction, the coordinates (x , y , z & θ) of the applicators were obtained; moreover, by using the coordinates, the applicators' location was emulated in the modeling *software*. As a final product, the coordinates (x , y , z & θ) of the vertices of two virtually reconstructed applicators were obtained through a series of photographs or images provided by

recording a video of them. The coordinates (x, y, z) are the only ones used to generate a 3D reconstruction in a modeling software according to the way in which the coordinates of each applicator are entered in the software. The orientation of the applicators is given automatically, so the coordinate θ is for the moment only informative.

1. Introducción

El cáncer es un proceso de crecimiento incontrolado de las células y diseminación de las mismas que puede aparecer prácticamente en cualquier lugar del cuerpo [1]. La mayoría de los cánceres forman una masa referida como tumor o crecimiento, el cual invade el tejido circundante y puede provocar metástasis en puntos distantes del organismo [1]. Para América Latina y el Caribe el número de muerte por cáncer proyecta un incremento de 1.2 millones a 2.1 millones entre el 2008 y el 2030 [2].

Los tipos de cáncer más comunes en los hombres son el cáncer de pulmón, próstata, colorrectal, estómago e hígado; mientras que el cáncer de mama, colorrectal, pulmonar, cervical y de tiroides son los más comunes entre las mujeres [3]. Tratamientos como la quimio y la radioterapia han sido ampliamente estudiados y aplicados para tratar estos tipos de cáncer más comunes. Sin embargo, algunos otros tipos de cáncer menos comunes no han sido ampliamente estudiados. En el caso del cáncer óseo a pesar de no ser un tipo de cáncer común ($< 0.2\%$ de todos los tipos de cáncer [4]), es de especial interés ya que su incidencia se presenta generalmente en personas jóvenes y en edad productiva, por lo cual, es importante contar con tratamientos que no comprometan la calidad de vida del paciente.

El cáncer de hueso puede originarse en cualquier hueso del cuerpo, pero en la mayoría de los casos afecta la pelvis o los huesos largos de los brazos y piernas [5]. La Sociedad Americana Contra el Cáncer estima que en el año 2021 en los Estados Unidos se diagnosticarán alrededor de 3,610 nuevos casos de cáncer óseo [4]; además, se prevén alrededor de 2,060 muertes a causa de estos tipos de cáncer [4].

El tratamiento para el cáncer de hueso depende del tipo de tumor óseo que vaya a ser tratado, su tamaño, localización, etapa del cáncer, así como de la salud general de la persona. Los tratamientos están comúnmente limitados a la cirugía, quimioterapia y radioterapia. Sin embargo, debido a las desventajas y efectos secundarios de los tratamientos tradicionales contra el cáncer óseo que van desde náuseas, vómito,

pérdida de cabello, fatiga, pérdida de apetito, diarrea hasta la amputación del miembro afectado, es necesario buscar nuevas alternativas de tratamiento para este padecimiento. Estas nuevas propuestas deben minimizar estos efectos secundarios, erradicar el tumor y mantener la calidad de vida del paciente.

Dentro de los tratamientos más prometedores para el tratamiento contra el cáncer se encuentra la termoterapia oncológica, la cual se refiere a un aumento en la temperatura del tumor, inducido de manera artificial. Ésta se divide principalmente en ablación térmica e hipertermia oncológica [6]; en esta última, el tumor se expone a temperaturas entre 40°C y 45°C por un tiempo de entre 60 minutos y 90 minutos [6][7][8]. Este tipo de tratamientos ha tomado relevancia en años recientes debido a que busca calentar el área del tumor causando el mínimo de daño a los tejidos sanos contiguos a éste; además de reducir los efectos secundarios sobre el paciente durante y después del tratamiento. De igual forma, éste es un tratamiento mínimamente invasivo o no invasivo, dependiendo de la forma en que se induzca el incremento en la temperatura en el tumor. Las fuentes de calor utilizadas en la hipertemia oncológica se dividen principalmente en dos: ondas mecánicas (ultrasonido) y ondas electromagnéticas (radiofrecuencia, microondas e infrarrojo) [9]; para la aplicación de ellas se utilizan distintos tipos de dispositivos denominados aplicadores.

Para el tratamiento de tumores profundos (más de 4 cm de profundidad), como lo son los tumores óseos, normalmente se coloca alrededor del paciente una matriz de aplicadores (e.g. guías de onda) [10] y el calentamiento se concentra en la ubicación del tumor mediante la selección adecuada de los ajustes de amplitud-fase de los aplicadores individuales [10]. Un método para seleccionar la técnica adecuada de hipertermia a utilizar es por medio de la Planeación del tratamiento de hipertermia (HTP por sus siglas en inglés: Hyperthermia Treatment Planning) [11]. En el HTP se predicen patrones de absorción de energía y distribuciones de temperatura, generados en un paciente mediante una configuración específica del o de los aplicadores [12]. De forma general, un HTP consiste en una serie de 5 etapas (Fig. 1): La primera de ellas es la obtención del modelo anatómico del paciente a tratar. La segunda etapa

corresponde a la obtención del modelo o geometría del o de los aplicadores involucrados. La tercera etapa consiste en la generación de una configuración adecuada de los aplicadores de acuerdo con la localización del tumor. Las etapas 4 y 5 corresponden a la resolución de los modelos electromagnéticos (EM) y la resolución de los modelos térmicos para finalizar con la optimización del HTP, respectivamente.

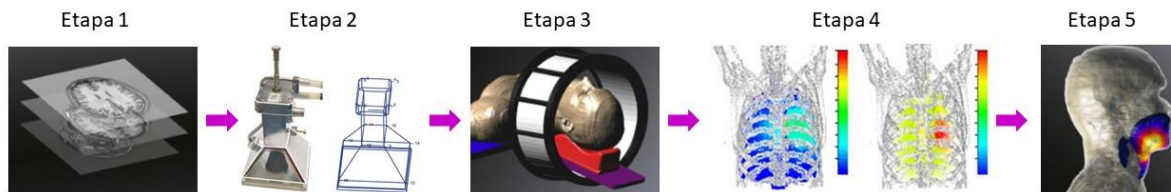


Fig. 1. Etapas de la planeación del tratamiento de hipertermia (HTP). Etapa 1: obtención del modelo anatómico. Etapa 2: obtención del o modelo o geometría del o de los aplicadores. Etapa 3: generación de una configuración adecuada de los aplicadores. Etapa 4: resolución de los modelos electromagnéticos. Etapa 5: resolución de los modelos térmicos.

El presente trabajo se enfoca en la implementación de la etapa 3, con el propósito de obtener la posición (x , y , z & θ) de un conjunto de aplicadores ubicados sobre un maniquí emulando un tratamiento de hipertermia. Dichos aplicadores fueron desarrollados por C. Trujillo et al. [11]. Se propuso el uso de la técnica de *Structure from Motion* (SfM) para reconstruir virtualmente a través de fotografías el aplicador o arreglo de aplicadores con ayuda del programa *COLMAP* [13]. Posterior a la reconstrucción, se realizará una adecuación del modelo obtenido en donde se eliminarán los elementos no deseados. Por otro lado, también es necesario realizar un escalamiento de la reconstrucción 3D para ajustarlo a las dimensiones reales de los aplicadores; además, es necesario obtener un modelo en formato *.stl* para que se obtengan las coordenadas x , y , z & θ de los aplicadores ubicados sobre el maniquí. Esta información (posiciones x , y , z & θ) será utilizada posteriormente para la implementación de un sistema HTP para poder simular los modelos electromagnéticos y térmicos con las configuraciones dadas de los aplicadores y así obtener los patrones de absorción de energía y las distribuciones de temperatura características de un tratamiento específico.

2. Antecedentes

2.1 Cáncer de hueso

Es el crecimiento incontrolado de células en el hueso que se puede diseminar a otras partes del cuerpo [14]. El término “cáncer de hueso” no incluye los tipos de cáncer que se originan en otra parte del cuerpo y se propagan a los huesos. Algunos tipos de cáncer de hueso se manifiestan sobre todo en los niños, mientras que otros afectan principalmente a los adultos [5].

2.2 Clasificación de tumores de hueso

Tumores óseos benignos: los tumores benignos no se propagan a otros tejidos ni a otros órganos, y no suelen representar un riesgo para la vida. Algunos tipos de tumores de hueso benignos son: osteoma osteoide, osteoblastoma, osteocondroma, encondroma y fibroma condromixoide.

Tumores óseos malignos: los tumores óseos malignos pueden diseminarse a otras partes del cuerpo. De este tipo de tumores existen diferentes tipos y su nombre se les asigna en función de la parte de hueso o de tejido adyacente afectada y del tipo de células que forman el tumor [15]. A continuación, se enlistan los diferentes tipos de tumores óseos malignos que existen.

Osteosarcoma: es el cáncer de hueso más común y se originan en las células de los huesos. La mayoría de las veces afecta a personas jóvenes de entre 10 y 30 años, pero alrededor del 10% de los casos de osteosarcoma ocurre en personas de entre 60 y 79 años [15]. Estos tumores se desarrollan con más frecuencia en los huesos de los brazos, las piernas o la pelvis.

Condrosarcomas: se originan en las células del cartílago, y es el segundo cáncer de hueso más común. Es poco común ver este cáncer en personas menores de 20 años,

el riesgo de que se genere un condrosarcoma aumenta hasta que la persona cumple aproximadamente 75 años [15].

Tumor de Ewing: también conocido como sarcoma de Ewing es el tercer tipo de cáncer de hueso primario más común. Es poco frecuente ver este cáncer en adultos mayores de 30 años [15]. Los sitios en los que este cáncer se desarrolla más comúnmente son la pelvis, la pared torácica y los huesos largos de las piernas o los brazos.

Fibrohistiocitoma maligno: se origina con más frecuencia en tejidos blandos (ligamentos, tendones, grasa y músculo). Cuando el fibrohistiocitoma maligno se origina en los huesos, generalmente, afecta las piernas o los brazos. Este tipo de cáncer se presenta con más frecuencia en adultos mayores o de mediana edad [15].

Fibrosarcoma: se trata de un tipo de cáncer que se desarrolla frecuentemente en los tejidos blandos; sin embargo, cuando se origina en los huesos, los más afectados son los de las piernas, brazos y mandíbula [15].

Cordoma: este tumor de hueso se presenta en la base del cráneo y en los huesos de la columna vertebral. Este cáncer se presenta con más frecuencia en adultos mayores de 30 años. Los cordomas suelen crecer lentamente y a menudo no se propagan a otras partes del cuerpo [15].

Tumor de hueso de células gigantes: este tipo de tumor de hueso puede presentarse en forma benigna o maligna. Por lo general, los tumores de hueso de células gigantes afectan las piernas o los brazos de adultos jóvenes y de mediana edad [15]. Estos tumores comúnmente presentan recurrencia local, y con cada recurrencia aumentan las posibilidades de que el tumor se propague a otras partes del cuerpo.

2.3 Tratamientos contra el cáncer de hueso

El tratamiento para el cáncer de hueso depende del tipo de éste, su tamaño, localización, etapa del cáncer, así como la salud general de la persona. Sin embargo, estos tratamientos están comúnmente limitados a la cirugía, quimioterapia y radioterapia [6].

2.3.1 Cirugía

Es el tratamiento más común para el cáncer de hueso, cuyo principal objetivo es remover todo el tumor, ya que, si se dejan incluso unas cuantas células cancerígenas, estas pueden crecer y generar un nuevo tumor. Para minimizar la cantidad de tejido sano extraído junto al tumor, el cirujano debe usar técnicas quirúrgicas específicas. La principal ventaja de la cirugía como tratamiento para este tipo de cáncer es que la remoción del tumor reduce los síntomas de malestar instantáneamente, además de tener la capacidad potencial de eliminar todas las células cancerosas. Es decir, que el paciente podría no requerir ningún otro tratamiento además de la cirugía. Dentro de las desventajas de la cirugía se encuentra el daño al tejido sano cercano al tumor a remover, la remanencia de células cancerígenas en las áreas cercanas a la cirugía y complicaciones durante la cirugía [16].

2.3.2 Quimioterapia

Es el uso de fármacos o drogas para tratar el cáncer. Este tratamiento es sistémico, lo cual implica que, el medicamento ingresa al torrente sanguíneo y circula para alcanzar y destruir las células cancerosas en todo el cuerpo. Dentro de las ventajas de la quimioterapia como tratamiento del cáncer de hueso se encuentran; la habilidad de eliminar células cancerígenas en el tumor principal y en otros tumores dentro del cuerpo, reducción del tamaño del tumor o ralentizar su crecimiento. La principal desventaja de la quimioterapia es que puede causar daño a células sanas, además de causar efectos secundarios durante el tratamiento tales como: náuseas, vómito, pérdida del apetito, pérdida de cabello, etc. [16].

2.3.3 Radioterapia

Utiliza rayos o partículas de alta energía para destruir a las células cancerígenas. Este tratamiento puede también ser utilizado después de la cirugía para eliminar células cancerígenas ubicadas en los bordes del tejido removido. Las ventajas de la radioterapia son: la eliminación de una porción considerable del tumor, capacidad de reducir el tamaño del tumor y la estimulación de la respuesta inmune contra el tumor. Las desventajas de la radioterapia son: daño a tejido circundante al tumor, dependiendo del tratamiento éste puede repetirse varias veces por semana durante un par de meses, además de causar efectos secundarios durante el tratamiento tales como fatiga, pérdida del apetito, náuseas, vómito, diarrea, etc. [16].

La Sociedad Americana Contra el Cáncer estima que en el año 2021 en los Estados Unidos:

- Se diagnosticarán alrededor de 3,610 nuevos casos de cáncer óseo
- Se prevén alrededor de 2,060 muertes a causa de estos tipos de cáncer [4]

Debido a las desventajas y efectos secundarios de los tratamientos tradicionales contra el cáncer óseo, es necesario buscar nuevas alternativas de tratamiento para este padecimiento. Dichas alternativas deben de minimizar los efectos secundarios, erradicar el tumor y mantener la calidad de vida del paciente, ya que este tipo de cáncer se presenta mayormente en personas en edad productiva [6]. Actualmente, diversas investigaciones se enfocan en el desarrollo de técnicas novedosas para inducir calor sobre los tejidos y obtener beneficios terapéuticos. Dentro de las técnicas más prometedoras para el tratamiento contra el cáncer se encuentra la termoterapia.

2.4 Termoterapia oncológica

La termoterapia oncológica se refiere a un aumento en la temperatura del tumor, inducido de manera artificial. Ésta se divide principalmente en ablación térmica e hipertermia oncológica [6].

2.4.1 Ablación térmica

Consiste en incrementar la temperatura del tumor más allá de los 60°C por cortos periodos de tiempo (segundos o minutos) [6]. Su objetivo es eliminar el tumor en su totalidad, quemando las células que lo componen (necrosis coagulativa). La ablación se puede generar mediante las técnicas de radiofrecuencia, microondas y ultrasonido.

2.4.2 Hipertermia oncológica

En esta técnica, el tumor se expone a temperaturas entre 40°C y 45°C por un tiempo de entre 60 minutos y 90 minutos [6][7][8]. El principio de operación consiste en la absorción de energía, la cual produce un incremento en la temperatura del tejido; se ha demostrado que el incremento de temperatura daña y mata las células cancerígenas, causando un daño mínimo al tejido normal [6]. La hipertermia se divide (por localización) principalmente en hipertermia local, hipertermia regional e hipertermia de cuerpo completo. A continuación, se describen cada una de ellas, así como su subclasificación.

2.4.2.1 Hipertermia local

El calor es aplicado en áreas específicas, usando varias técnicas para entregar energía y calentar el tumor. Distintos tipos de energía pueden ser utilizadas para aplicar calor, dentro de las cuales se encuentran las microondas, la radiofrecuencia y el ultrasonido [17]. Dependiendo de la ubicación del tumor, existen distintos enfoques para la hipertermia local:

Enfoque externo: se usa para tratar tumores que estén cerca de la piel (<4 cm de profundidad). Los aplicadores externos se posicionan alrededor o cerca de la región a tratar, y la energía se enfoca en el tumor para incrementar la temperatura [17].

Intraluminal o Endocavitaria: este método se usa para tratar tumores dentro o cerca de las cavidades corporales, tales como el esófago, el recto o la vagina. Las sondas se colocan dentro de la cavidad y se insertan en el tumor para suministrar energía y calentar el área directamente [11].

Intersticial: esta técnica es utilizada para tratar tumores en las profundidades del cuerpo, como los tumores cerebrales. Esta técnica permite que el tumor se caliente a temperaturas más altas de lo que permiten las técnicas externas. Bajo anestesia, se insertan sondas o agujas en el tumor, posteriormente la fuente de calor se inserta en la sonda [17].

2.4.2.2 Hipertermia regional

Se utiliza para tratar grandes áreas de tejido, tales como cavidades, órganos o miembros [11]. Esta se clasifica en hipertermia de tejido profundo, perfusión regional y perfusión peritoneal continua.

Hipertermia de tejido profundo: se utiliza para tratar tumores dentro del cuerpo, tales como cáncer cervical o de vejiga. Para este tipo de hipertermia se utilizan aplicadores externos que están basados en arreglos de radiadores de fase [6]. Los aplicadores se posicionan alrededor de la cavidad del cuerpo u órgano para ser tratada y la energía de microondas o radiofrecuencia es enfocada en el área deseada para ocasionar incremento en la temperatura del tumor [17].

Perfusión regional: esta técnica puede ser usada para tratar cáncer en los brazos y piernas, como por ejemplo el melanoma, o cáncer en algunos órganos, tales como el

hígado o el pulmón. En este procedimiento, se extrae sangre del paciente; ésta se calienta y es bombeada de vuelta dentro de la extremidad u órgano [11].

Perfusión peritoneal continua: es una técnica usada para tratar cáncer dentro de la cavidad peritoneal tales como el cáncer de estómago. Un medicamento anticancerígeno es calentado mediante un dispositivo a través de la cavidad peritoneal [11].

2.4.2.3 Hipertermia de cuerpo completo

Es usada para tratar el cáncer metastásico que se encuentra esparcido por todo el cuerpo. Esto se puede lograr mediante varias técnicas que elevan la temperatura corporal, tales como la conducción térmica, inducción extracorpórea y las técnicas de inducción electromagnética (EM) [11].

2.5 Tipos de aplicadores

Las fuentes de calor utilizadas en hipertermia se dividen principalmente en dos: ondas mecánicas (ultrasonido) y ondas electromagnéticas (radiofrecuencia, microondas e infrarrojo) [9]. Para la aplicación de las ondas, ya sean mecánicas o electromagnéticas, se utilizan distintos tipos de dispositivos denominados aplicadores. Algunos de los aplicadores para energía electromagnética (EM), utilizados en la hipertermia oncológica externa (para tratar tumores a una profundidad < 4 cm) se muestran en la Fig. 2 [18].

Para el tratamiento de tumores profundos (más de 4 cm de profundidad), normalmente se coloca alrededor del paciente una matriz que consta de múltiples antenas dipolo o guías de onda [10]. El calentamiento se concentra en la ubicación del tumor mediante la selección adecuada de los ajustes de amplitud-fase de las antenas individuales [10]. Para evitar el calentamiento superficial en la piel y tejidos superficiales del paciente tratado se utiliza un sistema de enfriamiento con agua, conocido en inglés como *wáter*

bolus o bolus de agua en español [19]. Antes de iniciar un tratamiento de hipertermia es necesario escoger la técnica y aplicador adecuados. Esto depende del tamaño y ubicación del tumor; un método para seleccionar la técnica de hipertermia a utilizar es por medio de la Planeación del Tratamiento de Hipertermia (HTP por sus siglas en inglés, Hyperthermia Treatment Planning) [11].

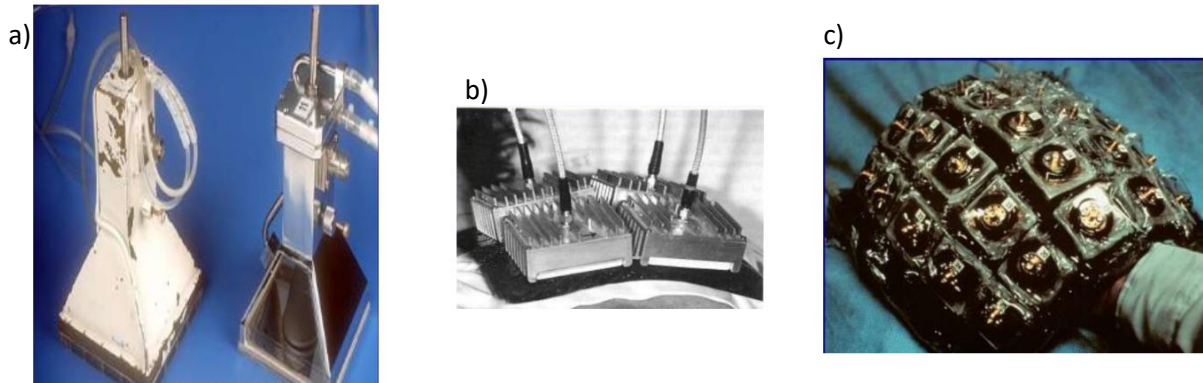


Fig. 2. Aplicadores de energía EM para hipertermia; a) guía de onda (LCA = *Lucite Cone Applicator*); b) aplicador *current sheet*; c) aplicador *microstrip*.

2.6 Planeación de los tratamientos de hipertermia (HTP)

En la planeación del tratamiento de hipertermia se predicen patrones de absorción de energía y distribuciones de temperatura, generados en un paciente al utilizar una configuración específica del o de los aplicadores (guías de onda) [12]. El flujo de trabajo para realizar una planeación del tratamiento paciente-específico de hipertermia se muestra en la Fig. 3.

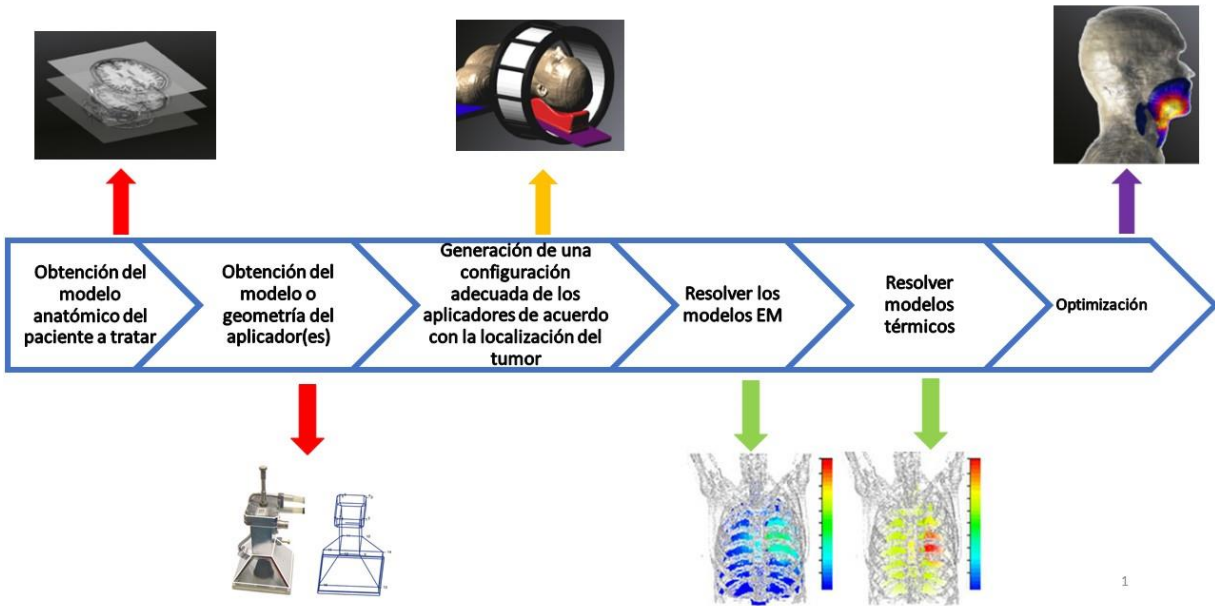


Fig. 3. Flujo de trabajo para la implementación de un sistema de planeación del tratamiento de paciente-específico hipertermia. Inicia con la obtención del modelo anatómico del paciente, después se obtienen los modelos de los aplicadores a utilizar, posteriormente se genera una configuración de los aplicadores de acuerdo con la localización del tumor. Se resuelven los modelos electromagnéticos (EM) y térmicos para finalmente generar una optimización de la ubicación y/o configuración de los aplicadores.

2.6.1 Generación de modelos anatómicos: segmentación de tejidos

La segmentación de los tejidos se realiza sobre las imágenes médicas del paciente a tratar. Normalmente, dichas imágenes se obtienen mediante escaneos por CT (tomografía computarizada) o MRI (Imágenes por resonancia magnética). Es importante que en el estudio de imágenes médicas el paciente mantenga la misma posición que tendrá durante el tratamiento de hipertermia. La segmentación puede ser realizada de forma manual o semiautomática.

2.6.2 Obtención del modelo o geometría del aplicador(es).

Para la hipertermia superficial, el tipo de aplicador usado y su orientación puede afectar sustancialmente el patrón de calentamiento. La planeación del tratamiento es aplicada para investigar las opciones de tratamiento para aplicaciones no estandarizadas de hipertermia superficial, para decidir qué aplicador y que orientación es más efectiva para calentar un objetivo específico [10]. Actualmente, se dispone de distintas técnicas para reconstruir la configuración de un tratamiento de hipertermia superficial; las cuales difieren en costo y facilidad de uso. La reconstrucción a partir de múltiples vistas es uno de los métodos utilizados en la práctica de la radioterapia para la creación y documentación de la posición del paciente durante el tratamiento [20]. Por lo cual, estas técnicas también pueden ser aplicadas para la reconstrucción 3D de los aplicadores a utilizar durante el tratamiento de hipertermia.

2.6.2 Modelado electromagnético (EM)

Después de que el modelo anatómico ha sido creado, el modelo 3D del aplicador se combina con el modelo del 3D del paciente en el software de modelado elegido para predecir el comportamiento de los aplicadores. Es decir, se crea la plantilla que servirá para la implementación de las simulaciones que ayudarán a predecir los resultados del tratamiento. Para realizar el modelado electromagnético es necesario asignarle a cada tejido sus propiedades dieléctricas; de acuerdo con lo reportado en la literatura. Sin embargo, se ha observado que la incertidumbre en las propiedades dieléctricas del tejido produce una inexactitud de ~20% tanto en el SAR (Specific Absorption Rate) como en las predicciones de temperatura [12]. El modelado electromagnético está definido, en este caso, por la fuente de calor externa descrita por la Ecuación (1) [7]. Esta fuente de calor es el campo eléctrico generado por los aplicadores electromagnéticos.

$$Q_{ext} = \frac{1}{2} \sigma_t |E|^2 \quad , \quad (1)$$

donde, σ_t es la conductividad eléctrica del tejido, y E es el campo eléctrico generado por la guía de onda o aplicador. Para obtener la distribución del campo electromagnético en el paciente, es necesario resolver numéricamente las ecuaciones

de Maxwell [12]. Para resolver las ecuaciones de Maxwell en este contexto existen diferentes métodos como, por ejemplo; el método de diferencias finitas en el dominio del tiempo (FDTD), el método del elemento finito (FEM) y la técnica de integración finita (FIT) [12]. Para la generación de simulaciones clínicamente representativas, la posición del paciente con la cual se genera la planeación del tratamiento (HTP) debe corresponder a la posición utilizada durante el tratamiento clínico.

2.6.3 Modelado térmico

El modelado térmico es importante para conocer la distribución de temperatura 3D que se genera en el paciente y que está determinada por los mecanismos de enfriamiento y deposición del SAR como la conducción térmica, la perfusión sanguínea y el enfriamiento causado por el bolus de agua [12]. Para relacionar el modelado electromagnético con el modelado térmico se utiliza comúnmente la Ecuación de bioalentamiento de Pennes [10] mostrada en la Ecuación (2) [7].

$$\rho C \frac{\partial T}{\partial t} + \nabla \cdot (-k \nabla T) = \rho_b C_b \omega_b (T_b - T) + Q_{met} + Q_{ext} \quad , \quad (2)$$

donde, ρ es la densidad del tejido, C es el calor específico del tejido, k es el coeficiente de conductividad térmica, ρ_b es la densidad de la sangre, C_b es el calor específico de la sangre, ω_b es la tasa de perfusión de sanguínea, T_b es la temperatura de la sangre arterial, Q_{met} es la fuente de calor del metabolismo y Q_{ext} es la fuente de calor externa (Ecuación 1).

2.6.4 Técnicas de optimización del tratamiento

La selección adecuada de la amplitud y la fase de los aplicadores es esencial para optimizar el calentamiento del tumor y evitar la generación de los puntos calientes sobre zonas no deseadas. Las técnicas de optimización son muy útiles para este propósito; se han desarrollado varias estrategias de optimización para sistemas de arreglos de aplicadores, que pueden dividirse en métodos de optimización basados en

las distribuciones de SAR y de temperatura [12]. Para obtener una distribución de temperatura 3D completa en el cuerpo humano, la planeación del tratamiento de hipertermia debe tomar en cuenta los aplicadores a utilizar, la anatomía humana, las propiedades (dieléctricas y térmicas) de los tejidos, cambios termo fisiológicos en los tejidos en respuesta al calor y la fisiología, especialmente la perfusión sanguínea [21].

2.7 Reconstrucción 3D

La reconstrucción 3D es el proceso mediante el cual objetos reales son reproducidos en la memoria de una computadora, manteniendo sus características físicas (dimensiones, volumen y forma) [22]. Crear un modelo 3D a partir de un conjunto de imágenes 2D, que sea tan real como sea posible es uno de los problemas fundamentales del modelo basado en imagen y visión por computadora [23].

2.7.1 Campos de aplicación de la reconstrucción 3D a partir de imágenes

La reconstrucción en 3D tiene distintos campos de aplicación tales como: medicina, mapeo robótico, planeamiento urbano, videojuegos, ambientes virtuales, turismo virtual, navegación robótica, arqueología, realidad aumentada, reconocimiento de gestos, etc. [23]. En medicina, la reconstrucción 3D a partir de múltiples imágenes 2D, tomadas con una cámara desde múltiples ángulos, pueden ser usadas para propósitos terapéuticos o de diagnóstico [23].

2.7.2 Métodos de reconstrucción activos y pasivos

Métodos activos. Estos métodos interfieren de forma activa con el objeto para ser reconstruido mediante algoritmos radiométricos o mecánicos. Por ejemplo, un mapa de profundidad se puede reconstruir usando un medidor de profundidad para medir la distancia relativa a un objeto colocado, usando métodos radiométricos a través de fuentes de luz que emiten resplandor hacia el objeto y luego miden la parte reflejada [23].

Métodos pasivos. Los métodos pasivos de reconstrucción 3D no interfieren con el objeto a ser reconstruido. Estos métodos usan un sensor para medir la luminancia reflejada o emitida por la superficie del objeto para deducir su estructura 3D a través del procesamiento de imágenes. Por ejemplo, la adquisición de imágenes con una cámara fotográfica es un método pasivo, ya que el sensor utilizado en la cámara es sensible a la luz visible y como tal el sensor no interfiere con el objeto a reconstruir. Para este caso, los elementos de entrada son un conjunto de imágenes (una, dos o más), por lo cual se trataría de una reconstrucción basada en imágenes [23].

2.7.3 Algoritmos de conversión 2D a 3D

Actualmente, existen distintos métodos para la reconstrucción 3D de objetos mediante imágenes 2D. Los algoritmos de conversión usan tipos específicos de *depth cues* para generar mapas de profundidad dependiendo del número de imágenes de entrada. Los algoritmos de conversión se pueden categorizar en dos grupos: algoritmos basados en dos o más imágenes y algoritmos basados en una sola imagen.

Algoritmos basados en una sola imagen: las *depth cues* de estos algoritmos operan con una sola imagen (la cual puede ser tomada por una cámara común) y se llaman *depth cues* monoculares [24]. Es importante mencionar que la profundidad monocular no se puede utilizar para estimar la profundidad real a menos que se corrijan los valores obtenidos mediante técnicas de aprendizaje automático [23], debido a que las características locales de la imagen son insuficientes para estimar la profundidad, además el enfoque de estos algoritmos está relativamente menos estudiado en la literatura [25].

Algoritmos basados en dos o más imágenes: dos o más imágenes de entrada pueden ser tomadas por múltiples cámaras calibradas ubicadas a diferentes ángulos de visión o por una sola cámara con objetos en movimiento en las escenas. Las *depth cues* de estos algoritmos son llamadas *depth cues* multi-oculares [24]. El objetivo de

estos algoritmos es adquirir distintas imágenes para diferenciarlas en su perspectiva del objeto, buscando cubrir el volumen del objeto de forma adecuada [25]. Algunos de estos algoritmos hacen uso de las secuencias de video como entrada, ya que los algoritmos utilizan la información compartida de fotogramas (*frames*) consecutivos, en forma de información de seguimiento de puntos característicos. Algunos de los factores que podrían afectar estos algoritmos son: el conocimiento o falta de conocimiento de los parámetros de calibración de la cámara y el tener varias cámaras con diferentes ángulos de visión o solo una cámara en movimiento [25].

La Tabla 1 muestra los *depth cues* usadas en los algoritmos de conversión 2D a 3D en función del número de imágenes de entrada.

Tabla 1. Algoritmos de conversión 2D a 3D [23] [24].

Número de imágenes de entrada	<i>Depth cues</i>
Dos o más imágenes	<i>Binocular disparity</i>
	<i>Image blur</i>
	<i>Silhouette</i>
	<i>Structure from motion</i>
Una sola imagen	<i>Linear perspective</i>
	<i>Atmospheres scattering</i>
	<i>Shape from shading</i>

Para realizar una evaluación eficaz de los diferentes algoritmos de conversión 2D a 3D multi-oculares, se realizó una comparación basada en 8 aspectos cualitativos tomados en cuenta por M. Aharchi et al. [23] y Q. Wei [24], cuyos resultados se muestran en la Tabla 2. Los aspectos comparados fueron profundidad absoluta o relativa, rango de profundidad, procesamiento en tiempo real, adquisición de imágenes, contenido de la imagen, presencia de movimiento, mapa de profundidad escaso (*sparse*) o denso (*dense*) y exactitud, los cuales se describen a continuación.

Profundidad absoluta o relativa: algunos de los algoritmos son capaces de proporcionar una distancia real (absoluta) entre la cámara de visualización y los objetos, con lo cual se puede estimar el tamaño real del objeto. Otros algoritmos miden la profundidad relativa analizando el sombreado, bordes, uniones, etc., proporcionando una profundidad relativa pero no los valores reales [23].

Rango de profundidad: este aspecto describe el rango de profundidad efectivo que un ser humano puede percibir basándose en cada señal de profundidad individual [23] [24]. Al considerar el rango de profundidad que puede producir un algoritmo de conversión de 2D a 3D, debe tenerse en cuenta que los rangos de profundidad de la mayoría de los *depth cues* son ilimitados. Depende del usuario elegir la escala de rango de profundidad correcta para que el mapa de profundidad recuperado cree los efectos estéreos más realistas [24].

Procesamiento en tiempo real: algunos de los artículos que abordan la aplicación de los algoritmos de conversión de 2D a 3D proveen tiempos de ejecución explícitos; otros solo indican si algoritmo utilizado es adecuado para una aplicación en tiempo real o no [24]. La velocidad de los algoritmos está relacionada con su precisión; una mayor precisión requiere más tiempo de procesamiento [23].

Adquisición de imágenes: este aspecto describe el sistema de adquisición de imágenes si el método es activo o pasivo [23].

Contenido de la imagen: este aspecto se refiere a qué tipo de características (colores, formato, calidad, tamaño, etc.) son necesarias en las imágenes para que los algoritmos funcionen de manera confiable. En la Tabla 2 a algunos algoritmos se les ha asignado el término “Todos”, lo que indica que no se necesita ningún requisito especial del contenido de la imagen [24].

Presencia de movimiento: este aspecto se refiere a la presencia de disparidad del mismo punto característico en las imágenes de entrada. Este aspecto solo aplica para *depth cues* multi-oculares [23] [24].

Mapa de profundidad escaso o denso: este aspecto trata del nivel de densidad del mapa de profundidad. Algunos algoritmos pueden generar mapas de profundidad densos y dispersos, dependiendo de si el algoritmo utiliza puntos de características locales o estructuras globales [23].

Exactitud: este aspecto se refiere al grado de exactitud que presenta la reconstrucción en función del algoritmo que se use. Por ejemplo, la *binocular disparity* sufre oclusión, es decir, existen puntos que una de las cámaras no puede ver. Mientras que un algoritmo multiocular proporciona resultados más precisos a medida que dispone de más vistas del objeto a reconstruir [24].

De acuerdo con la Tabla 2, se puede observar que el algoritmo de *Structure from Motion* (SfM) es una opción más versátil que otros algoritmos, al tener una alta exactitud, tener un Arango de profundidad amplio (< 30 m). Además, es capaz de manejar métodos de adquisición de imágenes pasivos y activos, no necesita que las imágenes adquiridas tengan algún requerimiento especial y es capaz de generar mapas de profundidad densos y escasos.

Tabla 2. Comparación de los algoritmos de conversión [23][24].

Algoritmo	Profundidad relativa/absoluta	Rango de profundidad	Procesamiento en tiempo real	Adquisición de imágenes	Contenido de la imagen	Presencia de movimiento	Mapa de profundidad escaso o denso	Exactitud
<i>Bonocular disparity</i>	Absoluta	< 30 m	Si	Activa	Todos	Si	Ambos	Porcentaje de pixeles cuyo error de disparidad es superior a 1: 0.16% -0.97% [24]
<i>Image blur</i>	Absoluta	N/A	Si	Activa	Objetos con superficie compleja	No	Denso	Tasa de erro relativo: 0.5% cuando el objeto está a 2.5 m de la cámara
<i>Silhouette</i>	Absoluta	Tamaño interior	Si	Activa	Fondo diferenciable del primer plano	Si	Escaso	El volumen puede ser bien reconstruido pero la textura no es realista.
<i>Structure from motion</i>	Absoluta	< 30 m	Si	Activa/pasiva	Todos	Si	Ambas	Alta precisión lograda cuando se realiza la integración de múltiples <i>frames</i> de secuencias de imágenes

2.7.4 Algoritmo *Structure from Motion* (SfM)

El algoritmo de *Structure from Motion* (SfM) es considerado como un método fotogramétrico automatizado de alta resolución y bajo coste [28]. Esta técnica usa una serie de imágenes bidimensionales de una escena u objeto para reconstruir una estructura tridimensional [23].

Para que el algoritmo de SfM cree una reconstrucción 3D, simplemente se necesitan varias imágenes de un área o un objeto con un alto grado de superposición, tomadas desde diferentes ángulos por una o varias personas en diferentes ubicaciones. No es necesario que la cámara sea especializada, las cámaras estándar de uso común funcionan bien para los algoritmos de SfM [23]. El SfM incremental es un *pipeline* de procesamiento secuencial con un componente iterativo de reconstrucción Fig. 4.



Fig. 4. *Pipeline* del SfM incremental [13].

2.7.5 Etapas de la búsqueda de correspondencia del SfM

La primera etapa en este algoritmo es la búsqueda de correspondencia, la cual encuentra la superposición de las imágenes de entrada $I = \{I_i \mid i = 1 \dots N_I\}$; además de identificar proyecciones de los mismos puntos en las imágenes superpuestas.

Extracción de características: para cada imagen I_i , el SfM detecta conjuntos de características locales $F_i = \{(x_j, f_j) \mid j = 1 \dots N_{F_i}\}$ en una ubicación $x_j \in \mathbb{R}^2$ representadas por un descriptor de apariencia específica f_j . Las características deben

ser invariantes bajo cambios radiométricos y geométricos para que el SfM pueda reconocerlas en otras imágenes. [13]

Emparejamiento: el SfM encuentra imágenes que contienen la misma parte de la escena aprovechando las características F_i como una descripción de la apariencia de las propias imágenes. Se prueba cada par de imágenes para la superposición de escenas; buscando la característica más similar en la Imagen I_a por cada característica en la imagen I_b . La salida de esta etapa es un conjunto de pares de imágenes potencialmente superpuestas $C = \{\{I_a, I_b\} \mid I_a, I_b \in I, a < b\}$ y sus características asociadas $M_{ab} \in F_a \times F_b$. [13]

Verificación geométrica: la tercera etapa verifica los pares de imágenes potencialmente superpuestos C . Dado que las correspondencias del emparejamiento a menudo están contaminadas con valores atípicos, se requieren técnicas de estimación sólidas como RANSAC [13]. La salida de esta etapa es un escenario gráfico con imágenes como nodos y pares de imágenes verificadas como bordes. [13]

2.7.6 Etapas de la reconstrucción incremental del SfM

La entrada para la etapa de la reconstrucción incremental es la escena gráfica, la cual, es proporcionada por la verificación geométrica durante la etapa de la búsqueda de correspondencia. La salida de la reconstrucción incremental son las posiciones estimadas $P = \{P_c \in SE(3) \mid c = 1 \dots N_p\}$ para imágenes registradas y la estructura de la escena reconstruida como un grupo de puntos $X = \{X_k \in \mathbb{R}^3 \mid k = 1 \dots N_x\}$.

Inicialización: SfM inicializa el modelo con una cuidadosa reconstrucción de dos vistas seleccionadas. La robustez, precisión y rendimiento de la reconstrucción depende de la ubicación del inicio del proceso incremental. La inicialización desde una ubicación densa en el gráfico de la imagen con muchas cámaras superpuestas normalmente da como resultado una reconstrucción más sólida y precisa debido a una mayor redundancia. [13]

Registro de imagen: a partir de una reconstrucción métrica, se pueden registrar nuevas imágenes en el modelo actual resolviendo el problema de *Perspective-n-Point* (PnP) utilizando correspondencias de características con puntos triangulados en imágenes ya registradas (correspondencias 2D-3D). El problema PnP involucra estimar la posición P_c y, para cámaras no calibradas, sus parámetros intrínsecos. [13]

Triangulación: una imagen recién registrada debe observar los puntos de escena existentes. Además, también puede aumentar la cobertura de la escena al extender el conjunto de puntos X a través de la triangulación. Se puede triangular un nuevo punto X_k y agregarlo X tan pronto como se registre al menos una imagen más, que también cubra la nueva parte de la escena, pero desde un punto de vista diferente. La triangulación es un paso crucial en SfM, ya que aumenta la estabilidad del modelo existente a través de la redundancia y permite el registro de nuevas imágenes al proporcionar correspondencias 2D-3D adicionales. [13]

Bundle adjustment (BA): el registro de imágenes y la triangulación son procedimientos separados, aunque sus productos están altamente correlacionados: las incertidumbres en la posición de la cámara se propagan a los puntos triangulados y viceversa, y de las triangulaciones adicionales pueden mejorar la posición inicial de la cámara mediante una mayor redundancia. BA es el refinamiento no lineal del conjunto de los parámetros de la cámara P_c y los parámetros de los puntos de la escena X_k que minimiza el error de reproyección usando una función π (Ecuación 3) que proyecta los puntos de la escena al espacio de la imagen y una función de pérdida p_j para reducir potencialmente los valores atípicos. [13]

$$E = \sum_j p_j \left(\left\| \pi (P_c, X_k) - x_j \right\|_2^2 \right) \quad (3)$$

3. Estado del arte

Para este trabajo el estado del arte se agrupó en cuatro módulos diferentes; 1) métodos de reconstrucción 3D, 2) planeación de tratamientos de hipertermia, 3) aquellos documentos que involucren la reconstrucción 3D para un HTP y 4) trabajos que describan la ubicación de un objeto en un espacio 3D.

3.1 Métodos de reconstrucción 3D

La Tabla 3 muestra los trabajos que abordan el tema de la reconstrucción 3D de forma general, también, se muestran trabajos que tiene como enfoque principal el método de SfM para la reconstrucción 3D a partir de una serie de imágenes fotográficas.

Tabla 3. Estado del arte de métodos de reconstrucción 3D

Autor	Año	Título	Resultados
Aharchi, M et al. [23]	2020	<i>A review on 3D reconstruction techniques from 2D images</i>	Aborda conceptos fundamentales de la reconstrucción 3D, los principales problemas que presenta esta técnica, una breve explicación de algunos de los algoritmos de reconstrucción 3D y comparación entre ellos.
T. B. Phan et al. [26]	2019	<i>Dense Optical Flow for the Reconstruction of Weakly Textured and Structured Surfaces: Application to Endoscopy</i>	En este artículo se presenta el método de SfM para reconstruir superficies a partir de imágenes que incluyen texturas y estructuras débiles. Dando como resultado la construcción de superficies complejas de la pared interna del estómago a través de imágenes gastroscópicas.
Özyesil, Onur et al. [27]	2017	<i>A survey of structure from motion</i>	Describe los trabajos tempranos del método de SfM, la estimación de la localización de la cámara, los pasos para realizar llevar acabo esta técnica y los métodos de resolución.

Tomás, R et al. [28]	2016	<i>Structure from motion (SfM): una técnica fotogramétrica de bajo coste para la caracterización y monitoreo de macizos rocosos</i>	Muestra el uso del algoritmo SfM en el ámbito de la mecánica de rocas, concluyendo que este método no tiene un buen funcionamiento cuando se trabaja en zonas densamente vegetadas en comparación con zonas urbanas.
J. L. Schonberger and J. M. Frahm [13]	2016	<i>Structure-from-Motion Revisited</i>	Descripción del método de <i>Structure from Motion</i> , retos y nuevas contribuciones aterrizando en un programa <i>open-source</i> "COLMAP"
Wei, Q et al. [24]	2005	<i>Converting 2D to 3D: a survey</i>	Aborda los algoritmos de reconstrucción 3D de acuerdo con el número de imágenes de entrada en el sistema y se realiza una comparación entre los algoritmos mencionados.

En los trabajos escritos por Wei Q. [22] y Aharchi, M et al. [21] se realizaron revisiones de los algoritmos de reconstrucción 3D. Los cuales se dividieron en función del número de imágenes de entrada (monocular o multi-ocular). Se describió cada algoritmo y se presentaron los trabajos representativos de cada uno de ellos. Posterior a la descripción de cada algoritmo, se realizó la comparación entre ellos basándose en 9 aspectos: la adquisición de la imagen, contenido de la imagen, presencia de movimiento, procesamiento en tiempo real, exactitud, profundidad absoluta o relativa, mapa de profundidad denso o escaso, rango de profundidad y estado de cada *depth cue*. Una de las conclusiones de este trabajo es que las *depth cues* multi-oculares toman en cuenta la información de la imagen espacial como temporal, lo que produce, en general, un resultado más preciso. En ambos trabajos se muestran las ventajas del algoritmo *SfM*, dentro de las cuales destacan su exactitud y su tiempo de procesamiento. De acuerdo con estos antecedentes y con las aplicaciones y mejoras del método de *SfM* en los trabajos de J. L. Schonberger et al [13], T. B. Phan et al [26]

y Thomas R. et al [28], se retoma este método para aplicarlo en la reconstrucción 3D de los elementos involucrados en la emulación de un tratamiento de hipertermia.

3.2 Planeación de un tratamiento de hipertermia

La Tabla 4 muestra un par de trabajos que describen la planeación de los tratamientos de hipertermia, abordando y describiendo los pasos principales involucrados en este proceso.

Tabla 4. Estado del arte de la planeación de un tratamiento de hipertermia

Autor	Año	Título	Resultados
Kok H. P. et al. [10]	2020	<i>Hyperthermia treatment planning: clinical application and ongoing research</i>	Análisis de los pasos fundamentales de una planeación de tratamiento de hipertermia. La posibilidad de mejora en este proceso recae en perfeccionar la fiabilidad cuantitativa de las predicciones de esta planeación apoyándose en los modelos térmicos.
J. Crezee et al. [29]	2020	<i>Loco-regional Hyperthermia Delivery: Patient-specific set-up Procedures for Treatment Optimisation</i>	Este trabajo describe procedimientos de configuración paciente específicos para la optimización de un tratamiento de hipertermia locoregional usando sondas en o cerca de la ubicación del tumor.
Kok H. P. et al. [12]	2015	<i>Current state of art of regional hyperthermia planning: a review</i>	Describe de amplia manera los pasos involucrados en una planeación de tratamiento de hipertermia, su propósito y el software específico para el desarrollo de esta planeación.

En los trabajos previamente mencionados [10][12][29] se describieron los temas de investigación en la planeación del tratamiento de hipertermia. Específicamente en [12] se realiza una revisión del estado del arte de la planeación del tratamiento de hipertermia regional, en el cual se describe el concepto de la planeación del tratamiento de hipertermia, sus aplicaciones y el flujo de trabajo tradicional de este. También se describe cada etapa del flujo de trabajo citando distintos artículos que abordan propuestas de solución para cada etapa, así como las principales dificultades de estas. De acuerdo con estos antecedentes, el presente trabajo se enfoca

específicamente en la generación de una configuración adecuada de los aplicadores de acuerdo con la localización del tumor a ser tratado en el paciente.

3.3 Reconstrucción 3D de aplicadores para la planeación de un tratamiento de hipertermia

La Tabla 5 muestra un trabajo que aplica la reconstrucción 3D para simular la posición de los aplicadores en un tratamiento de hipertermia.

Tabla 5. Estado del arte de la reconstrucción 3D de aplicadores para la planeación de un tratamiento de hipertermia.

Autor	Año	Título	Resultados
Drizdal, T et al. [8]	2012	<i>Reconstruction of applicator positions from multiple-view images for accurate superficial hyperthermia treatment planning</i>	Obtención de un modelo en 3D de la posición de los aplicadores para una para una planeación de un tratamiento de hipertemia superficial.
M. Linthorst, et al. [30]	2011	<i>Procedure for creating a three-dimensional (3D) model for superficial hyperthermia treatment planning</i>	En este artículo se presenta el procedimiento de realizar una tomografía computarizada (CT) y segmentar las imágenes obtenidas para crear un modelo específico del paciente y del tratamiento para la planeación del tratamiento de hipertermia superficial que representa la configuración real del tratamiento del paciente.

En el trabajo desarrollado por T. Drizdal et al. (2012) se obtuvo un modelo 3D de un arreglo de seis aplicadores LCA. El *software* utilizado en aquel trabajo fue *PhotoModeler Scanner*, para la reconstrucción del modelo se tomaron 8 fotos de cada uno, validando cada uno al comparar las dimensiones obtenidas con las dimensiones reales. Tomando este antecedente, en el presente trabajo se retoma la reconstrucción 3D a través de fotografías a través del método *SfM*.

3.4 Localización de objetos en un espacio 3D

La Tabla 6 muestra un trabajo en donde se presenta un método para la localización de tags de ultra alta frecuencia (UHF) RFID con un robot móvil.

Tabla 6. Estado del arte de obtención de la posición de un objeto en un espacio 3D

Autor	Año	Título	Resultados
A. Tzitzis, A. C. Raptopoulos, and T. Yioultis, [31]	2021	<i>A Real-Time Multi-Antenna SAR-based method for 3D localization of RFID Tags by a Moving Robot</i>	Localización 3D de tags de UHF RFID por un robot en movimiento. Logra un error 3D medio menor de 20 cm entre la posición del robot y el tag con un tiempo de .17 s

De este trabajo se puede rescatar la forma en que se determinó la posición del tag en un espacio 3D; ésta fue a través de la distancia euclidiana entre el tag y el receptor. Además, también se relacionó la radiofrecuencia emitida por dicho tag y la recibida por el receptor. Para la realización del presente trabajo, se retoma de [31] el uso de la distancia euclidiana para la determinación de un punto en el espacio 3D, en este caso en específico, los vértices de los aplicadores.

4. Planteamiento del problema

Una de las partes más complicadas durante un tratamiento de hipertermia es la obtención de una distribución de temperatura en 3D. Esto debido a que los perfiles de temperatura obtenidos en los pacientes bajo tratamiento se registran con un número limitado de sensores de temperatura insertados sobre la región tratada. Por lo tanto, sólo se tiene un registro puntual de las temperaturas alcanzadas durante el tratamiento. Es decir, esta forma de registro no es capaz de generar mapas de distribución de temperatura volumétricos. Para resolver este problema es necesario el uso de herramientas digitales, con el propósito de predecir o conocer el efecto térmico de los aplicadores (previamente diseñados) sobre los pacientes. Una forma de predecir los resultados de un tratamiento es mediante el desarrollo de sistemas de planeación de tratamientos en pacientes específicos. Este tipo de sistemas, también sirven como una herramienta de ayuda en la toma de decisiones; apoyando al médico especialista para decidir si un paciente puede o no ser tratado con hipertermia.

Mediante la planeación del tratamiento es posible conocer, antes de su aplicación, la configuración óptima de éste. Es decir, permite conocer el número de aplicadores, posición de los aplicadores, potencias aplicadas, tiempo de tratamiento entre otros parámetros importantes para la aplicación del tratamiento. La planeación del tratamiento debe ser implementada de acuerdo con la anatomía del paciente y la ubicación del tumor a tratar; es decir, debe ser personalizada. Por lo tanto, antes de implementar un tratamiento de hipertermia, e incluso el sistema de planeación de tratamientos, es necesario crear un sistema capaz de reconstruir la posición del o de los aplicadores (guías de onda) con los cuales un paciente específico será tratado. De esta manera, se pretende que en un futuro este sistema sea utilizado en la generación semiautomática de la planeación de tratamientos en paciente específicos, para tratar tumores ubicados en la interfaz músculo-hueso.

5. Objetivo general

Obtener la posición (x , y , z & θ) de cada uno de los aplicadores propuestos en un tratamiento terapéutico de hipertermia, a través de una reconstrucción 3D, a partir de fotografías bidimensionales. Para su futura implementación en un sistema de planeación del tratamiento de tumores óseos.

5.1 Objetivos particulares

- Evaluar de forma preliminar la factibilidad del aplicador o guía de onda rectangular (desarrollada por C. Trujillo *et. al* [11]) para su implementación en el tratamiento de tumores en la interfaz músculo-hueso, esto a través de un modelo multicapa bidimensional usando el método del elemento finito en un *software* de modelado.
- Diseñar con un software CAD un aplicador que emule las dimensiones reales de la guía de onda.
- Realizar la obtención de las coordenadas (x , y , z & θ) de los vértices del diseño CAD del aplicador.
- Diseñar con un software CAD un arreglo de 3 aplicadores que emule un tratamiento de hipertermia.
- Obtener la posición (x , y , z & θ) de los vértices de los aplicadores diseñados en función un marco de referencia arbitrario.
- Exportar las coordenadas obtenidas al software de modelado.
- Construcción de 3 réplicas no funcionales de los aplicadores, siguiendo el diseñado propuesto por C. Trujillo *et. al* [11].
- Montar un sistema experimental con dos aplicadores y un maniquí para emular un tratamiento de hipertermia
- Reconstruir en 3D los elementos de un tratamiento de hipertermia (maniquí y el arreglo de los aplicadores) con el algoritmo *SfM*.
- Obtener la posición (x , y , z & θ) de los vértices de los aplicadores reconstruidos en función un marco de referencia arbitrario.

- Exportar las coordenadas obtenidas al *software* de modelado.

6. Solución propuesta

Como primer paso, para el desarrollo de un sistema de planeación de tratamientos en pacientes específicos, es necesario conocer la posición que los aplicadores (guías de onda) tendrán durante el tratamiento. Es importante mencionar que dicha posición siempre estará en función de la anatomía del paciente y de la ubicación del tumor a tratar. Por lo tanto, es de esperarse que la posición de los aplicadores puede variar tanto en los ejes x , y & z , así como en su ángulo de inclinación (θ). Por lo tanto, en este proyecto se propone realizar la reconstrucción virtual en 3D de la posición de los aplicadores en un tratamiento de hipertermia. Es decir, las guías de onda se posicionarán sobre un paciente (maniquí) con el fin de emular su posición durante un tratamiento real de hipertermia. Posteriormente, mediante la reconstrucción virtual 3D se obtendrá información sobre la posición de los aplicadores (x , y , z & θ) (Ver Fig. 5). Dicha información será utilizada en un futuro en el desarrollo de modelos computacionales basados en el método del elemento finito (sistema de planeación de tratamientos) que ayudarán a conocer el efecto de una terapia térmica sobre un paciente específico. Además, también ayudará a obtener características propias del tratamiento tales como: potencia de aplicación y tiempo de terapia.

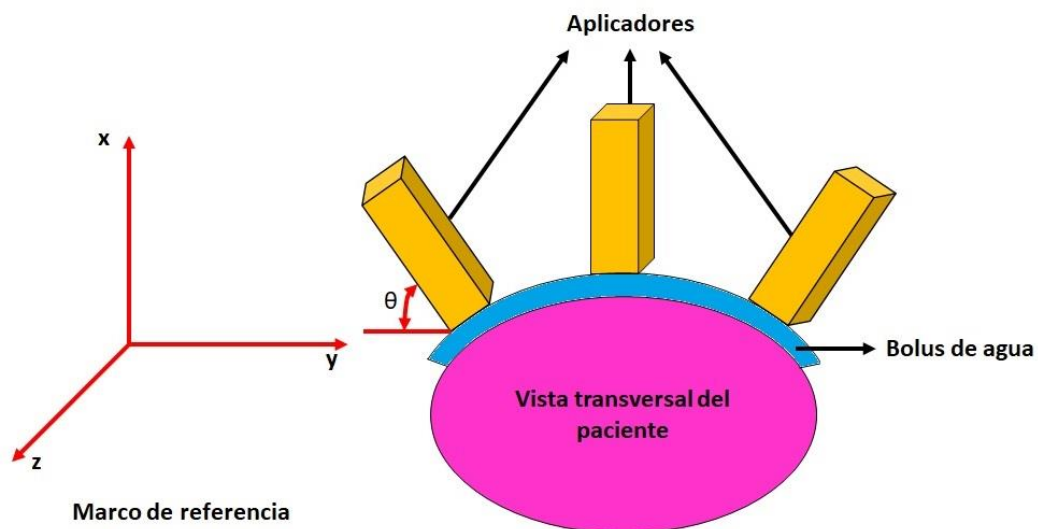


Fig. 5. Sistema propuesto para la obtención de las coordenadas de los aplicadores en un tratamiento de hipertermia.

Para el desarrollo de este proyecto se propone el uso de los aplicadores de RF diseñados por C. Trujillo *et al.* (2011). Como primer paso, se realizó un estudio para evaluar si es posible utilizar estos aplicadores en el tratamiento de tumores óseos localizados en la interfaz músculo-hueso. Dicho estudio se realizó utilizando el método del elemento finito, mediante el *software COMSOL Multiphysics*. Después de corroborar el funcionamiento de dichos aplicadores, se procedió a trabajar en el objetivo general de este trabajo.

Debido a que los protocolos de seguridad derivados de la pandemia del COVID-19 impedían el libre acceso a las instalaciones del CINVESTAV unidad Zacatenco, la solución propuesta, para alcanzar el objetivo general de este proyecto, se desarrolló en 3 etapas; 1) Obtención de las coordenadas del diseño CAD de un aplicador, 2) Obtención de las coordenadas del diseño CAD de tres aplicadores y 3) La reconstrucción 3D de la emulación de un tratamiento de hipertermia.

En la primera etapa (ver Fig. 6), como un primer acercamiento al objetivo general, se realizó el diseño CAD de un solo aplicador para determinar su posición mediante la detección de sus vértices y la obtención de las coordenadas de cada uno de ellos.

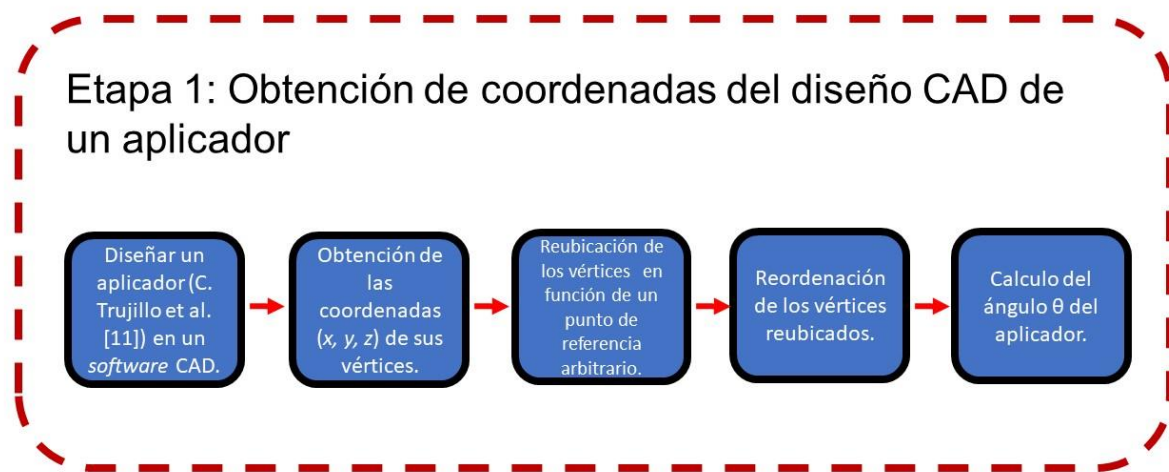


Fig. 6. Flujo de trabajo de la Etapa 1. Inicia con la elaboración un diseño CAD del aplicador propuesto por C. Trujillo et al [11]. Continúa con la obtención las coordenadas del diseño. Después se realiza la reubicación de los vértices en función de un punto de referencia arbitrario. Seguido de la reordenación de estos vértices reubicados para finalmente calcular el ángulo θ .

Posteriormente, se reubicaron los vértices en función de un punto de referencia arbitrario para después ordenar los vértices de una forma específica. Esto con el objetivo de facilitar su futura exportación al *software* de modelado. Como último paso en esta etapa, se calculó el ángulo θ para este aplicador.

En la segunda etapa (ver Fig. 7), se siguió la misma línea de trabajo que en la etapa 1, con la diferencia que el diseño CAD consistió en un arreglo de 3 aplicadores colocados sobre un cilindro con medidas antropométricas, los cuales emulan un tratamiento de hipertermia utilizando un arreglo multi-aplicador. Otra diferencia entre la etapa 2 y la 1, es que en la etapa 2 se realizó la exportación de las coordenadas de los vértices de los aplicadores al *software* de modelado, obteniendo una reconstrucción de los aplicadores en este *software* con los datos (coordenadas) previamente obtenidas.



Fig. 7. Flujo de trabajo de la Etapa 2. En esencia esta etapa es una extensión de la Etapa 1, implementando un arreglo de tres aplicadores y un cilindro con medidas antropométricas además de la adición de la exportación de las coordenadas (x, y, z) de los vértices de los aplicadores al *software* de modelado.

En la tercera etapa (ver Fig. 8), se construyeron de 3 aplicadores no funcionales de madera los cuales se colocaron sobre un maniquí para emular un tratamiento de hipertermia. Una vez ubicados los aplicadores sobre el maniquí se tomó un video de todo el sistema para obtener una serie de imágenes para realizar la reconstrucción 3D con el algoritmo de *SfM*. Después de obtener la reconstrucción 3D del sistema se realizó una edición de ésta para facilitar su procesamiento (detección de vértices, obtención de coordenadas, etc.). Una vez obtenida la reconstrucción editada se siguió el flujo de trabajo de la etapa 2 para culminar con la reconstrucción de los aplicadores en el *software* de modelado.

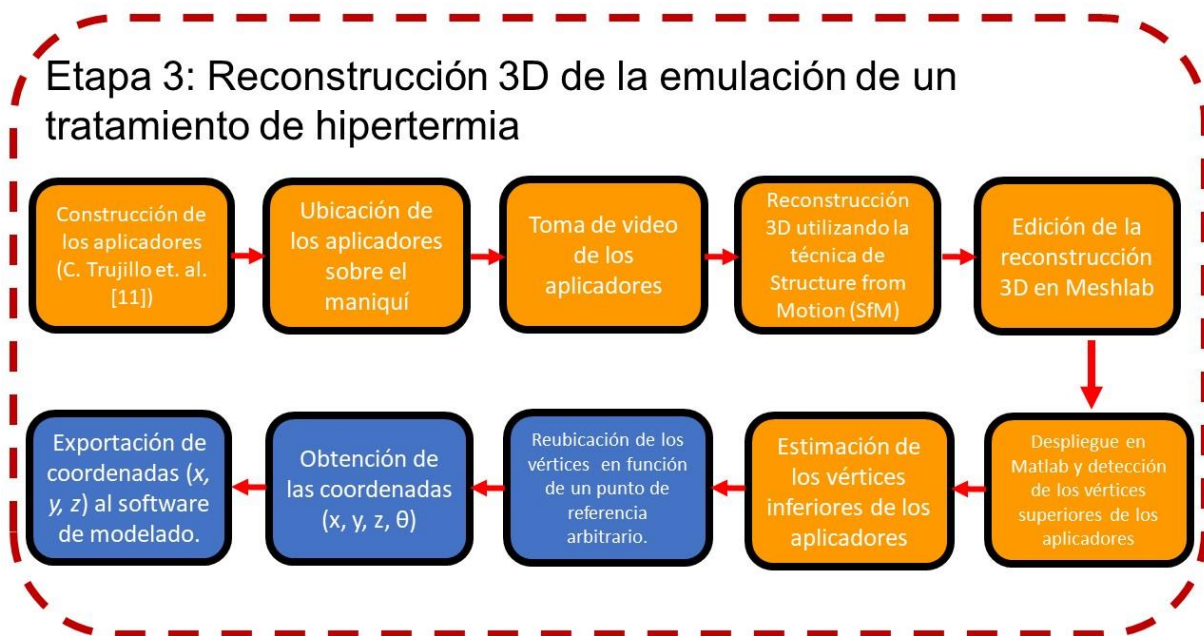


Fig. 8. Flujo de trabajo de la Etapa 3. Este flujo de añade una serie de pasos (recuadros naranjas) para la obtención de la reconstrucción 3D de los elementos involucrados en la emulación de un tratamiento de hipertermia (maniquí y 2 aplicadores) e implementa los pasos realizados en las Etapas 1 y 2 (recuadros azules) para la obtención de las coordenadas de los vértices de los aplicadores.

7. Metodología

7.1 Evaluación de la factibilidad de uso del aplicador “guía de onda” para la implementación del tratamiento de tumores en la interfaz músculo-hueso.

El primer paso en el desarrollo de este proyecto consistió en realizar una evaluación preliminar de la guía de onda rectangular (C. Trujillo *et al* 2011), para comprobar su fiabilidad en el tratamiento de tumores en la interfaz músculo-hueso. Para implementar este estudio, se realizó el modelo 2D de la guía de onda. El aplicador utiliza agua desionizada como medio dieléctrico en su interior cuya permitividad relativa es $\epsilon_r=78.5$. Utilizar este medio dieléctrico al interior del aplicador da como resultado una guía de onda con dimensiones de 84 mm X 42 mm X 169 mm a una frecuencia de funcionamiento de 250 MHz

El estudio se realizó mediante el modelado basado en el método del elemento finito por medio del software *COMSOL-Multiphysics*. La Fig. 9 muestra el modelo 2D, que consiste en la guía de onda, el bolus de agua cuyas funciones son el acoplamiento y evitar sobrecalentamiento en la piel; así como los tejidos que se encuentran entre la guía de onda y la interfaz músculo-hueso. La Tabla 7 muestra los valores de las propiedades dieléctricas y térmicas de los medios utilizados en el modelo. El ancho de los tejidos se consideró del doble de la anchura de la guía de onda; es decir, 168 mm. Estas dimensiones nos ayudaran a observar la distribución de temperatura lograda en los tejidos. Por otro lado, los espesores de los medios utilizados se muestran en la Tabla 8.

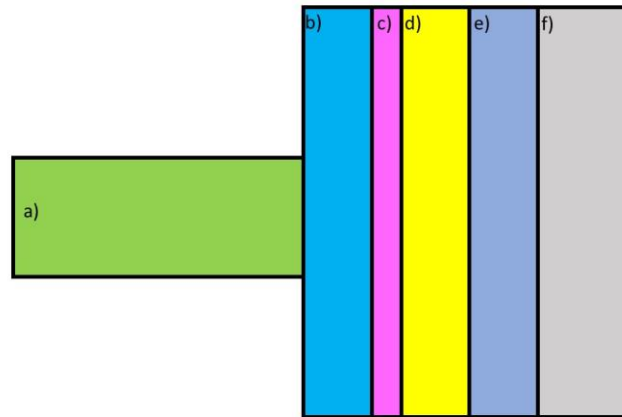


Fig. 9. Modelo 2D propuesto para evaluar la efectividad de la guía de onda en el tratamiento de tumores en la interfaz músculo-hueso. a) guía de onda, b) bolus de agua, c) piel, d) grasa, e) músculo y f) hueso

Tabla 7. Propiedades térmicas y dieléctricas de los medios implicados en el modelo 2D a una frecuencia de 250 MHz [32] [33] [34] [35] [36] [37].

Medio	Agua desionizada (guía de onda)	Agua destilada (bolus de agua)	Piel	Grasa	Músculo	Hueso	Sangre
Permitividad relativa (-)	78.5	81	52.2	11.8	59	13.1	-
Conductividad eléctrica (S/m)	5.5×10^{-8}	0.0002	0.614	0.0745	0.757	0.0784	-
Calor específico (J/ (kg K))	-	-	3391	2348	3421	1313	4180
Tasa de perfusión sanguínea (1/s)	-	-	0.02	4.58×10^{-4}	12.5×10^{-4}	4.36×10^{-4}	-
Densidad (kg/m ³)	-	-	1125	916	1000	1990	1000
Coeficiente de transferencia de calor (W/m ² /K)	-	65	-	-	-	-	-
Conductividad térmica (W/m*K)	-	-	0.37	0.21	0.5	0.32	-

Tabla 8. Espesores de los medios utilizados en el estudio paramétrico [7][34][35][38][39].

Medio	Espesor (mm)	Temperatura (°C)
Bolus de aguda (agua destilada)	5 & 20	37 & 42
Piel	2	37
Grasa	1:1:30	37
Músculo	1:1:50	37
Hueso	50	37

7.1.1 Estudio paramétrico

Se implementó un estudio paramétrico con el fin de evaluar la viabilidad de tratar tumores cercanos a la interfaz músculo-hueso con la guía de onda rectangular propuesta. Se evaluó el efecto sobre la profundidad de penetración de acuerdo con ciertos parámetros tales como la temperatura del bolus del agua, el espesor del bolus de agua, los espesores de los tejidos (grasa y músculo) y el tiempo de tratamiento. Los tiempos de tratamiento analizados se eligieron de acuerdo con lo reportado en la literatura; *i. e.* se analizaron tiempos de 60 min y 90 min. La Tabla 8 muestra los parámetros implementados en el estudio paramétrico, así como su variación.

La primera simulación se hizo variando el espesor de la grasa y el espesor del músculo que se muestra en la Tabla 8. El tiempo de simulación fue de aprox. 6 horas en un ordenador con una memoria RAM de 8 GB, un procesador Core i5 novena generación y un sistema operativo de 64 bits. Para las simulaciones siguientes sólo se tomaron algunos valores de los espesores de la grasa y el músculo, para reducir el tiempo de simulación (aprox. 3 horas) y obtener una muestra representativa sin perder información importante. Los valores de espesor utilizados para la grasa fueron de 1 mm, 15 mm y 30 mm. En el caso del músculo los espesores fueron de 1 mm, 9 mm, 17 mm, 25 mm, 33 mm y 50 mm. La malla utilizada fue generada con un tamaño de elemento mínimo de 6.22×10^{-6} m y un tamaño de elemento máximo de 0.00311 m, estos valores se obtuvieron con ayuda de los tamaños predeterminados del *software*. Al cambiar el tamaño de los elementos a dimensiones mayores el modelo no tuvo

convergencia. La forma de la malla fue *free tetrahedral* y consistió en un total de 4166 elementos cuando los espesores tuvieron su mayor valor.

7.1.2 Modelo electromagnético (EM)

El modelo EM se define por la fuente de calor externa (Q_{ext}) descrita por la Ecuación (4)[7]:

$$Q_{ext} = \frac{1}{2} \sigma_t |E|^2 \quad , \quad (4)$$

donde, σ_t es la conductividad eléctrica del tejido, y E es el campo eléctrico generado por la guía de onda. Las paredes de la guía de onda se consideraron como conductores eléctricos perfectos. Esto significa que la componente tangencial del campo eléctrico es igual a cero en los límites. La guía de onda se excitó por una onda transversal eléctrica (TE_{10}), que no tiene componentes de campo eléctrico en la dirección de propagación. Mientras que los contornos exteriores de los tejidos se configuraron de tal manera que la propagación de las ondas provenientes de la guía de onda fue absorbida por los bordes.

7.1.3 Modelo térmico

Para relacionar la onda electromagnética generada por la guía de onda rectangular con el aumento de temperatura en el tejido, se utilizó la Ecuación de biocalentamiento (5) [7]

$$\rho C \frac{\partial T}{\partial t} + \nabla \cdot (-k \nabla T) = \rho_b C_b \omega_b (T_b - T) + Q_{met} + Q_{ext} \quad , \quad (5)$$

donde, ρ es la densidad tisular, C es el calor específico del tejido, k es la conductividad térmica del tejido, ρ_b es la densidad de la sangre, C_b es el calor específico de la sangre,

ω_b es la tasa de perfusión de sangre, T_b es la temperatura de la sangre arterial, Q_{met} es la fuente de calor del metabolismo y Q_{ext} es la fuente de calor externa.

7.1.4 Parámetros de evaluación

Con el fin de evaluar la eficiencia en la transmisión de energía se analizó la relación de onda estacionaria (SWR, por sus siglas en inglés Standing Wave Ratio). El SWR se define como la relación entre la cantidad de energía emitida por la fuente y la cantidad de energía reflejada de vuelta por el medio en donde se transmite y es descrita por la Ecuación (6)[7]:

$$SWR = \frac{1+\Gamma}{1-\Gamma} \quad , \quad (6)$$

donde Γ es el coeficiente de reflexión. El valor SWR ideal es 1; lo que indica que toda la potencia entregada por el aplicador es transmitida al tejido. Por lo tanto, un SWR alto representa una mayor pérdida de energía en comparación con un SWR cercano al 1 (valor ideal) [7]. En otras palabras, cuanto mayor sea el SWR, mayor será la pérdida en la energía transferida. El SWR se utilizó para evaluar la eficiencia del sistema mediante la evaluación de la energía transmitida. Con el fin de conocer el efecto del bolus de agua en el campo eléctrico y cómo interactúa con los tejidos se analizaron las distribuciones de campo eléctrico. Además, obtuvieron las distribuciones de temperatura en los tejidos para conocer el efecto de los espesores de los tejidos y del bolus de agua. También se obtuvieron las temperaturas máximas alcanzadas y la profundidad de penetración.

7.2 Etapa 1: Obtención de las coordenadas del diseño CAD de un aplicador

Después de corroborar el funcionamiento de los aplicadores en el tratamiento de tumores óseos localizados en la interfaz músculo-hueso con el método del elemento finito a través del *software COMSOL Multiphysics*, el siguiente paso consistió desarrollar la Etapa 1 (ver Fig. 6). Es decir, a partir del diseño CAD de un aplicador se

obtuvieron las coordenadas x , y , z & θ de cada uno de sus vértices Fig. 6; los cuales describen su posición.

7.2.1 Diseño CAD de un aplicador

Se realizó el diseño de una guía de onda ideal en un *software* CAD. Se consideró que este diseño es “ideal” porque sus dimensiones exteriores son exactamente las mismas que las medidas presentes en los diseños originales (ver Fig. 10); además, debido a la omisión de algunos detalles tales como, la conexión al sistema de RF (entrada coaxial), los tornillos presentes en las pestañas del aplicador y la apertura de la guía de onda en uno de sus extremos. Es importante mencionar que este diseño se guardó en formato .stl para su despliegue y procesamiento en el *software* *Matlab*.



7.2.2 Obtención de las coordenadas (x , y , z & θ) de los vértices del aplicador

Después de obtener el diseño CAD de un solo aplicador en formato .stl, se desarrolló un algoritmo llamado *ext_coor_apli.mat* (Apendice I) en el *software Matlab*. En la Fig. 11 se muestra el flujo de trabajo seguido para la implementación de dicho algoritmo, cuyos bloques se describen a continuación.

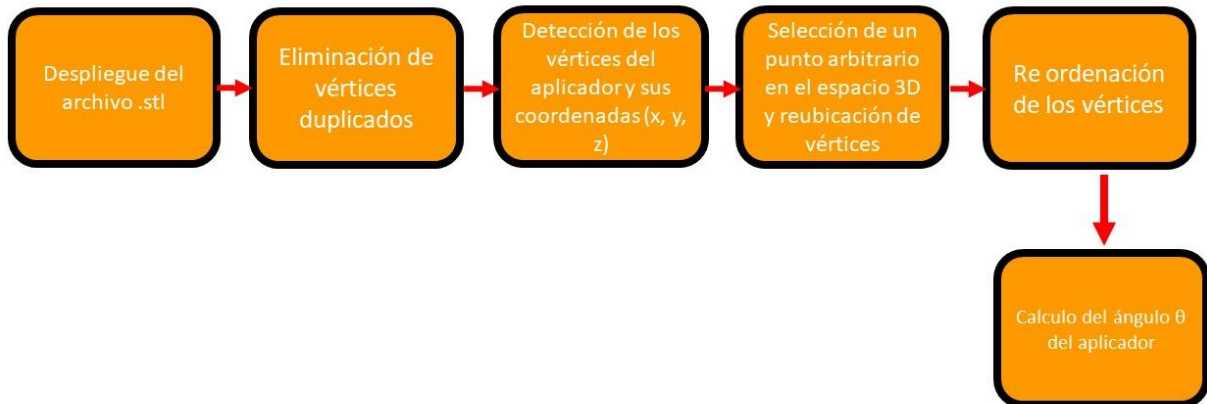


Fig. 11. Diagrama de flujo para la obtención de las coordenadas del diseño CAD de un aplicador.

7.2.2.1 Fases del diagrama de flujo para la obtención de las coordenadas del diseño CAD de un aplicador

Despliegue del archivo .stl y eliminación de vértices duplicados: en el programa *ext_coor_apli.mat* se utilizó una función llamada *stlread* [40], esta función es capaz de desplegar archivos .stl dentro de *Matlab*, además de almacenar en un tipo de dato llamado *structure* las variables *faces* (caras) y *vertices* (vértices) del archivo desplegado. Una vez que el programa ejecuta el despliegue del archivo .stl, el siguiente paso fue eliminar todos los vértices duplicados con la función *unique*, la cual se aplica a la variable *vertices*.

Detección de los vértices del aplicador: una vez que han sido eliminados los vértices duplicados, se realizó la detección de aquellos vértices (y sus respectivas coordenadas x , y , z) que tienen una distancia entre sí que coincide con alguna de las dimensiones exteriores de los aplicadores. En la Fig. 12 se muestran los vértices de una guía de onda ideal. En la Tabla 9 se muestran las distancias entre el vértice A y el resto de los vértices del diseño CAD del aplicador.

ext_coor_apli. Sin embargo, al trabajar con modelos ideales se usó una tolerancia de 5×10^{-3} mm.

Selección de un punto arbitrario en el espacio 3D y reubicación de los vértices: una vez que los vértices fueron detectados (8 vértices por tratarse de un solo aplicador) por el programa, éste solicita al usuario dar clic en algún punto en el espacio 3D desplegado. Los vértices del aplicador se reubican en función de este punto realizando la operación descrita por la Ecuación 8.

$$R = C_a - C_r , \quad (8)$$

donde R es la nueva ubicación de las coordenadas de los vértices, C_a son las coordenadas (x, y, z) de cada uno de los vértices del aplicador y C_r son las coordenadas (x, y, z) del punto seleccionado. Por esta razón el aplicador desplegado se renombra como el aplicador reubicado.

Reordenación de los vértices: después de colocar los vértices en función del nuevo origen es necesario que estos tengan un orden específico para que sean ingresados en el *software* de elemento finito, utilizado para predecir el comportamiento de los aplicadores. Por lo cual, se realizó la suma del valor de la coordenada x, más el valor de la coordenada y, más el valor de la coordenada z, de cada uno de los vértices. El vértice cuya suma tenga el menor valor se denomina el vértice A. Un ejemplo de este proceso se muestra en la Fig. 13. Una vez que se ha identificado el vértice A, los vértices restantes (B, C, D, E, F, G y H) se determinan calculando la distancia (Ecuación 7) entre el vértice A y el resto de los vértices reubicados. Como resultado final se obtienen los vértices (y sus respectivas coordenadas) ordenados de la siguiente forma; vértice A, B, C, D, F, E, G y H.

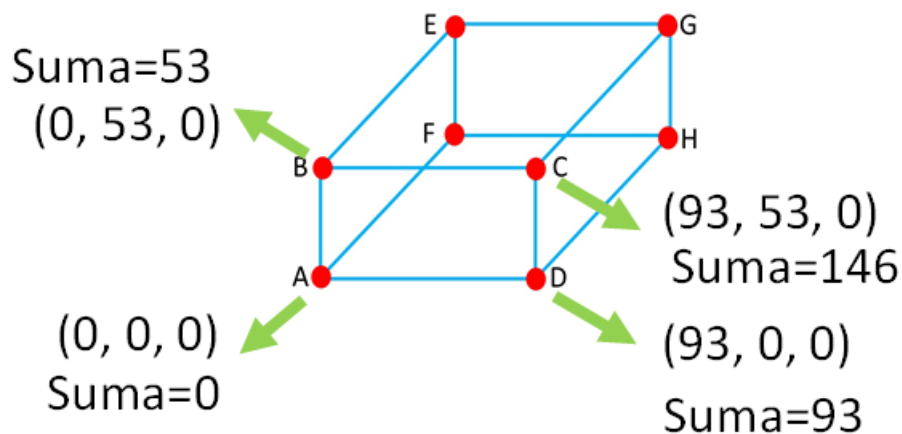


Fig. 13. Ejemplo de la suma de las coordenadas de los vértices para su correcto acomodo.

Cálculo del ángulo θ del aplicador: una vez ordenados los vértices y sus respectivas coordenadas, el siguiente paso fue determinar el ángulo la posición del aplicador. Para calcular el ángulo, se propuso un aplicador de referencia que tenga una posición básica, aunque solo se toman las distancias primordiales (distancia entre los vértices AB, AD y AF, ver Fig. 13). Tomando como referencia el vértice A del aplicador reubicado se suma la distancia AB al eje z de este vértice para generar un vértice del aplicador de referencia. Para generar el siguiente vértice se suma la distancia AD al eje x y para generar el último vértice se suma la distancia AF al eje y. Por lo tanto, el aplicador de referencia coincide con el aplicador reubicado en el vértice A tal y como se muestra en la Fig. 14.

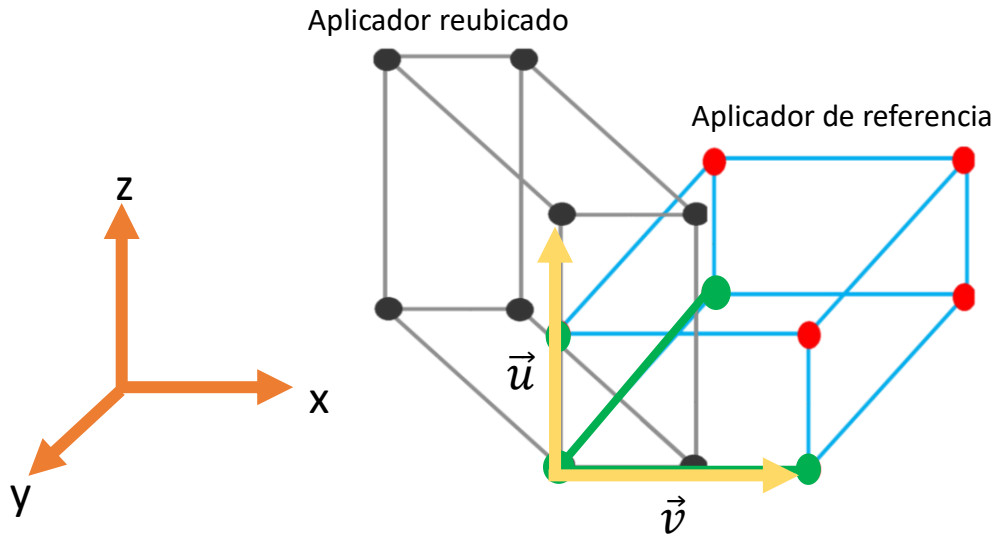


Fig. 14. Aplicador de referencia (azul y rojo), aplicador desplegado reubicado (gris y negro), distancias de referencia (verde).

Dada esta propuesta en la que se generan vértices de referencia, se calculan los ángulos que existen entre las distancias AB, AD y AF de ambos aplicadores (el de referencia y el reubicado). También fue necesario calcular los ángulos de cada plano (xy, xz & yz) para cada uno de los vértices. Para calcular los ángulos correspondientes, se utilizó la Ecuación 9;

$$\theta = \arccos\left(\frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \cdot |\vec{v}|}\right) \quad , \quad (9)$$

donde \vec{v} es el vector de referencia, \vec{u} el vector perteneciente al aplicador reubicado. Sin embargo, para el uso de esta Ecuación es necesario que las distancias funcionen como vectores, esto es, tomando el vértice A para ambos aplicadores como el origen y dibujando el vector desde ese punto hasta el vértice correspondiente para generar el vector con magnitud AB, AD o AF tal y como se muestra en la Fig. 14, para este caso, el vector \vec{v} es el vector perteneciente al aplicador de referencia con magnitud AD y el vector \vec{u} es el perteneciente al aplicador reubicado también con magnitud AD.

7.3 Etapa 2: Obtención de las coordenadas del diseño CAD de tres aplicadores

Después de contar con la metodología adecuada para obtener la posición (coordenadas x , y , z & θ) de un aplicador diseñado en un *software* CAD, se procedió a la extrapolación de dicha metodología para obtener la posición de un arreglo de aplicadores. En la Etapa 2 (ver Fig. 7) se diseñó un modelo conformado por 3 aplicadores ubicados alrededor de un cilindro para emular un tratamiento de hipertermia dado sobre un miembro inferior del paciente.

7.3.1 Diseño CAD de un arreglo de 3 aplicadores

Los aplicadores diseñados en un *software* CAD (ver Fig. 15) se colocaron con una separación entre sí de 30 mm: dos de los aplicadores se ubicaron en la misma orientación mientras que el tercero (colocado en medio) se encuentra rotado 90° con respecto a los otros dos aplicadores. Todos ellos tienen una separación de 10 mm (la cual es una media del espesor de un bolus de agua, elemento no considerado en este proyecto) con respecto al cilindro. El cilindro tiene medidas antropométricas; es decir, se consideró un diámetro de 116 cm. El formato de este diseño también es .stl al igual que el diseño CAD de un solo aplicador.

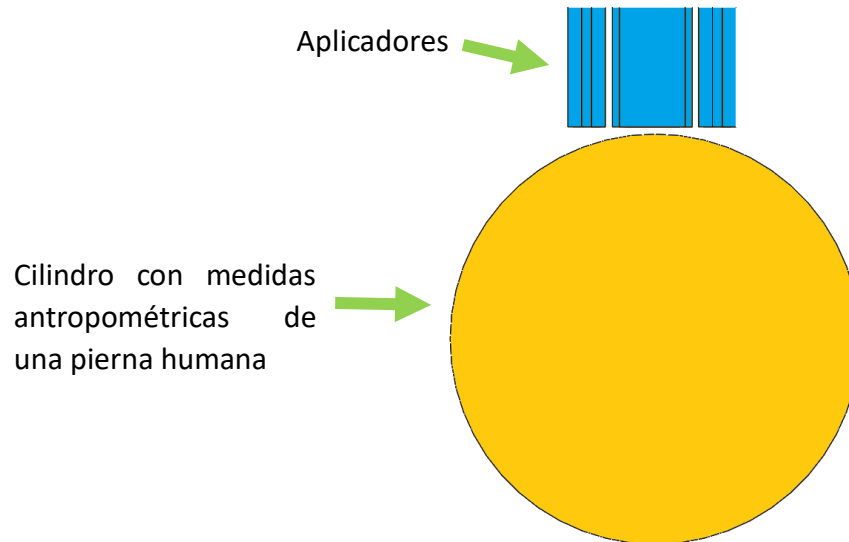


Fig. 15. Diseño CAD del arreglo de 3 aplicadores colocados sobre un cilindro con medidas antropométricas.

7.3.2 Obtención de las coordenadas (x , y , z & θ) de los aplicadores

Para conocer la ubicación de los 3 aplicadores (x , y , z & θ), se siguió la metodología mostrada en la Fig. 11. Por lo tanto, primero se desplegó el diseño CAD del arreglo de aplicadores, después se eliminaron los vértices duplicados y posteriormente se realizó la detección de aquellos vértices que pertenecen a los aplicadores, midiendo la distancia entre ellos con la Ecuación 7. En este caso, como se trabajó con 3 aplicadores, el número de vértices detectados fue de 24 (8 vértices por cada aplicador). Posterior a la identificación de los vértices, el programa *ext_coor_apli.mat* solicita al usuario dar clic en algún punto en el espacio del modelo desplegado para reubicar los vértices de los aplicadores en función de este nuevo punto (Ecuación 8).

Una vez que los vértices han sido reubicados, el siguiente paso fue ordenar los vértices para su fácil exportación al *software* de modelado basado en el método del elemento finito. Para esto, se realizó la suma del valor de la coordenada x , más el valor de la coordenada y , más el valor de la coordenada z , de cada uno de los vértices. El vértice cuya suma tenga el menor valor se denomina el vértice A (ver Fig. 13). Una vez que

se ha identificado el vértice A, los vértices restantes (B, C, D, E, F, G y H) se determinan calculando la distancia (Ecuación 7) entre el vértice A y el resto de los vértices reubicados. Como resultado final se obtienen los vértices (y sus respectivas coordenadas) ordenados de la siguiente forma; vértice A, B, C, D, F, E, G y H.

Después de que los vértices se reordenan, se calcularon los ángulos de los aplicadores en función de los aplicadores de referencia. Es importante mencionar que se utilizó un aplicador de referencia por cada aplicador reubicado, ya que cada par de aplicadores deben coincidir con el vértice A (ver Fig. 14).

7.3.3 Exportación de las coordenadas (x, y, z) de los vértices de los aplicadores al software de modelado

El programa *ext_coor_apli.mat* tiene la capacidad de guardar en una estructura la información relevante; *i. e.* las coordenadas de cada vértice de cada aplicador y el ángulo de cada aplicador en cada uno de los planos. Aun cuando se obtuvieron ambos datos, coordenadas y ángulos de los aplicadores; es importante mencionar que, para la reconstrucción de la posición de los aplicadores en el *software* del elemento finito, únicamente es necesario introducir las coordenadas de los vértices. Es decir, se utilizó la opción de generar un hexaedro; por lo tanto, los vértices del 1 al 4 corresponden a la cara superior del hexaedro en el sentido horario (A, B, C y D); mientras que los vértices del 5 al 8 corresponden a la cara inferior. Es importante mencionar que los vértices se ingresan también en el sentido horario (F, E, G y H), en la Fig. 16 se muestran los vértices del aplicador y la pestaña de ajustes del hexaedro en el *software* del elemento finito.

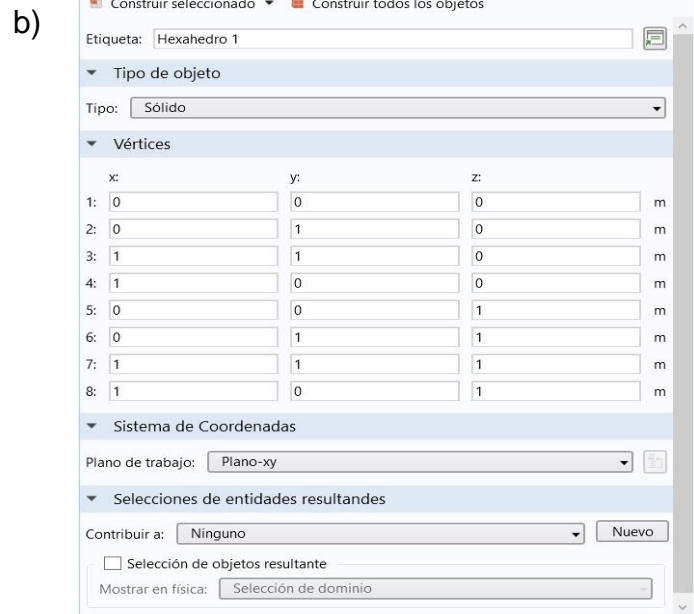
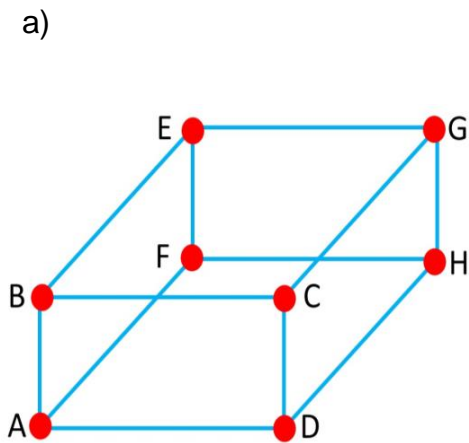


Fig. 16. Exportación de las coordenadas de los vértices de los aplicadores al software de modelado, a) vértices de la geometría del aplicador, b) Pestaña de ajustes del hexaedro en el software del elemento finito.

7.4 Etapa 3: Reconstrucción 3D de la emulación de un tratamiento de hipertermia

Una vez que se obtuvieron las coordenadas de los vértices de los aplicadores y su posterior reconstrucción en el *software* de modelado, como etapa final de este proyecto, se implementaron los ajustes necesarios a la metodología ya planteada para ahora reconstruir la posición de un arreglo de aplicadores utilizados en un tratamiento real de hipertermia (ver Fig. 8). Esto con el objetivo de utilizar dicha información para realizar una reconstrucción 3D y ser capaces de modelar el comportamiento de dichos aplicadores bajo condiciones muy específicas; como lo puede ser la posición del paciente y la zona del tratamiento. La Fig. 17 muestra el sistema montado sobre el maniquí (paciente) y la reconstrucción 3D del mismo.

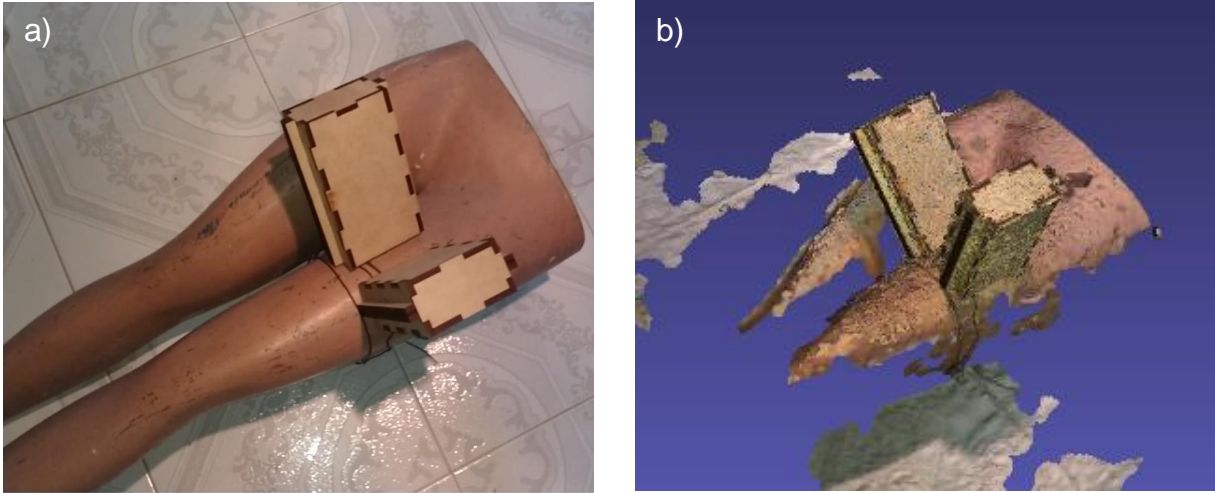


Fig. 17. Reconstrucción 3D de un tratamiento de hipertermia, a) aplicadores (guías de onda) montados sobre el maniquí (paciente), b) reconstrucción 3D de los aplicadores y el maniquí utilizando el algoritmo de *SfM*.

7.4.1 Construcción de los aplicadores propuestos [11]

Debido a la situación de orden mundial producida por la pandemia de COVID-19 y dado que no se tenía acceso al laboratorio para trabajar con los aplicadores ya construidos, fue necesario construir replicas no funcionales de los aplicadores diseñados por C. Trujillo et al. en el año 2011 [11]. El diseño fue proporcionado por el Laboratorio de Radiaciones Electromagnéticas y Ultrasónicas (LAREMUS) CINVESTAV Zacatenco; sin embargo, fue necesario adecuar este diseño para su fabricación con MDF (*Medium Density Fibreboard*) de 6 mm y una cortadora láser. Los aplicadores obtenidos cuentan con las medidas exteriores idénticas a las guías de onda construidas por [11]. La Fig. 18 muestra el diseño CAD utilizado para manufacturar las piezas de aplicadores en MDF utilizadas para su construcción.

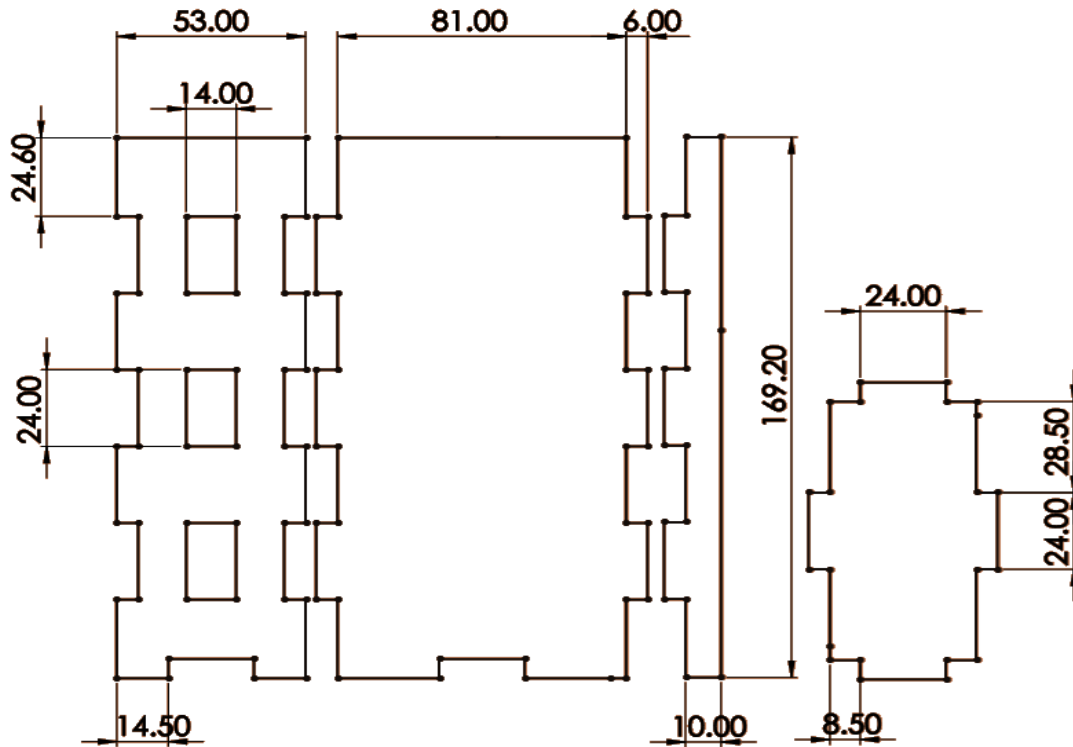


Fig. 18. Planos del diseño CAD de los aplicadores utilizados para su manufactura en una cortadora láser.

7.4.2 Ubicación de los aplicadores sobre el maniquí

De acuerdo con lo descrito previamente, para emular un tratamiento de hipertermia real, es necesario contar, por un lado, con los aplicadores y por otro con un individuo que juegue el papel de paciente. Por lo tanto, en esta fase del proyecto se utilizaron los aplicadores manufacturados en MDF y como paciente se utilizó un maniquí. Por lo tanto, los aplicadores contruidos se colocaron sobre el maniquí. Específicamente, se emuló un tratamiento sobre el miembro inferior de una persona recostada boca arriba (ver Fig. 19). Como un primer acercamiento se colocaron dos aplicadores sobre el maniquí en la zona de la pierna. Ambos aplicadores se colocaron a una distancia de 3 cm entre sí (ver Fig. 17a).



Fig. 19. Maniquí utilizado para emular a un paciente y colocar los aplicadores para emular un tratamiento de hipertermia.

7.4.3 Toma de video de los elementos de un tratamiento de hipertermia

Para realizar una reconstrucción 3D de un ambiente u objeto, es necesario contar con un conjunto de fotografías o bien con un video que lo muestre desde diferentes ángulos. Por lo tanto, después de colocar los aplicadores sobre el maniquí se realizó la grabación de un video de todo el montaje (elementos que emulan el tratamiento de hipertermia). La grabación de este video se llevó a cabo para obtener un conjunto de fotografías que serán utilizadas para la reconstrucción 3D; la cual se implementó con el algoritmo de *SfM* a través del *software COLMAP* [13]. Para la grabación del video se utilizó la cámara integrada al celular Motorola G7 plus, cuyas características se muestran en la Tabla 10.

Tabla 10. Propiedades generales de la cámara integrada en el celular motorola G7 plus [41].

Característica	Valor real
Cámara principal	16 MP
Apertura	f/1.7

Tamaño de pixel	1.22 μm
Tamaño de la fotografía (en pixeles)	1546 X 3264

El video fue capturado sin usar el autoenfoco, sin hacer zoom y utilizando solo la cámara principal, en la Tabla 11 se muestran las configuraciones específicas para la grabación del video.

Tabla 11. Configuración para la grabación del video.

Característica	Valor propuesto
Tamaño de fotografía	10.1 MP
Tamaño de video	60 fps
Distancia de grabación	50 cm – 70 cm

El tiempo de grabación depende de los objetos a reconstruir; es decir, depende del tiempo necesario para grabar todos los objetos. Es importante cubrir todos los ángulos posibles para obtener suficientes imágenes y obtener una reconstrucción 3D más fiel. Para este caso en específico (maniquí y dos aplicadores), los tiempos de grabación oscilaron entre los 30 s – 70 s dando como resultado entre 1,800 - 4,200 *frames* por video. Esta cantidad de *frames* es excesivo para el programa COLMAP [13]. Durante las pruebas preliminares, se encontró un tiempo de procesamiento de más de 3 horas, en un ordenador con una memoria RAM de 8 GB, un procesador Core i5 novena generación y un sistema operativo de 64 bits; sin obtener resultados realmente satisfactorios. Para agilizar el proceso de reconstrucción 3D se tomó un número determinado de los *frames* totales; por lo tanto, se propone la Ecuación 10 para determinar el número de *frames* a utilizar para la reconstrucción 3D.

$$No. \text{ frames} = \text{duración del video} * fps_{deseado} \quad (10)$$

donde $fps_{deseado}$ es el número de imágenes por segundo, este número debe ser menor a 60 que es el número original de fps. Para obtener las imágenes (*frames*) del video grabado se utilizó el *Freeware Free Video to JPG Converter*. Este *software* es capaz de extraer los *frames* de cualquier video dependiendo ya sea por tiempo, por fotogramas o bien asignando un número específico de imágenes. La Fig. 20 muestra la interfaz de este *software*.

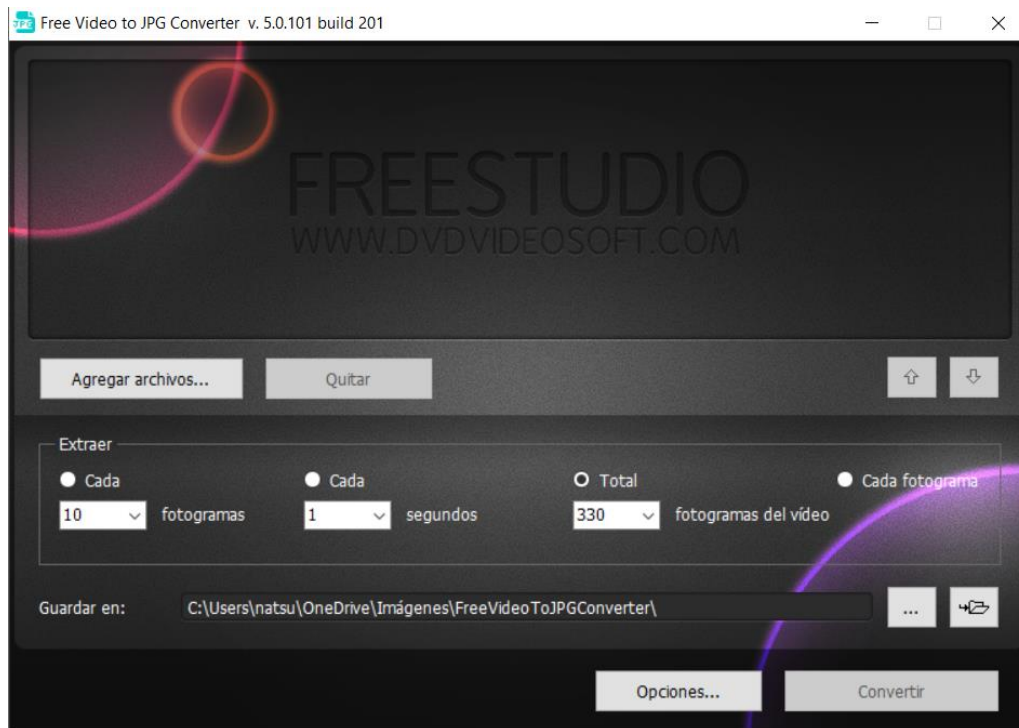


Fig. 20. Interfaz del programa *Freeware Free Video to JPG Converter* utilizado para extraer las imágenes (*frames*) del video del maniquí y los aplicadores.

7.4.4 Reconstrucción 3D utilizando el algoritmo *Structure from Motion (SfM)* implementado en el *software COLMAP* [13]

Los *frames* o imágenes obtenidas del video grabado de los elementos que emulan un tratamiento de hipertermia se utilizaron para generar una reconstrucción virtual 3D en *COLMAP* [13]. Este es un *software open-source* desarrollado por J. L. Schonberger *et*

al. en el año 2016 y está diseñado para realizar reconstrucciones 3D a partir de una serie de imágenes de un objeto con el algoritmo de *Structure from Motion* (SfM). Dentro de las opciones que presenta dicho *software* se encuentra la *Automatic reconstruction*, esta opción permite al usuario realizar una reconstrucción 3D automática de una serie de imágenes de forma automática al realizar una configuración específica de las carpetas de trabajo, calidad de la reconstrucción (baja, media, alta o extrema) y tipo el tipo de imágenes de entrada (imágenes de internet, imágenes individuales o *frames* de video).

En la Fig. 21 se muestra la ventana que se despliega en *COLMAP* al seleccionar la opción de *Automatic reconstruction*. Los parámetros a configurar son: *Workspace folder*, *Image folder*, *Vocabulary tree*, *Data type*, *Quality*, *Shared intrinsics*, *Sparse model*, *Dense model*, *Meshes*, *num_threads*, GPU y *gpu_index*. Sin embargo, los parámetros que se configuraron para este trabajo son solo 4; *Workspace folder*, *Image folder*, *Data type* y *Quality* Esto debido a que los demás parámetros tienen una configuración predeterminada por *COLMAP* y se obtuvieron resultados satisfactorios al trabajar con estas configuraciones.

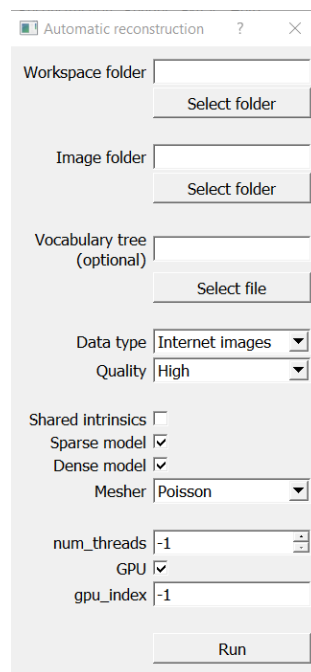


Fig. 21. Ventana de configuraciones de la opción de reconstrucción automática en *COLMAP* [13].

Workspace Folder: esta opción permite al usuario seleccionar la carpeta de trabajo, que es el lugar donde se guardaran los archivos obtenidos de la reconstrucción obtenida, que son el modelo denso (*dense model*) y modelo escaso (*sparse model*).

Image folder: esta opción permite seleccionar la carpeta donde se encuentran las imágenes del objeto a reconstruir. En este caso en específico se selecciona la carpeta que contiene los *frames* obtenidos del video. *Freeware Free Video to JPG Converter* genera una carpeta por cada extracción de imágenes que realiza. Por lo tanto, se selecciona esta carpeta en la opción de *Image folder*

Data Type: es el tipo de dato que procesará el COLMAP; este parámetro se puede configurar de 3 formas; imágenes de internet (*internet images*), imágenes individuales (*individual images*) o *frames* de video (*video frames*). En las pruebas preliminares se configuró este parámetro en un inicio como *video frames*; sin embargo, las reconstrucciones obtenidas presentaban deformaciones en los objetos de interés (aplicadores). Por esta razón, este parámetro se configuró como *individual images*.

Quality: esta configuración se refiere a la calidad con la que se obtiene la reconstrucción 3D; para este caso en específico, se utilizó una calidad baja (*Low*) ya que los tiempos de procesamiento con esta calidad se vieron reducidos a menos de 60 minutos obteniendo buena calidad en las reconstrucciones. Es decir, no se generaron puntos aislados en la reconstrucción 3D y la deformación en los objetos de interés se vio reducida.

Las configuraciones mencionadas se obtuvieron buscando una calidad satisfactoria en la reconstrucción 3D, tiempos de procesamiento reducidos (<60 minutos) y manejando alrededor de 200 imágenes por video.

7.4.5 Edición de la reconstrucción 3D

Al contar con la reconstrucción 3D del sistema que emula un tratamiento de hipertermia, fue necesario adecuarla para facilitar su análisis con el programa *ext_coor_apli.mat*. Para realizar esta tarea se utilizó el software de open-source *Meshlab*. En la Fig. 22 se muestra el diagrama de flujo utilizado para la edición de la reconstrucción 3D en *Meshlab*.

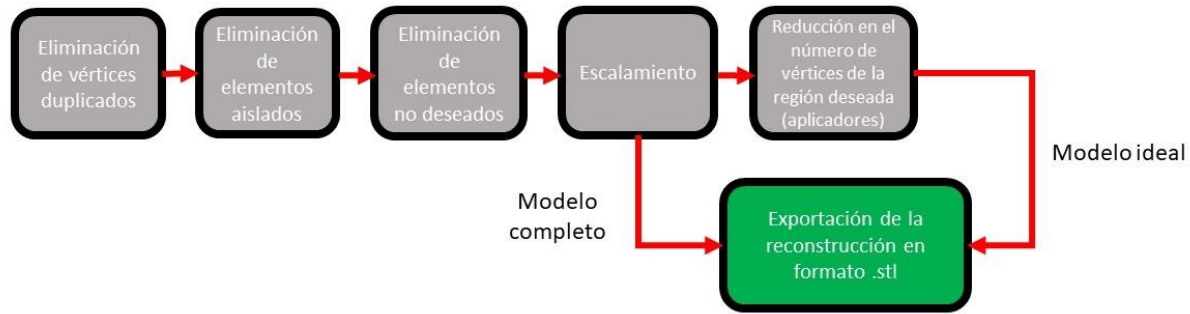


Fig. 22. Diagrama de flujo para la edición de la reconstrucción 3D en *Meshlab*. Inicia con la eliminación de vértices duplicados, de elementos aislados y de elementos no deseados, después sigue con el escalamiento. En este punto se divide en la obtención del modelo completo y el modelo ideal (cuya diferencia es la reducción en el número de vértices) para la obtención de dos archivos en formato *.stl*.

Eliminación de vértices duplicados: se eliminan todos aquellos vértices que se encuentran duplicados en la reconstrucción obtenida con el método *SfM*. Esto con el propósito de eliminar el número de datos a procesar con el programa *ext_coor_apli*.

Eliminación de elementos aislados: se eliminan pequeños grupos de vértices presentes en la reconstrucción 3D, que no están unidos a un elemento de mayor número de vértices. Como configuración predeterminada se eliminan todos aquellos elementos que tengan una distancia total del 10% de las dimensiones máximas utilizadas en la reconstrucción.

Eliminación de elementos no deseados: al realizar la reconstrucción 3D del sistema que emula un tratamiento de hipertermia, también se reconstruyen elementos que no son de interés. En este caso en específico, se reconstruye todo el maniquí, por ejemplo, incluso aquellas zonas que no tienen contacto con los aplicadores. Por lo tanto, dado que por el momento sólo nos interesa reconstruir la posición de los aplicadores, es necesario recortar o eliminar el resto de los elementos.

Escalamiento de la reconstrucción: dado que la reconstrucción obtenida no cuenta con las dimensiones reales, es necesario hacer un escalamiento de esta. Para realizar esta tarea fue necesario tener una medida de referencia. En este caso, se utilizaron las uniones de las paredes de los aplicadores construidos, ya que estas uniones son visibles en la reconstrucción digital. Estas uniones miden 24 mm; por lo tanto, se mide con *Meshlab* cuántas unidades mide esta unión en el *software*. El valor obtenido divide a 24 (que es la dimensión real) y el valor resultante es el factor de escalamiento. Este factor de escalamiento se aplica a toda la reconstrucción para dotarla de dimensiones reales. Después del escalamiento el diagrama de flujo, se divide en dos caminos que permiten generar dos modelos de la reconstrucción:

- 1) **Modelo completo:** este modelo se genera inmediatamente después del escalamiento; se guarda el archivo obtenido en formato *.stl* sin realizar ninguna otra modificación.

- 2) **Modelo ideal de la reconstrucción 3D:** este es un modelo en el cual se reduce el número de vértices obtenidos en la reconstrucción 3D. Es decir, se reduce el número de datos (vértices) a analizar en el programa *ext_coor_apli.mat* y por consiguiente se reduce el tiempo de procesamiento. El número de vértices se reduce a un número aproximado a la cantidad de vértices de los aplicadores. Es decir, en este caso se está trabajando con 2 aplicadores; por lo tanto, el número de vértices de los aplicadores es de 16. En este caso, se redujo el número de vértices de la reconstrucción al doble del número total de vértices, en este caso 32. Al finalizar la reducción del número de vértices, es necesario guardar el archivo obtenido en formato *.stl*; este modelo se denomina modelo ideal.

Como resultado final de la edición de la reconstrucción 3D en el *software Meshlab* se obtienen dos modelos de una sola reconstrucción 3D; 1) el modelo completo y 2) el modelo ideal. Ambos modelos son utilizados en el programa *ext_coor_apli.mat* para

obtención de las coordenadas (x , y , z & θ) de los aplicadores involucrados en la simulación del tratamiento de hipertermia.

7.4.6 Obtención de la posición (x , y , z & θ) de los vértices de los aplicadores reconstruidos

Una vez concluida la metodología descrita en el punto anterior (obtención de los dos modelos: modelo completo y modelo ideal), se procede a analizar dichos modelos. Por lo tanto, se implementaron modificaciones al programa *ext_coor_apli.mat* obteniendo un nuevo programa llamado *coor_recons.mat* (Apendice II). Este nuevo programa es descrito en flujo de trabajo mostrado en la Fig. 23.

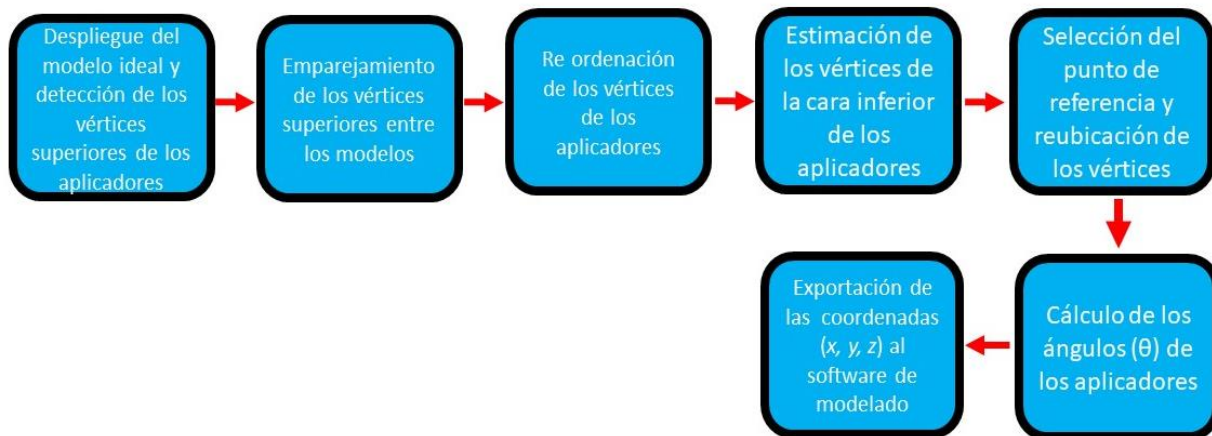


Fig. 23. Flujo de trabajo para la obtención de las coordenadas de los aplicadores en la reconstrucción 3D obtenida con el algoritmo *SfM*. Inicia con el despliegue del modelo ideal y la detección de los vértices superiores para continuar con el emparejamiento de estos con los vértices del modelo completo. Después los vértices encontrados se reordenan para dar paso a la estimación de los vértices inferiores. Posteriormente los vértices (inferiores y superiores) se reubican en función de un punto de referencia para dar paso al cálculo del ángulo θ de cada uno de los aplicadores y finalizar con la exportación de las coordenadas (x , y , z) al *software* de modelado.

Este flujo de trabajo es similar al realizado con los modelos CAD ideales de los aplicadores; sin embargo, se implementaron algunos caminos para la detección de los vértices. Cuando se trabajó con los diseños CAD, fue posible detectar los 8 vértices

del aplicador en una sola etapa. Sin embargo, al trabajar con la reconstrucción 3D hecha con el método *SfM*, los aplicadores se encuentran pegados al maniquí lo cual dificulta la detección de los vértices de esta zona. Por lo tanto, primero se detectan los vértices denominados como superiores (aquellos que no se encuentran sobre el maniquí) y después se estima la posición de los vértices denominados como inferiores (aquellos que se encuentran sobre el maniquí).

Despliegue del modelo ideal y detección de los vértices superiores de los aplicadores: el primer paso después de la edición de la reconstrucción 3D es el despliegue del modelo ideal con la función *stlread* [40], después se eliminaron los vértices duplicados con la función *unique*. Para realizar la detección de los vértices superiores de los aplicadores se utilizó la Ecuación 7; se buscó que la distancia entre los vértices sea igual AB (53 mm) \pm tolerancia (10 mm). Sin embargo, de esta forma no se detectaron solo los vértices superiores sino todos aquellos que cumplen con la condición previamente mencionada. Para llevar a cabo la discriminación de los vértices superiores el programa *coor_recons.mat* solicita al usuario un punto de referencia cerca de las caras superiores de los aplicadores; la selección de dicho punto se realiza únicamente dando clic en dicha zona. La Fig. 24 muestra la cara superior de un aplicador; sobre la cual se trabaja en este punto.

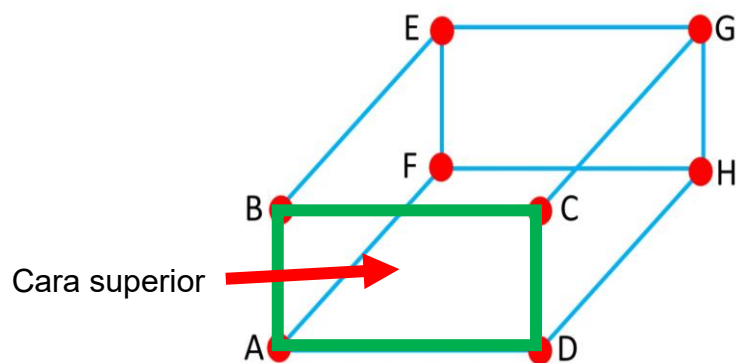


Fig. 24. Identificación de la cara superior de un aplicador.

Una vez que el usuario seleccionó un punto cercano a las caras superiores de los aplicadores involucrados (2 en este caso), se utilizó nuevamente la Ecuación 7 para

calcular la distancia entre los puntos seleccionados y los vértices. Esto con el objetivo de que aquellos 4 vértices más cercanos a los puntos seleccionados sean los pertenecientes a las caras superiores de los aplicadores.

Emparejamiento de los vértices superiores entre el modelo ideal y el modelo completo: Posterior a la detección de los vértices superiores de los aplicadores en el modelo ideal se emparejan estos vértices con los pertenecientes al modelo completo, este último tiene alrededor de 40,000 vértices, por esta razón primero se realiza la detección en el modelo ideal, el cual tiene 32 vértices. El emparejamiento de los vértices entre ambos modelos se realizó utilizando la Ecuación 7 para buscar la distancia mínima entre los vértices del modelo ideal y los vértices del modelo completo.

Reordenación de los vértices de los aplicadores: Para ordenar los vértices superiores para su fácil exportación al *software* de modelado basado el método del elemento finito se realizó la suma del valor de la coordenada x , más el valor de la coordenada y , más el valor de la coordenada z , de cada uno de los vértices. El vértice cuya suma tenga el menor valor se denomina el vértice A (ver Fig. 13). Una vez que se ha identificado el vértice A, los vértices restantes (B, C, D, E, F, G y H) se determinan calculando la distancia (Ecuación 7) entre el vértice A y el resto de los vértices reubicados. Como resultado final se obtienen los vértices (y sus respectivas coordenadas) ordenados de la siguiente forma; vértice A, B, C, D, F, E, G y H, tal y como se ordenaron con los diseños CAD previamente mencionados (Etapa 1 y Etapa 2).

Estimación de los vértices de la cara inferior de los aplicadores: una vez que los vértices superiores del modelo completo han sido ordenados, el siguiente paso fue estimar los vértices inferiores o aquellos que se encuentran pegados al maniquí. Si tomamos el vértice A como punto de partida, tenemos que, para determinar el vértice F (Fig. 25) en el modelo completo, se deben buscar aquellos vértices que sean perpendiculares al vector formado entre el vértice A y vértice D, el cual se denomina vector \vec{v} . Para determinar esto se utilizó la Ecuación 11 entre el vector \vec{v} y el vector

que se forma entre el vértice A y los demás vértices del modelo completo, el cual se denomina vector \vec{w} . La Ecuación 11 debe ser igual a 90° y todos aquellos vértices que no cumplan con esta condición son eliminados.

$$90^\circ = \arccos\left(\frac{\vec{w} \cdot \vec{v}}{|\vec{w}| \cdot |\vec{v}|}\right) \quad (11)$$

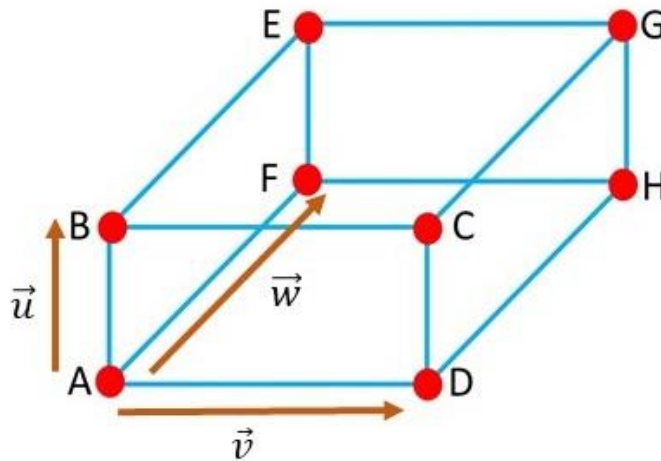


Fig. 25. Vectores utilizados para estimar los vértices inferiores.

La siguiente condición que los vértices restantes deben cumplir es que deben tener una distancia de 169.2 mm con respecto al vértice A, esta distancia corresponde a la dimensión AF del aplicador. Los vértices que cumplen con esta función son nuevamente analizados para evaluar si son perpendiculares al vector \vec{u} , el cual está formado entre el vértice A y el resto de los vértices; si no cumplen con esta última condición, son eliminados. Para comprobar los resultados obtenidos se utilizó nuevamente la Ecuación 6, solo que esta vez se aplicó entre el vector \vec{u} y el vector \vec{w} . Este proceso se repite para los vértices E, G y H, para obtener todos los vértices inferiores de un solo aplicador y se repite para cada una de las guías de onda involucradas (2 en este caso).

Selección del punto de referencia y reubicación de los vértices: una vez estimados los vértices inferiores, se cuenta con todos los vértices de los aplicadores

involucrados en el modelo completo. Por lo cual, el paso siguiente fue seleccionar un punto en el espacio 3D, el cual se vuelve el nuevo punto de referencia. Los vértices de los aplicadores se reubican en función de este punto realizando la operación descrita por la Ecuación 8.

Cálculo de los ángulos (θ) de los aplicadores: después de reubicar los vértices, se determinó el ángulo de los aplicadores involucrados en el modelo. Para calcular el ángulo, se utilizó el mismo método propuesto para las Etapas 1 y 2. Es decir, se propone un aplicador de referencia que tenga una posición básica y el uso de la Ecuación 5 entre este aplicador y los aplicadores pertenecientes al modelo completo obtenido a partir de la reconstrucción 3D obtenida mediante las fotografías tomadas a un sistema que emuló un tratamiento de hipertermia. Como producto final del procesamiento del programa *coor_recons.mat* se obtuvieron las coordenadas de cada uno de los vértices de los aplicadores involucrados en el modelo completo además de los ángulos en cada uno de los planos (*xy*, *xz* & *yz*) para cada aplicador.

Exportación de las coordenadas (*x*, *y*, *z*): la exportación de las coordenadas se realizó de la misma forma que en la Etapa 2, basta con transcribir las coordenadas tal y como se obtienen en el programa *coor_recons*. Por lo tanto, es necesario crear un hexaedro por cada uno de los aplicadores involucrados en el modelo completo.

8. Resultados

8.1 Evaluación de la factibilidad de uso del aplicador “guía de onda” para la implementación del tratamiento de tumores en la interfaz músculo-hueso.

La Fig. 26 muestra cómo viaja el campo eléctrico dentro de la guía de onda rectangular; de la misma manera, se muestra como el campo eléctrico se transmite a cada capa de tejido. La Fig. 26a) y 26b) muestra la distribución del campo eléctrico obtenida para un bolus de agua (37°C) con espesores de 5 mm y 20 mm y para espesores de grasa y músculo de 30 mm y 50 mm, respectivamente. En la figura 26b) se aprecia cómo el campo eléctrico tiene una deformación cuando el espesor del bolus de agua fue de 20 mm; incluso se observa que el valor de campo eléctrico disminuye (550 V/m) en comparación con su contraparte donde el valor máximo fue de 667 V/m.

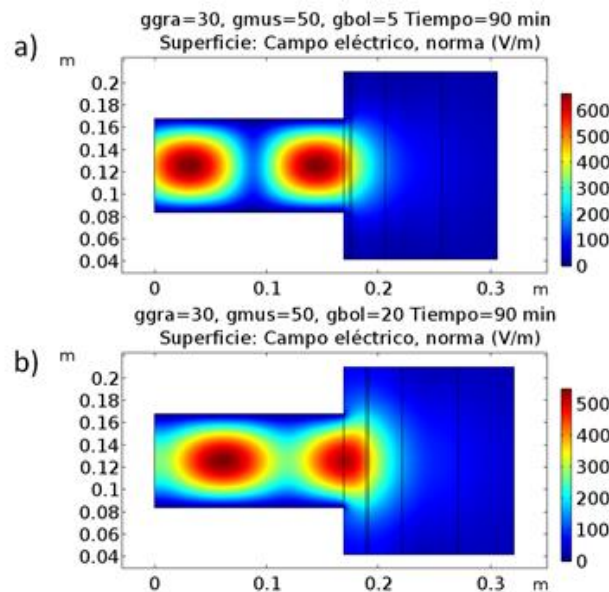


Fig. 26. Distribuciones de campo eléctrico logradas al utilizar distintos espesores de bolus de agua, a) campo eléctrico normal en la guía de ondas con un bolus de agua destilada de 5 mm de espesor, espesores máximos de grasa (30 mm) y músculo (50 mm), b) campo eléctrico normal en la guía de ondas con un bolus de agua destilada de 20 mm de espesor, espesores máximos de grasa (30 mm) y músculo (50 mm). Una potencia de 50 W en ambos casos.

La Fig. 27 muestra cómo aumenta el SWR cuando aumentan los espesores de grasa o músculo. Es importante mencionar que cuando el músculo tiene 1 mm de espesor, el valor de SWR aumenta, pero de una manera menor (~0.4 desde su valor mínimo hasta su valor máximo) en comparación con los otros espesores (~1.6 en promedio desde el valor mínimo a su valor máximo).

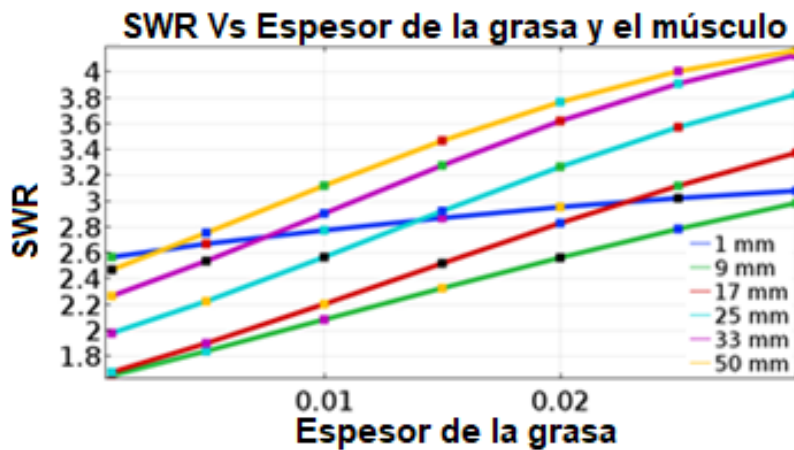


Fig. 27. SWR a diferentes espesores de grasa (eje x) y músculo (indicado en las etiquetas).

La Fig. 28 muestra la distribución de temperatura obtenida cuando se utilizó un bolus de agua de 5 mm y 20 mm. En este caso, la temperatura del bolus de agua fue de 37°C y la guía de onda se alimentó con 50 W. Se aprecia que cuando el bolus tiene un espesor de 5 mm se obtiene una temperatura de 38.4°C, en comparación con un bolus de agua de 20 mm, la temperatura máxima disminuye a 38.2°C.

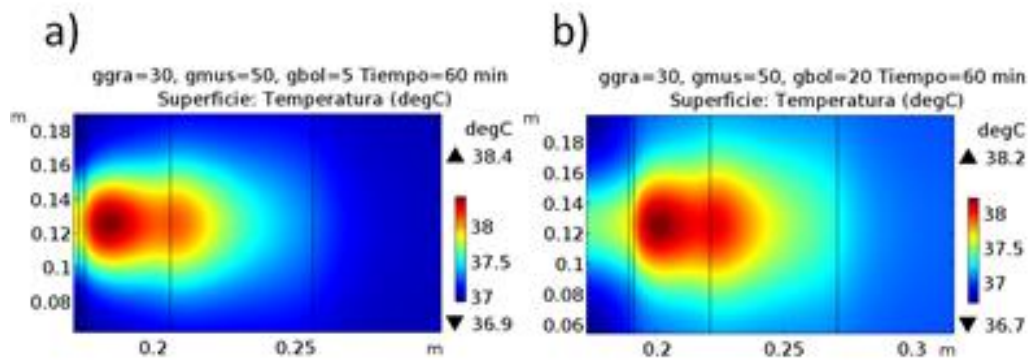


Fig. 28. Resultados del estudio paramétrico obtenidos con diferentes espesores en el bolus de agua, a) distribución de temperatura con bolus de agua destilada de 5 mm de espesor, b) distribución de temperatura con bolus de agua destilada de 20 mm de espesor.

La Fig. 29 muestra la distribución de temperatura obtenida cuando el bolus de agua tiene un espesor de 5 mm y 20 mm, una temperatura de 42°C y la guía de onda se alimenta con 100 W. Se observa cómo influye la temperatura del bolus de agua en la distribución de la temperatura en los tejidos. Es claro que la temperatura aumenta con una potencia de 100 W (aumenta 1.5°C en comparación con un bolus de agua de 5 mm a 50 W). Sin embargo, para un bolus de agua de 20 mm a una temperatura inicial de 42°C, el calor se enfoca dentro del bolus induciendo un aumento de temperatura más alto en él (40.4°C) y no en los tejidos (39.5°C).

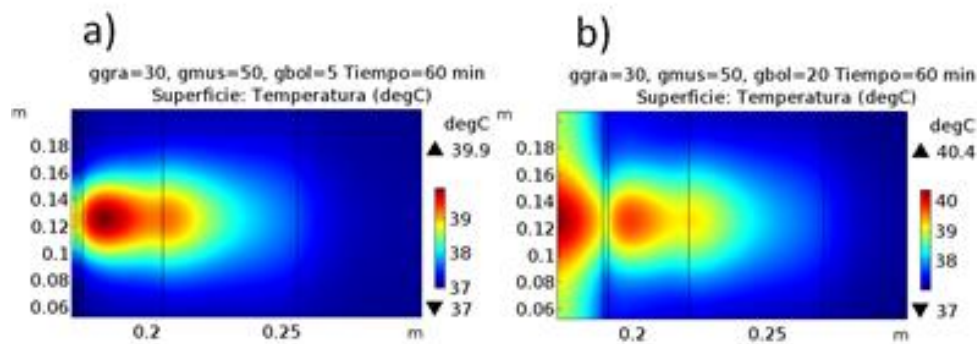


Fig. 29. Resultados del estudio paramétrico obtenidas con diferentes espesores en el bolus de agua a una temperatura de 42°C. a) distribución de temperatura con un bolus de agua destilada de 5 mm de espesor, b) distribución de temperatura con un bolus de agua destilada de 20 mm de espesor.

La Fig. 30 muestra los perfiles de temperatura obtenidos a lo largo de una línea en el eje x, justo en el medio de los tejidos. Aquí, se evaluó la profundidad de penetración del calor cuando el espesor de la grasa se consideró constante (1 mm), el espesor del músculo varía de 1 mm a 50 mm y la guía de onda se alimenta con 50 W de potencia. La temperatura máxima (40.6°C) se alcanzó cuando el espesor del músculo fue de 9 mm, sin embargo, su profundidad de penetración es de aprox. 9 mm. Mientras que la temperatura que se alcanzó a 16.5 mm fue 40.21°C con un espesor del músculo de 17 mm. En el resto de los casos, la temperatura es inferior a los 40°C.

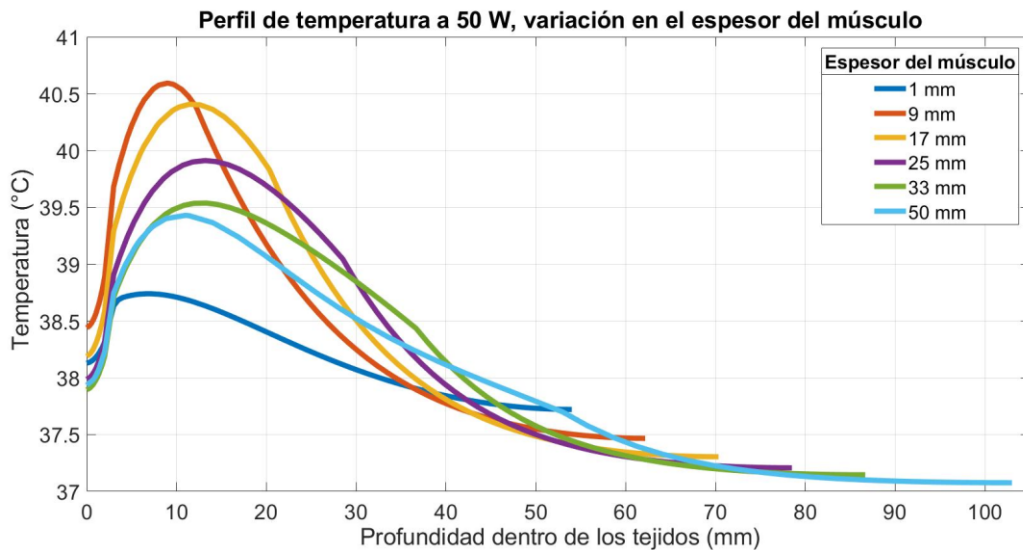


Fig. 30. Perfiles de temperatura registrados en los tejidos con 1 mm de espesor en grasa y valores variables en el espesor del músculo (1 mm, 9 mm, 17 mm, 25 mm, 33 mm y 50 mm), con una potencia de 50 W.

La Fig. 31 muestra el perfil de temperatura cuando el espesor de la grasa es variante y se aplica una potencia de 50 W y 100 W. Es importante mencionar que el espesor del músculo se consideró constante (9 mm). Se observó un aumento de temperatura cuando la guía de onda es alimentada con 100 W, el aumento entre las temperaturas máximas es más grande que 1.5°C. Este aumento en la temperatura también implica un aumento en la profundidad de penetración. Cuando la grasa tiene un espesor de 1 mm, la temperatura tiene una penetración de 13.26 mm con un valor de 40.19°C, esto con una potencia de 50 W. Cuando el espesor de la grasa se consideró de 1 mm pero

con una potencia de 100 W la penetración de la temperatura fue de 25.81 mm con un valor de 40.1°C. Además, se pudo observar que, si el espesor de la grasa aumenta, la temperatura máxima disminuye en ambos casos. Por ejemplo, cuando la grasa tiene un espesor de 1 mm la temperatura máxima es de 44.15°C a 100 W, pero cuando el espesor de la grasa fue de 30 mm, la temperatura máxima disminuye hasta 40.7°C a la misma potencia.

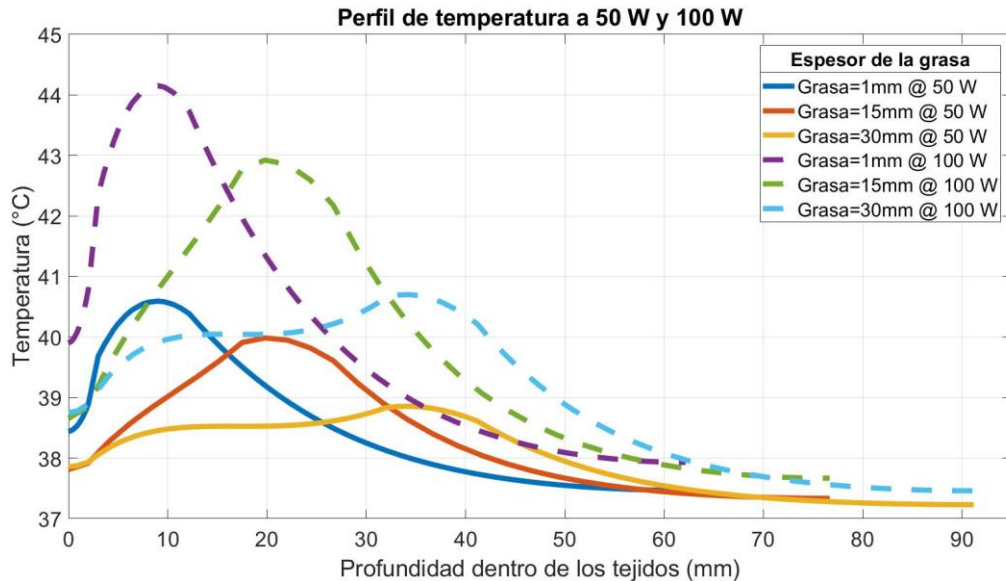


Fig. 31. Perfil de temperatura registrado en los tejidos con 9 mm de espesor de músculo y espesor de grasa variable. Aplicando 50 W y 100 W de potencia.

La Fig. 32 muestra una comparación entre las temperaturas alcanzadas cuando la grasa y el músculo tienen 1 mm y 9 mm de espesor respectivamente, durante 60 min y 90 min, con una potencia de 50 W y 100 W. Se puede apreciar una diferencia notable en las temperaturas cuando las potencias de aplicación varían, a 60 minutos la diferencia es de 3.56°C y a 90 minutos a 3.82°C

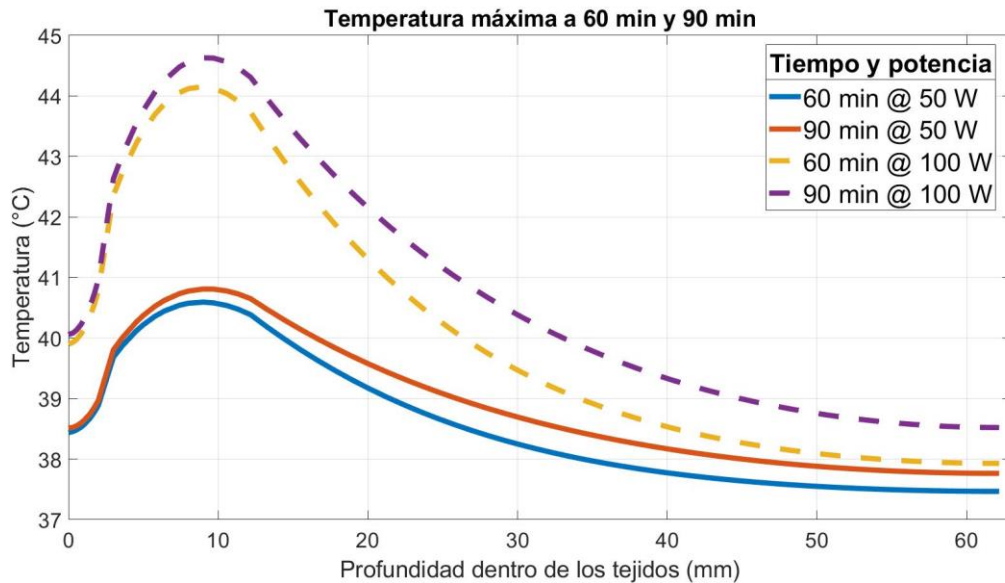


Figura 32. Temperaturas máximas obtenidas al utilizar espesores constantes de grasa (1 mm) y músculo (9 mm), a dos potencias de funcionamiento (50 W y 100 W). Teniendo un tiempo de aplicación de 60 min y 90 min.

8.2 Etapa 1: Obtención de las coordenadas del diseño CAD de un aplicador

8.2.1 Diseño CAD de un aplicador

En la Fig. 33 se muestran las vistas frontal, lateral, superior y trimétrica del diseño CAD ideal del aplicador (guía de onda). Se puede observar en las diferentes vistas líneas punteadas de color azul, las cuales nos indican que el aplicador es hueco, además el aplicador no posee las ranuras en donde se colocan los tornillos (pestañas laterales); también es importante mencionar que no se diseñó el espacio en donde se localiza la conexión hacia el sistema de RF (entrada coaxial).

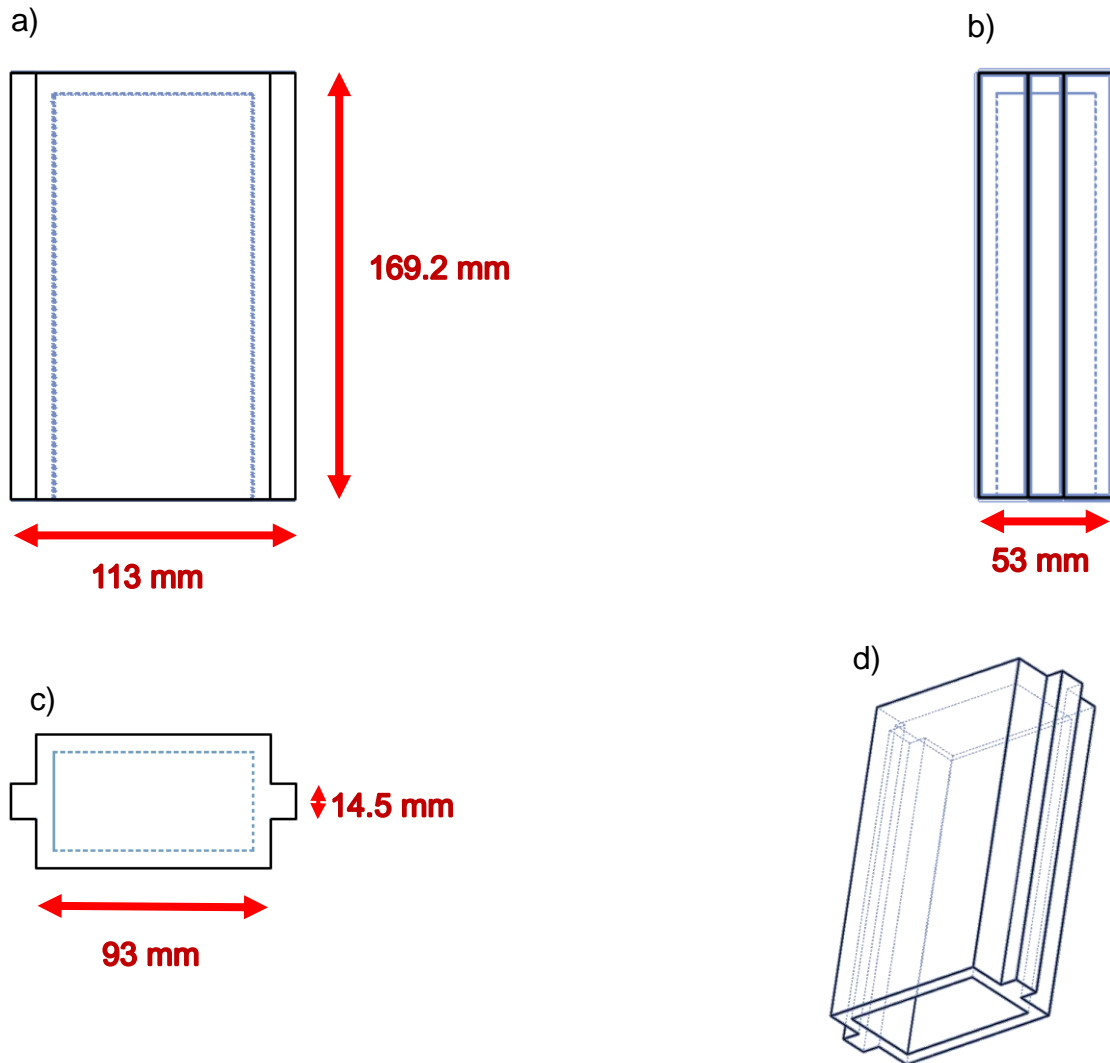


Fig. 33. Vistas del diseño CAD del aplicador ideal, a) vista frontal, b) vista lateral, c) vista superior, d) vista trimétrica.

8.2.2 Obtención de las coordenadas (x , y , z & θ) de sus vértices

8.2.2.1 Despliegue del archivo .stl en *Matlab* y eliminación de vértices duplicados.

Posterior a la obtención del diseño CAD del aplicador, el algoritmo llamado *ext_coor_apli.mat* despliega el diseño CAD (archivo en formato .stl) en *Matlab* con la función *stlread* [40]. En la Fig. 34 se muestra el aplicador desplegado, se puede

observar que el aplicador no se encuentra en el origen del plano tridimensional. Las unidades de los ejes son los milímetros de acuerdo con las dimensiones del aplicador. Después de que el archivo ha sido desplegado, se eliminan los vértices duplicados con la función *unique*; esta función se aplica a la variable que contiene todos los vértices del archivo desplegado (esta variable es la denominada *vertices*).

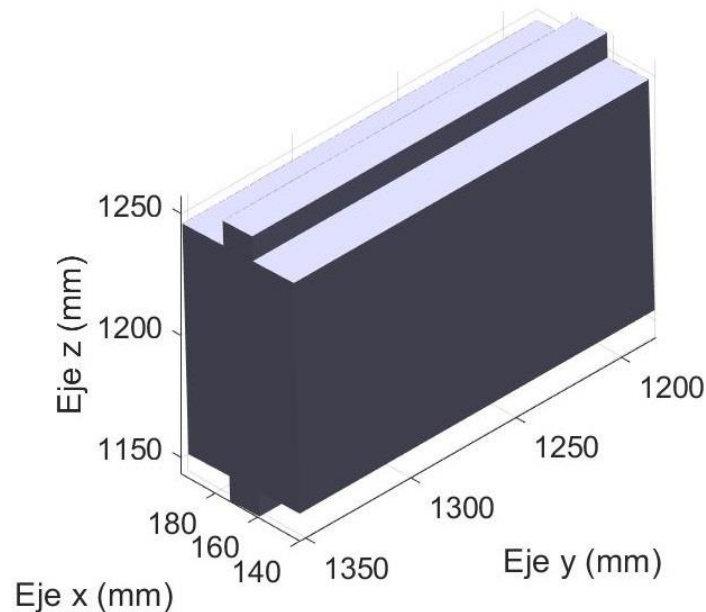


Fig. 34. Aplicador ideal desplegado en Matlab.

8.2.2.2 Detección de los vértices del aplicador y sus coordenadas (x, y, z)

Después de que se eliminaron los vértices duplicados, se realizó la detección de los vértices pertenecientes al aplicador. Para llevar esto a cabo, se utilizó la Ecuación 7 comparando dos vértices a la vez. Al utilizar esta ecuación se obtiene la distancia entre los dos vértices que son analizados; si el resultado de la Ecuación 7 es igual a algún valor de la Tabla 9 (para este caso en particular 53 mm) ambos vértices se guardan en una nueva variable llamada *coor_vertices_encontrados1*. Esta variable almacena las coordenadas x, y, z de los vértices que cumplen con alguna distancia de la Tabla 9. En la Fig. 35 se muestran los vértices detectados y los vértices originales.

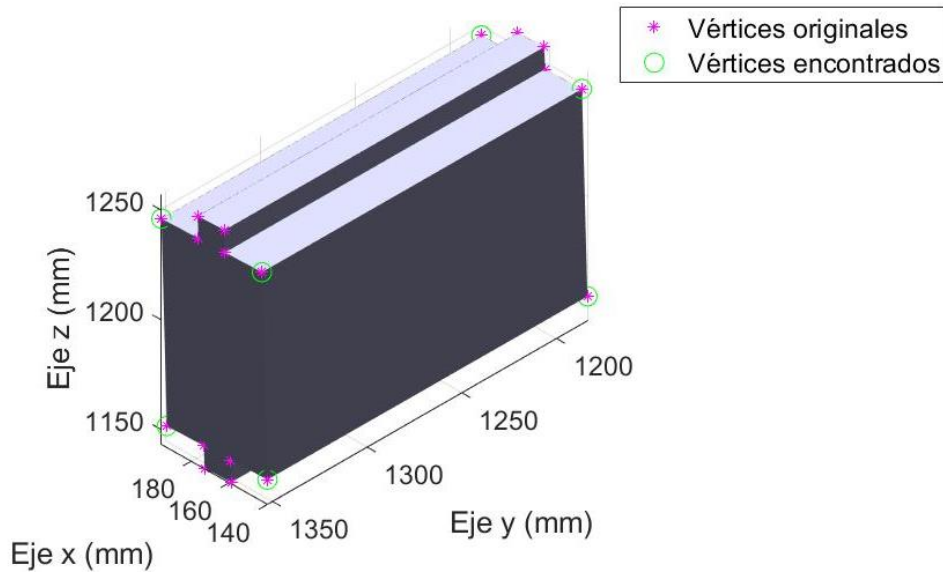


Fig. 35. Despliegue de los vértices detectados y los vértices originales.

8.2.2.3 Selección de un punto arbitrario en el espacio 3D y reubicación de los vértices

Después de que se detectaron los vértices en el diseño CAD del aplicador, el algoritmo *ext_coor_apli.mat* solicita al usuario seleccionar un punto en el espacio 3D para reubicar los vértices detectados en función de este punto, el cual se convierte en el nuevo origen (0,0,0). En la Fig. 36 se muestran los vértices detectados del diseño CAD y el punto seleccionado por el usuario. En la Fig. 37 se muestran los vértices del aplicador reubicado en función del punto seleccionado (origen del plano con coordenadas 0, 0, 0).

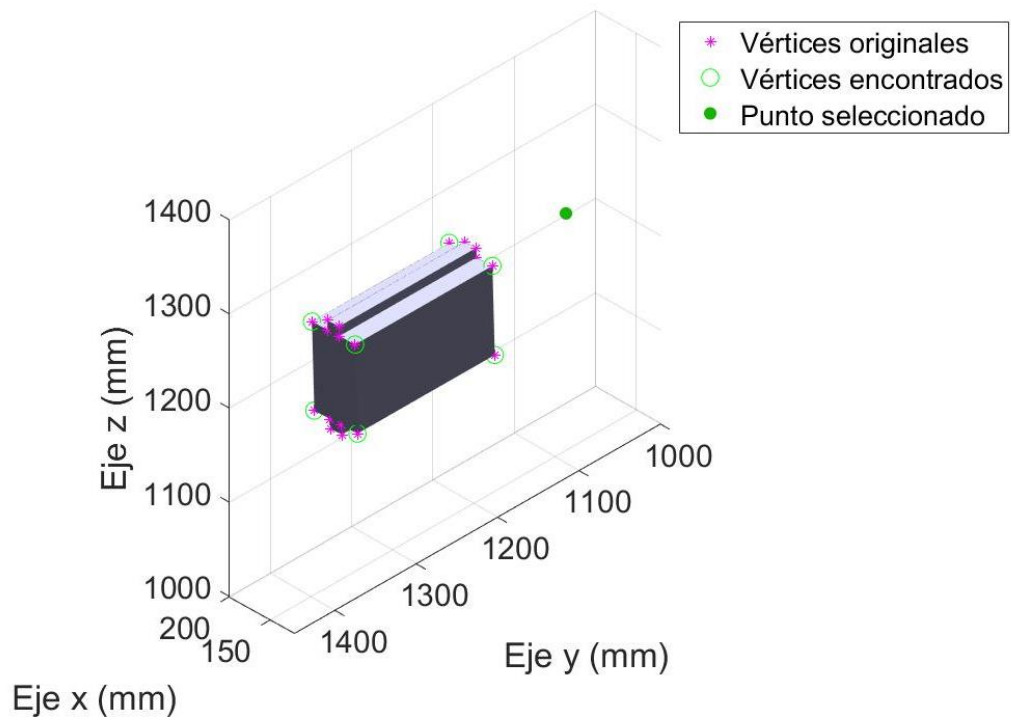


Fig. 36. Despliegue del aplicador, los vértices detectados y el punto seleccionado por el usuario (nuevo origen).

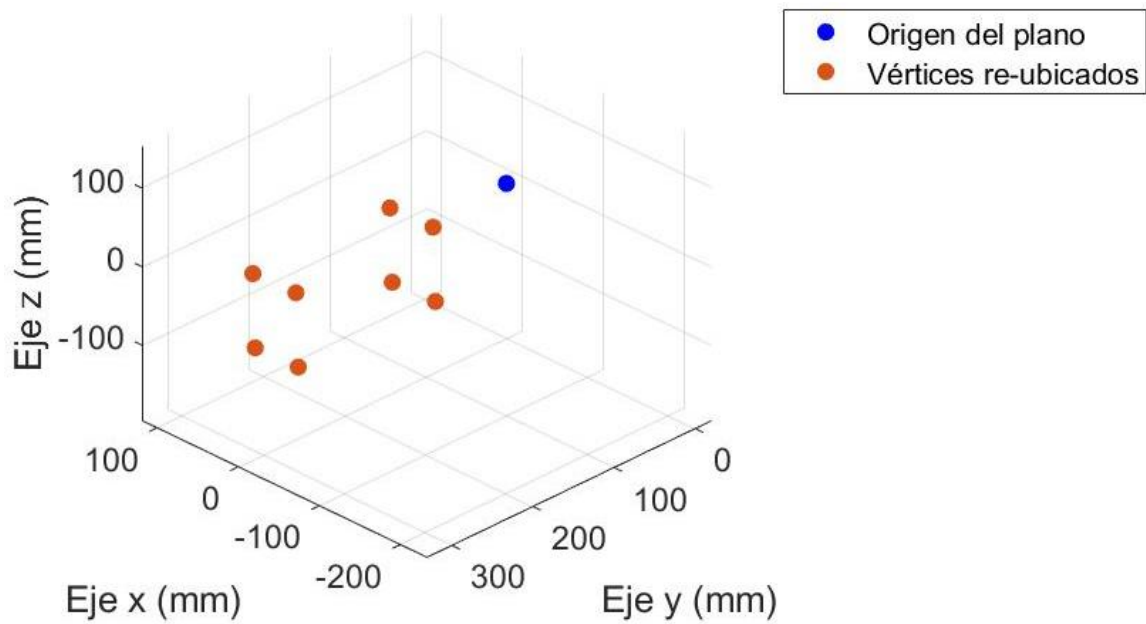


Fig. 37. Despliegue de los vértices del aplicador reubicados en función del punto seleccionado (origen del plano) por el usuario.

8.2.2.4 Reordenación de los vértices del aplicador

Una vez que los vértices fueron reubicados en función del nuevo origen, estos se ordenaron de una forma específica. En la Fig. 38 se muestran los vértices del aplicador antes y después de ser ordenados. La variable denominada *coor_reubicadas* es aquella que contiene las coordenadas de cada uno de los vértices del aplicador justo después de ser reubicados de acuerdo con el nuevo origen (punto seleccionado por el usuario) y la variable *coordenadas_exportación* contiene las coordenadas de estos mismos vértices solo que ordenadas para que puedan ser utilizadas en el *software* de modelado.

<i>coor_reubicadas</i> =			<i>coordenadas_exportacion</i> =		
-20.2236	280.3114	-83.4633	33.0173	111.2006	-85.5988
-17.1350	280.1101	9.4852	32.7471	280.3999	-85.2233
-19.9534	111.1121	-83.8387	-20.2236	280.3114	-83.4633
-16.8649	110.9107	9.1097	-19.9534	111.1121	-83.8387
32.7471	280.3999	-85.2233	36.1058	110.9992	7.3497
35.8357	280.1985	7.7252	35.8357	280.1985	7.7252
33.0173	111.2006	-85.5988	-17.1350	280.1101	9.4852
36.1058	110.9992	7.3497	-16.8649	110.9107	9.1097

Fig. 38. Coordenadas de cada uno de los vértices del aplicador (8 vértices) después de su reubicación (*coor_reubicadas*) y después de su reordenación (*coordenadas_exportacion*).

8.2.2.5 Cálculo del ángulo θ del aplicador

El ángulo θ del aplicador se calcula para cada uno de los planos (*xy*, *xz* & *yz*) y para los ángulos existentes entre las distancias AB, AD y AF del aplicador reubicado y el aplicador de referencia (tal y como se explica en la sección 7.2.2.1). Como resultado *Matlab* despliega una variable denominada *theta*, la cual es una matriz de 3X3; en esta variable, las columnas son los planos (*xy*, *xz* & *yz*) y las filas son las distancias AB, AD y AF. La Tabla 12 muestra el resultado obtenido de calcular los ángulos para las distancias y los planos respectivos del aplicador desplegado en *Matlab*, el ángulo real del aplicador de acuerdo al diseño CAD y el porcentaje de erro entre el ángulo real y

el calculado. La Fig. 39 muestra los vértices del aplicador reubicado, así como los vértices de referencia que sirvieron para calcular los ángulos θ en cada uno de los planos.

Tabla 12. Ángulos de los vértices del aplicador reubicado con respecto a los vértices del aplicador de referencia en los diferentes planos.

	Ángulo en el plano xy (°)		% de error	Ángulo en el plano xz (°)		% de error	Ángulo en el plano yz (°)		% de error
	Calculado	Real		Calculado	Real		Calculado	Real	
Vector 1 (distancia AB)	NaN	NaN	NaN	88.5	90	1.6%	2.87	0	NaN
Vector 2 (distancia AD)	3.73	0	NaN	88.5	90	1.6%	NaN	NaN	NaN
Vector 3 (distancia AF)	0.091	0	NaN	NaN	NaN	NaN	0.127	0	NaN

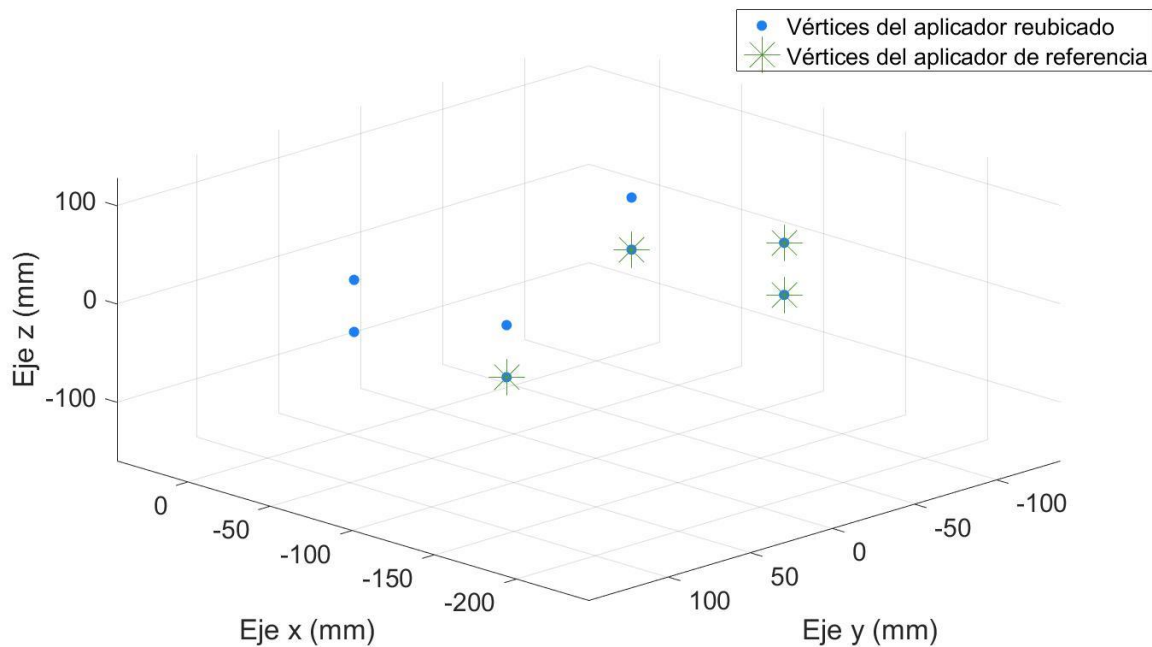


Fig. 39. Vértices reubicados del aplicador y vértices del aplicador de referencia para al cálculo del ángulo θ .

La Tabla 12 presenta en algunos casos las letras *NaN* las cuales significan *Not a Number*. Estas letras se presentan debido a que no se puede calcular el ángulo en ese

plano; esto se debe a que la distancia que se está analizando es perpendicular a dicho plano. Por ejemplo, en el vector 1 en el plano xy, la distancia AB es perpendicular al plano xy por lo cual no se puede calcular un ángulo. Las letras NaN se presentan en la columna correspondiente al % de error, esto debido a que la fórmula utilizada para obtener este valor es:

$$\% \text{ de error} = \frac{|\text{valor real} - \text{valor calculado}|}{\text{valor real}} \quad (12)$$

por lo tanto, no se puede resolver esta ecuación cuando el valor real es igual a 0. De acuerdo con la información obtenida en la variable *theta*, podemos hacer caso omiso a los ángulos resultantes en algunos planos, por ejemplo, en el caso mencionado en la Tabla 12 los ángulos obtenidos en los plano xy & yz, no brindan información fidedigna de los vértices reubicados debido a la posición del aplicador. En este caso en específico, este aplicador se encuentra solo rotado a lo largo del eje y dando como resultado que el plano xz nos brinde la información de esta rotación. Considerando que un aplicador puede estar rotado en más de un eje, la variable *theta* calcula el ángulo en cada plano y tocaría al usuario decidir qué información (ángulos) son de utilidad, en este proyecto la coordenada *theta* es considerada de forma provisional como datos meramente informativos. Después del cálculo de esta coordenada, la Etapa 1 finaliza con el agrupamiento de las coordenadas (*x*, *y*, *z*, θ) de cada uno de los vértices de un solo aplicador, esta información nos brinda la posición de este de acuerdo con un punto de referencia arbitrario.

8.3 Etapa 2: Obtención de las coordenadas del diseño CAD de tres aplicadores

8.3.1 Diseño CAD de un arreglo de 3 aplicadores

Después de obtener las coordenadas (*x*, *y*, *z* & θ) de un solo aplicador, se realizó un nuevo diseño CAD, el cual involucra la participación de 3 aplicadores colocados alrededor de un cilindro para emular un tratamiento de hipertermia. En la Fig. 40 se

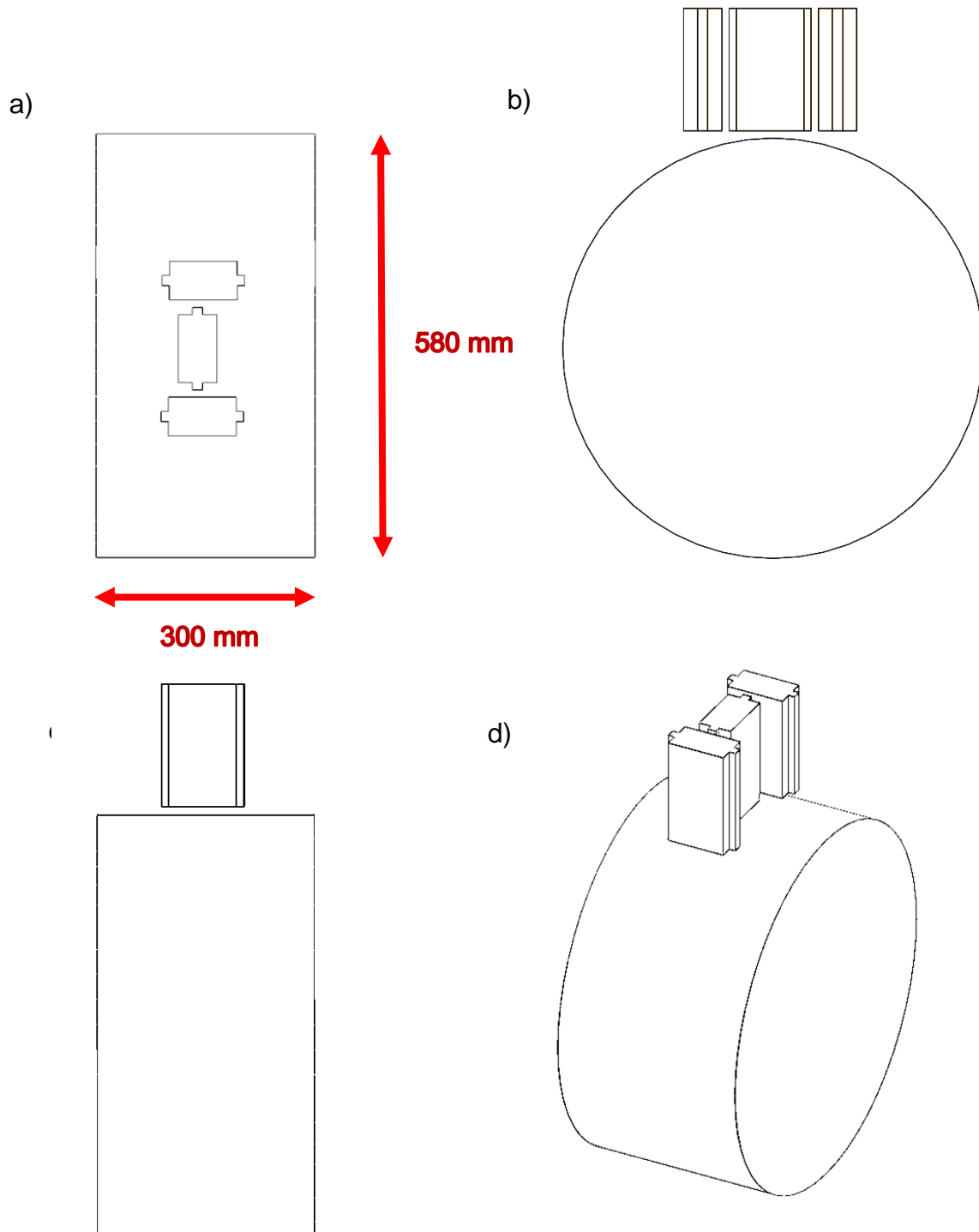


Fig. 40. Vistas del diseño CAD del arreglo de 3 aplicadores ubicados sobre un cilindro que emula una pierna humana (medidas antropométricas), a) vista superior, b) vista frontal, c) vista lateral, d) vista trimétrica.

muestra la vista superior, frontal, lateral y trimétrica del arreglo de los aplicadores y el cilindro. En este diseño CAD, los aplicadores se encuentran a una distancia de 20 mm entre sí y una separación del centro del cilindro a los vértices de los aplicadores se encuentra en un rango de 302.95 mm – 324.37 mm; esta distancia depende del vértice que esté siendo inspeccionado. El cilindro tiene un diámetro de 580 mm que corresponden a medidas antropométricas y tiene un espesor de 300 mm. El aplicador localizado en medio se encuentra rotado 90° con respecto a los aplicadores laterales.

8.3.2 Obtención de las coordenadas (x , y , z & θ) de los aplicadores

Después de la obtención del diseño CAD del arreglo de los aplicadores, el siguiente paso es la obtención de las coordenadas (x , y , z & θ) de ellos; para esto se siguió el mismo flujo de trabajo de la Etapa 1 (Fig. 6), además de la exportación de coordenadas al *software* de modelado. A continuación, se describen las fases de la etapa 2 para la obtención y exportación de las coordenadas (x , y , z & θ) de los aplicadores.

8.3.2.1 Despliegue del archivo .stl y eliminación de los vértices duplicados.

El algoritmo *ext_coor_apli.mat* despliega el diseño CAD (archivo en formato .stl) del arreglo de los tres aplicadores en *Matlab* con la función *stlread* [40] al igual que en la Etapa 1. En la Fig. 41 se muestra diseño CAD desplegado, en la cual se puede observar que ni los aplicadores ni el cilindro se encuentran en el origen del plano tridimensional. Los vértices duplicados se eliminan con la función *unique*, aplicada a la variable *vértices* (la cual es proporcionada al desplegar el archivo . stl con la función *stlread*).

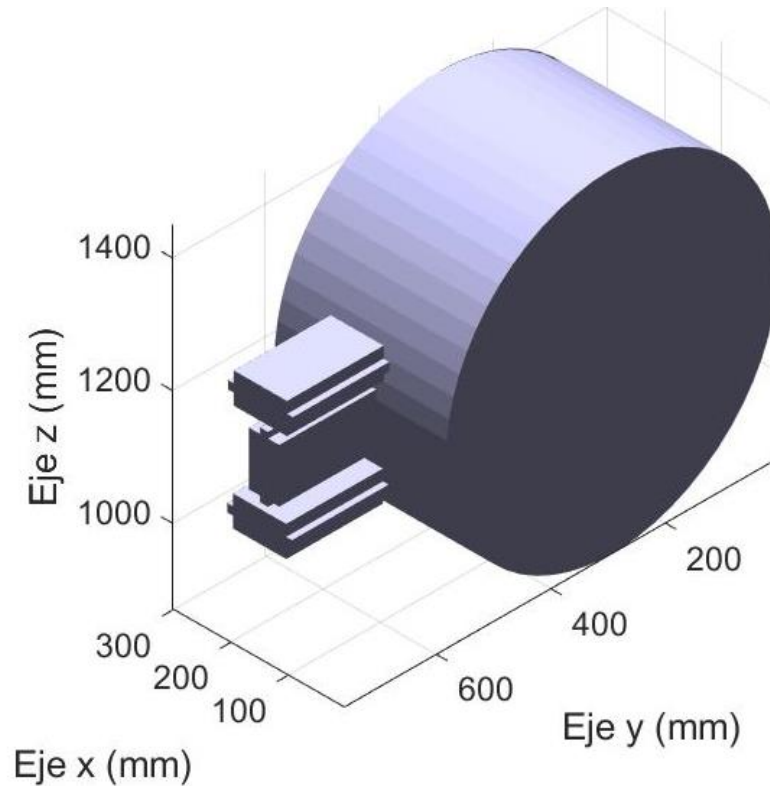


Fig. 41. Arreglo de aplicadores desplegado en *Matlab*.

8.3.2.2 Detección de los vértices de los aplicadores y sus coordenadas (x, y, z)

Posterior a la eliminación de los vértices duplicados, se realizó la detección de los vértices pertenecientes a los aplicadores. En la Fig. 42 se muestra el arreglo de aplicadores desplegado y todos los vértices del diseño CAD (vértices originales), además de los vértices detectados. Es decir, aquellos vértices que tienen una distancia de 53 mm (dimensión AB) entre sí.

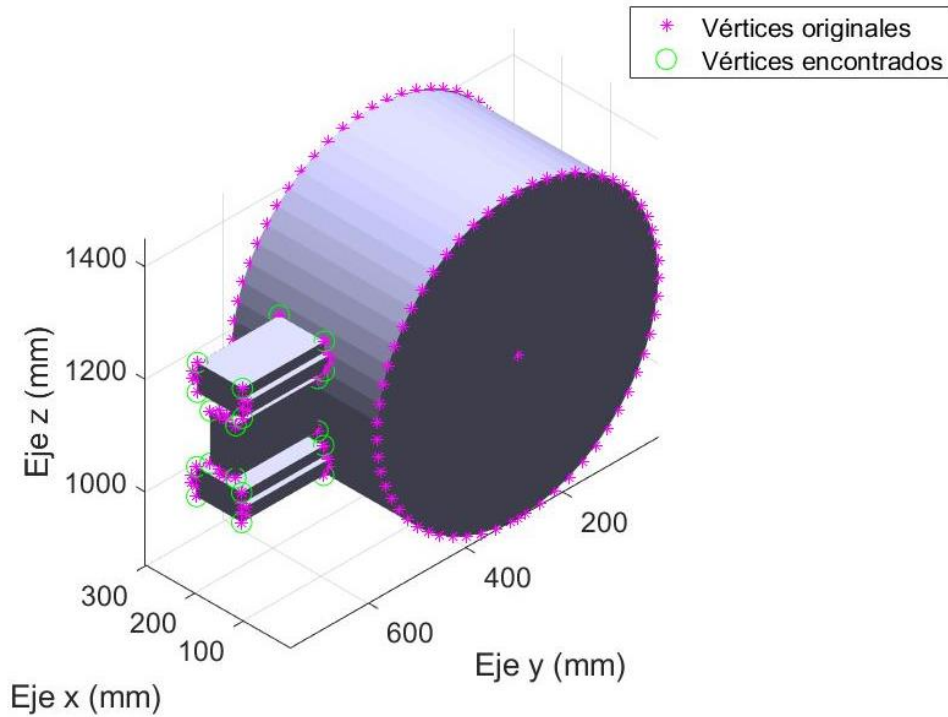


Fig. 42. Despliegue de los vértices encontrados y los vértices originales del arreglo de aplicadores.

8.3.2.3 Selección de un punto arbitrario en el espacio 3D y reubicación de los vértices

Una vez que se detectaron los vértices en el diseño CAD, el algoritmo *ext_coor_apli.mat* solicita al usuario seleccionar un punto en el espacio 3D para reubicar los vértices encontrados en función de este punto de la misma forma que en la Etapa 1. En la Fig. 43 se observa el punto seleccionado por el usuario además de los vértices originales y los vértices encontrados. En la Fig. 44 se muestran los vértices de los aplicadores reubicados en función del punto seleccionado (el cual pasa a ser el origen del plano con coordenadas 0, 0, 0).

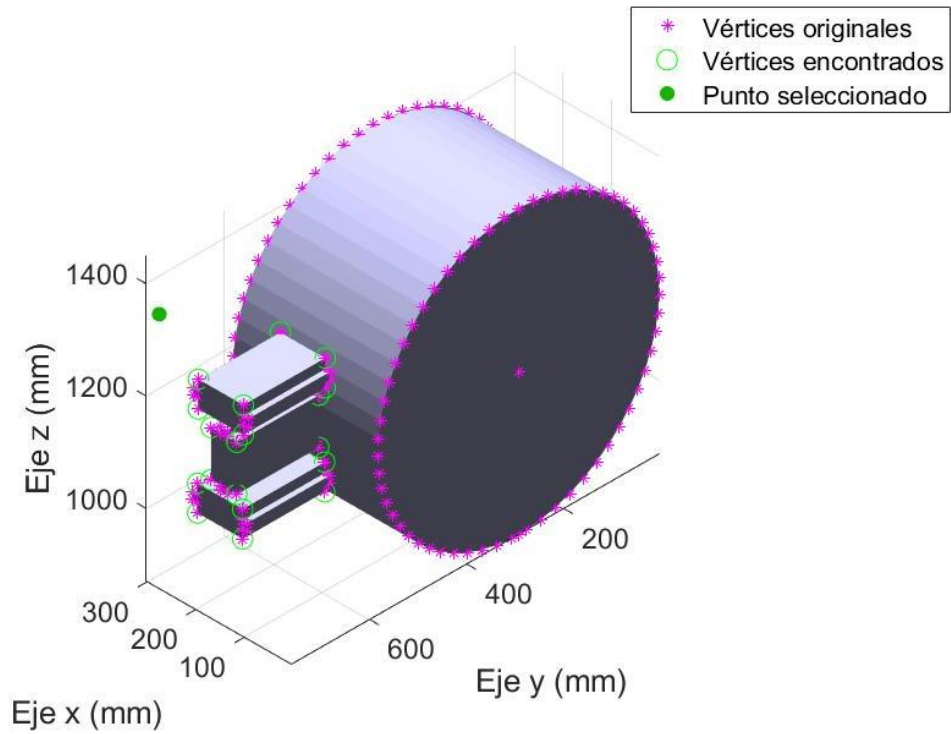


Fig. 43. Despliegue del arreglo de aplicadores, los vértices encontrados y el punto seleccionado por el usuario (nuevo origen).

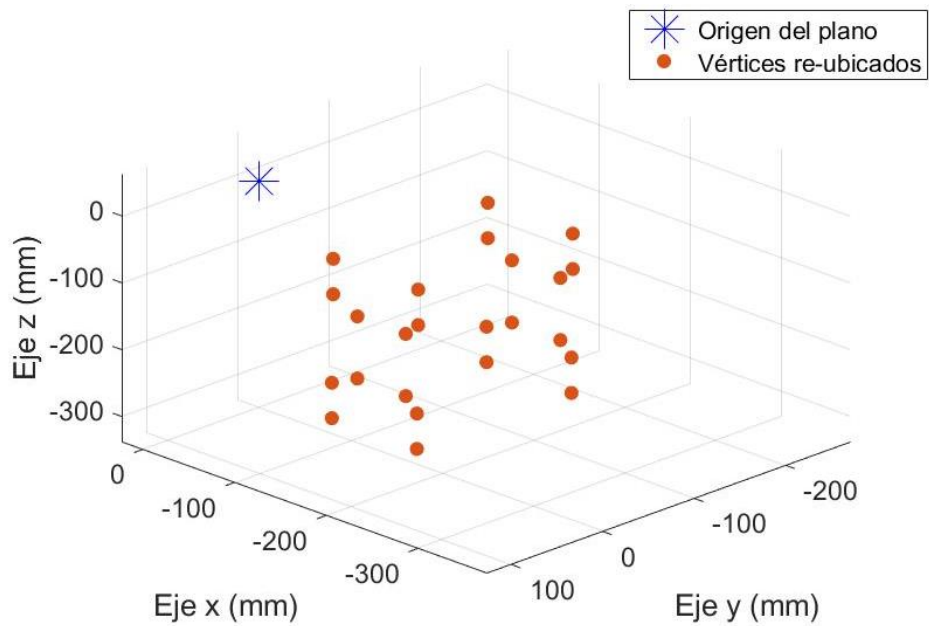


Fig. 44. Despliegue de los vértices de los aplicadores reubicados en función del punto seleccionado (origen del plano) por el usuario.

8.3.2.4 Reordenación de los vértices del arreglo de aplicadores

Después de que los vértices de los aplicadores fueron reubicados en función del nuevo origen (punto seleccionado por el usuario), estos vértices se ordenaron de una forma específica (de acuerdo con lo descrito en la subsección 7.2.2). La Fig. 45 muestra los vértices del aplicador antes y después de ser ordenados tal y como se hizo en la Etapa 1. En este caso, como se trabajó con tres aplicadores, las variables *coord_reubicadas* y *coordenadas_exportacion* están agrupadas en tres matrices (una por cada aplicador) con un tamaño de 3X8, ya que son tres coordenadas (x, y, z) por cada vértice de cada aplicador (8 vértices por cada uno).

```
coord_reubicadas(:, :, 1) =      coordenadas_exportacion(:, :, 1) =
-41.7248  -48.9079  568.3460      -41.7248  -48.9079  568.3460
 51.2752  -48.9079  568.3460      51.2752  -48.9079  568.3460
-41.7248  120.2921  568.3460      51.2752  -48.9079  621.3460
 51.2752  120.2921  568.3460      -41.7248  -48.9079  621.3460
-41.7248  -48.9079  621.3460      -41.7248  120.2921  568.3460
 51.2752  -48.9079  621.3460      51.2752  120.2921  568.3460
-41.7248  120.2921  621.3460      51.2752  120.2921  621.3460
 51.2752  120.2921  621.3460      -41.7248  120.2921  621.3460

coord_reubicadas(:, :, 2) =      coordenadas_exportacion(:, :, 2) =
-40.3009  -41.1403  373.6502      -40.3009  -41.1403  373.6502
 52.6991  -41.1403  373.6502      52.6991  -41.1403  373.6502
-40.3009  128.0596  373.6502      52.6991  -41.1403  426.6502
 52.6991  128.0596  373.6502      -40.3009  -41.1403  426.6502
-40.3009  -41.1403  426.6502      -40.3009  128.0596  373.6502
 52.6991  -41.1403  426.6502      52.6991  128.0596  373.6502
-40.3009  128.0596  426.6502      52.6991  128.0596  426.6502
 52.6991  128.0596  426.6502      -40.3009  128.0596  426.6502

coord_reubicadas(:, :, 3) =      coordenadas_exportacion(:, :, 3) =
-22.8527  128.1158  447.8826      30.3882  -40.9950  445.7472
-19.7641  127.9145  540.8311      30.1180  128.2043  446.1226
-22.5825  -41.0835  447.5072      -22.8527  128.1158  447.8826
-19.4940  -41.2850  540.4556      -22.5825  -41.0835  447.5072
 30.1180  128.2043  446.1226      33.4767  -41.1965  538.6956
 33.2066  128.0029  539.0711      33.2066  128.0029  539.0711
 30.3882  -40.9950  445.7472      -19.7641  127.9145  540.8311
 33.4767  -41.1965  538.6956      -19.4940  -41.2850  540.4556
```

Fig. 45. Coordenadas de cada uno de los vértices de cada aplicador después de su reubicación (*coord_reubicadas*) y después de su reordenación (*coordenadas_exportacion*).

8.3.2.5 Cálculo del ángulo θ de los aplicadores

Después de que los vértices de los aplicadores fueron reordenados, se calculó el ángulo θ de cada aplicador para cada uno de los planos (xy , xz & yz) y para los ángulos existentes entre las distancias AB, AD y AF del aplicador reubicado y el aplicador de referencia (de la misma forma que en la Etapa 1). Como resultado *Matlab* despliega una variable denominada *theta* (matriz de 3X3) para cada aplicador. En la Fig. 46 se muestra la variable *theta* desplegada en *Matlab*.

```
theta(:,:,1) =  
  
    NaN     0     0  
     0     0    NaN  
     0    NaN     0  
  
theta(:,:,2) =  
  
    NaN     0     0  
     0     0    NaN  
     0    NaN     0  
  
theta(:,:,3) =  
  
    NaN    90    NaN  
    NaN    90    NaN  
     0    NaN     0
```

Fig. 46. Variable *theta* desplegada en *Matlab*.

Para este caso los aplicadores 1 y 2 (los cuales son los aplicadores que se encuentran en los lados, ver Fig. 42) se encuentran perfectamente alineados con el aplicador de referencia (ver Fig. 47). Lo cual da como resultado 0° en cada plano para cada una de las distancias. Sin embargo, para el aplicador que se encuentra en medio (aplicador 3), hay una rotación de 90° con respecto al aplicador de referencia (ver Fig. 48). Se puede observar que los vértices del aplicador central no coinciden en su totalidad con los vértices del aplicador de referencia. Esta rotación se presenta en el plano xz (ver Tabla 13) para las distancias AB y AD.

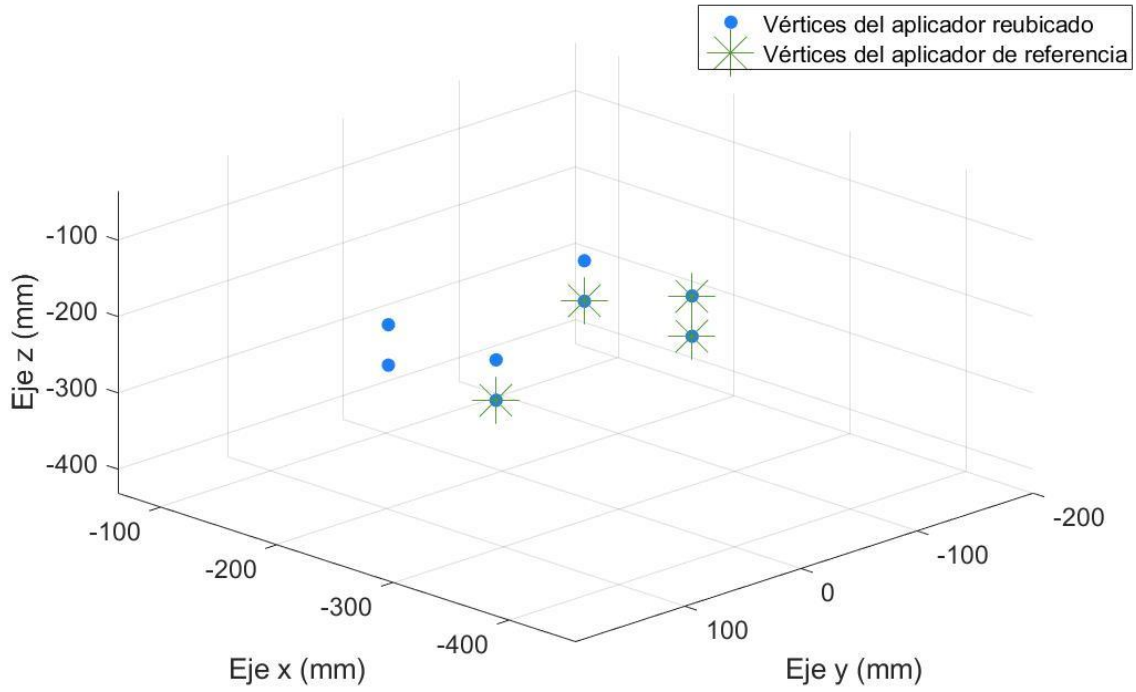


Fig. 47. Comparación de los vértices de uno de los aplicadores laterales y los vértices del aplicador de referencia.

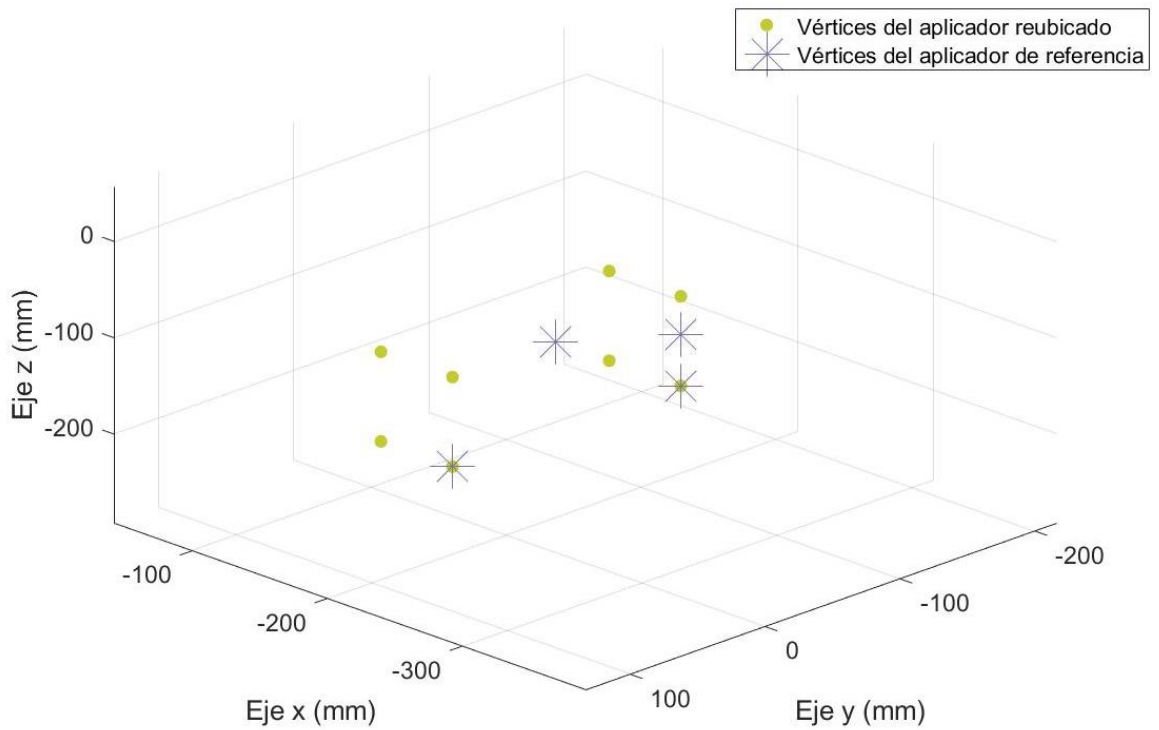


Fig. 48. Comparación de los vértices del aplicador central y los vértices del aplicador de referencia.

Tabla 13. Ángulos de los vértices del aplicador central con respecto a los vértices del aplicador de referencia en los diferentes planos.

	Ángulo en el plano xy (°)		% de error	Ángulo en el plano xz (°)		% de error	Ángulo en el plano yz (°)		% de error
	Calculado	Real		Calculado	Real		Calculado	Real	
Vector 1 (distancia AB)	NaN	NaN	NaN	90	90	0	NaN	0	NaN
Vector 2 (distancia AD)	NaN	0	0	90	90	0	NaN	NaN	NaN
Vector 3 (distancia AF)	0	0	0	NaN	NaN	NaN	0	0	0

En la Tabla 13, además de la presencia de las letras NaN, se puede apreciar que el % de error es 0, ya que tanto el ángulo calculado y el ángulo real proporcionado desde el diseño en *software* CAD son iguales. Y al igual que en el caso de la Etapa 1, se hace caso omiso de los ángulos obtenidos en los planos *xy* & *yz*.

8.3.2.6 Exportación de las coordenadas (x, y, z) de los vértices de los aplicadores al software de modelado

El siguiente paso en la Etapa 2 después del cálculo de la coordenada θ , es la exportación de las coordenadas (x, y, z) de cada uno de los aplicadores al *software* de modelado. Para llevar a cabo esta fase, solo es necesario transcribir los datos de la variable *coordenadas_exportacion* de *Matlab* al *software* de modelado. En este *software* se utilizan prismas rectangulares para reconstruir los aplicadores utilizando las coordenadas de los vértices detectados, a través de la geometría denominada hexaedro. Esto da como resultado una reconstrucción “ideal” de los aplicadores, ya que se eliminan las pestañas del diseño CAD, además del espacio interior de los aplicadores. Dentro del *software* de modelado para generar el hexaedro, este pide como datos de entrada las coordenadas (x, y, z) de cada uno de los vértices que formaran al hexaedro (coordenadas x, y, z provenientes de *Matlab*). En la Fig. 49 se

muestra una comparación entre los vértices reubicados de los aplicadores en *Matlab* y los vértices desplegados en el *software* de modelado.

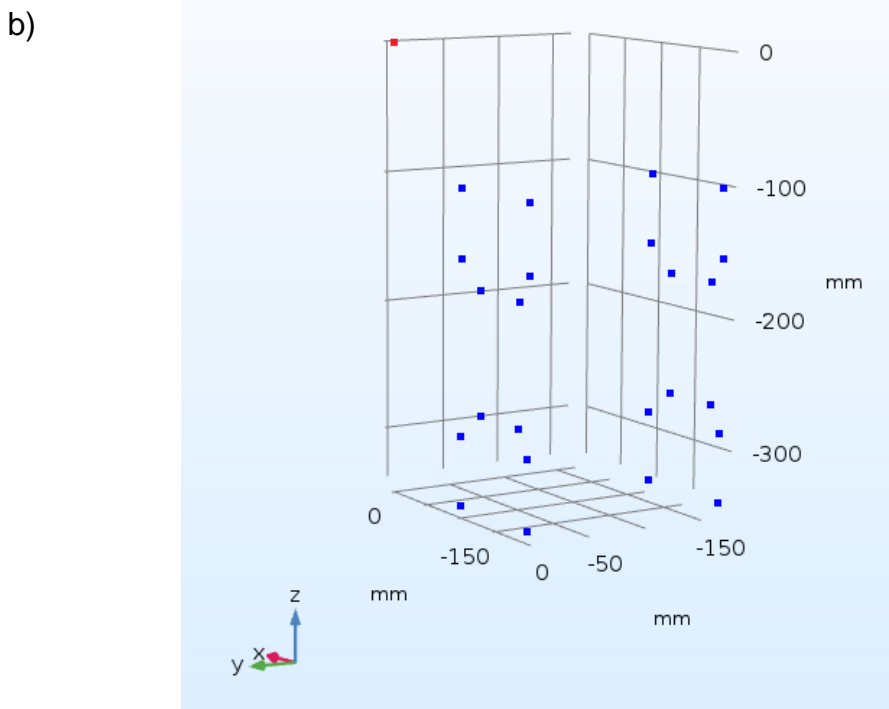
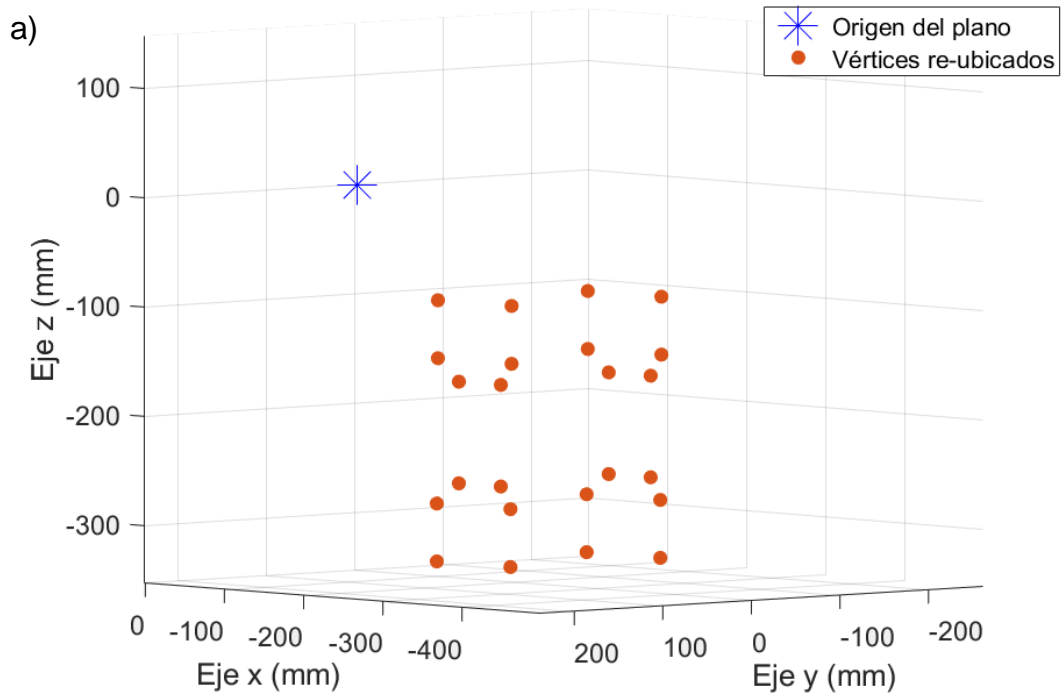


Fig. 49. Comparación entre los vértices reubicados en *Matlab* (a) y los vértices desplegados en el *software* de modelado (b).

La Fig. 49b muestra el punto de origen (0,0,0) en color rojo que coincide con el asterisco presente en la Fig. 49a; se observa que la orientación de los ejes es la misma en ambos casos y el rango de valores en los ejes difiere un poco por la forma de graficar de cada uno de los *softwares*. Sin embargo, al ser una transcripción de datos (vértices) las posiciones son exactamente iguales.

A modo de validación de la posición de los vértices reubicados en *Matlab* se midió la distancia existente entre ellos, tomando las siguientes consideraciones; se seleccionaron los vértices de dos aplicadores distintos cercanos entre sí tal y como se observa en la Fig. 50. También en esta figura se observa que se trabaja sobre el plano xz; esto se debe a que en este plano se localiza la superficie del cilindro sobre el cual se colocaron los aplicadores. Para medir la distancia entre los aplicadores, se alinearon los vértices seleccionados; esto se realizó igualando el valor de la coordenada x de ambos vértices. En este paso, es indistinto si se alinea el vértice de un aplicador o de otro. La consideración importante es que ambos valores en x de los vértices coincidan, para este caso en específico se alineo el vértice dos (vértices de

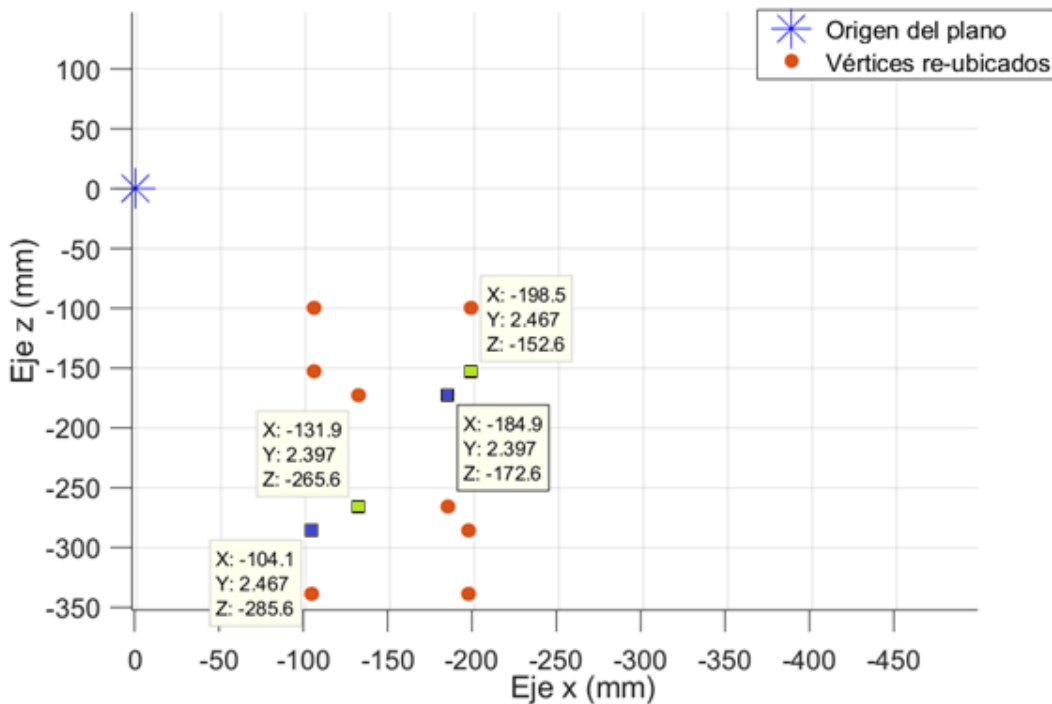


Fig. 50. Vértices seleccionados para la medición de distancia entre los aplicadores (plano xz).

color azul en la Fig. 50 y los vértices de color verde son los vértices 1). Finalmente, se calcula la distancia euclidiana existente entre ambos vértices. Primero, se mide la distancia entre el nombrado vértice *a* correspondiente al aplicador lateral y el vértice *b* que corresponde al aplicador central y después se mide la distancia entre el llamado vértice *a* del aplicador central y el vértice *b* del aplicador lateral. Los resultados obtenidos se observan en la Tabla 14.

Tabla 14. Distancia entre el diseño CAD del arreglo de aplicadores a partir de dos vértices.

Vértice a (mm)			Vértice b (mm)			Vértice b alineado (mm)			Distancia calculada (mm)	Distancia real (mm)	% de error
x	y	z	x	y	z	x	y	z			
-198.5	2.467	-152.6	-184.9	2.397	-172.6	-198.5	2.397	-172.6	20.0001	20	5X10 ⁻⁴
-131.9	2.397	-265.6	-104.1	2.467	-285.6	-131.9	2.467	-285.6	20.0001	20	5X10 ⁻⁴

De acuerdo con los resultados de la Tabla 14, se observa que la distancia entre los aplicadores no cambió en demasía por lo cual, es posible considerar la reconstrucción en el *software* de modelado como fiable ya que son exactamente los mismos datos en este *software* que los analizados en *Matlab*.

8.4 Etapa 3: Reconstrucción 3D de la emulación de un tratamiento de hipertermia

Una vez que se obtuvo la reconstrucción en el *software* de modelado del arreglo de aplicadores, se inició con la etapa final de este proyecto siguiendo el flujo de trabajo de la Fig. 8. Esta etapa consiste en realizar la reconstrucción 3D de los elementos de un tratamiento de hipertermia para obtener la posición de los aplicadores involucrados.

8.4.1 Construcción de los aplicadores propuestos [11]

En la Fig. 51 se muestran distintas vistas de un aplicador no funcional construido con MDF, así como sus medidas. Se construyeron en total 3 aplicadores utilizando una cortadora laser, de los cuales se utilizaron 2 para la emulación del tratamiento de hipertermia.

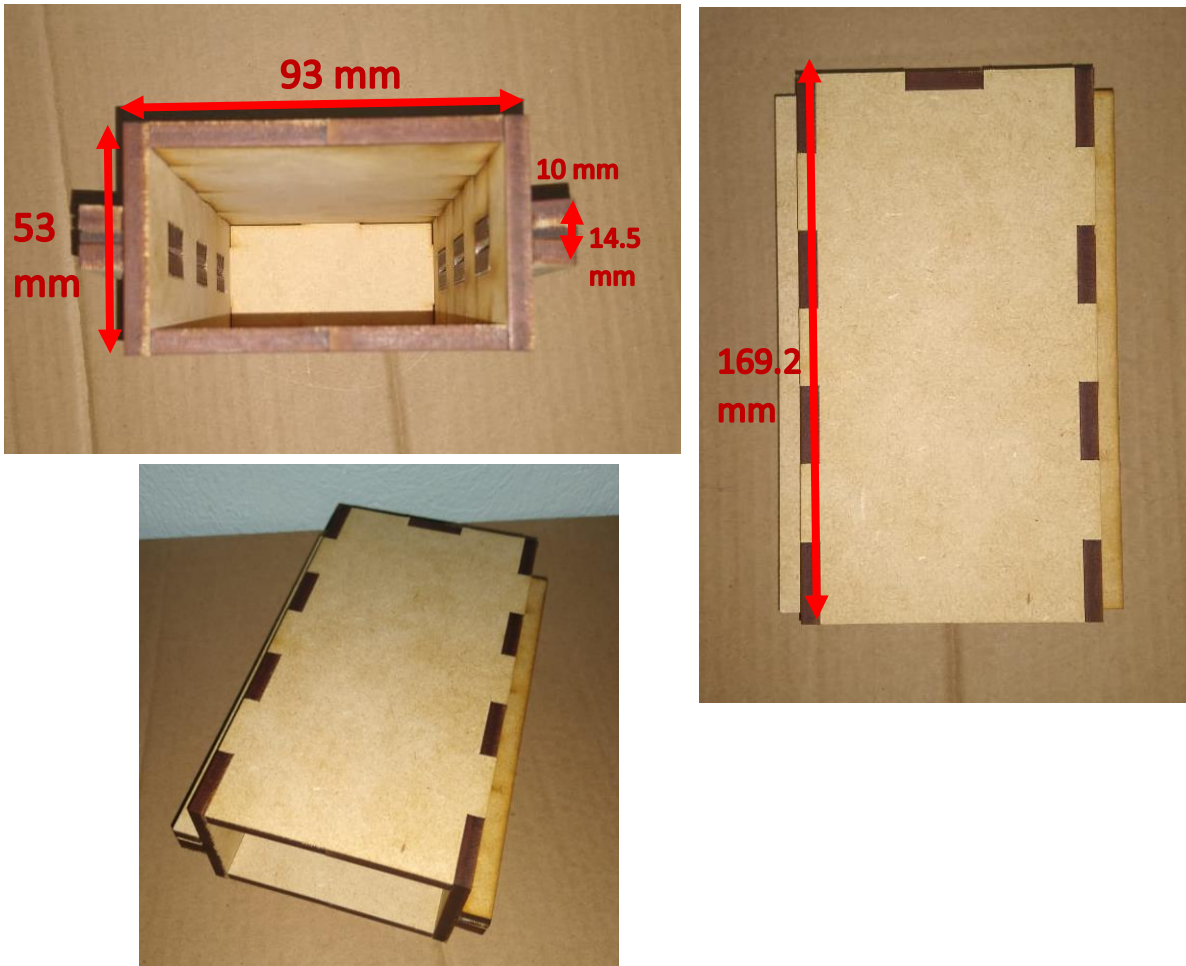


Fig. 51. Aplicador (guía de onda) no funcional construido con MDF.

8.4.2 Ubicación de los aplicadores sobre el maniquí

Como se mencionó en la sección 7.4.2, para emular el tratamiento de hipertermia, se colocaron dos aplicadores (con una separación de 3 cm entre sí) en la zona de la pierna en el maniquí, para emular el tratamiento de hipertermia, tal y como se muestra en la Fig. 52.



Fig. 52. Aplicadores colocados sobre el maniquí para emular un tratamiento de hipertermia.

8.4.3 Toma de video de los aplicadores

Se realizaron grabaciones de distintos elementos para buscar los valores óptimos de duración, número de *frames* a utilizar, etc. La Tabla 15 muestra la información de los videos grabados para realizar la reconstrucción 3D de los distintos objetos filmados, dentro de esta tabla se resaltan los datos del video utilizado para realizar la reconstrucción 3D. Con ayuda del el *Freeware Free Video to JPG Converter*, se obtuvieron los *frames* del video para llevar a cabo la reconstrucción 3D con ayuda del *software COLMAP*.

Tabla 15. Datos de los videos grabados de distintos objetos para su futura reconstrucción 3D.

Elementos a grabar	Duración del video (segundos)	Frames (Imágenes) totales	Distancia de grabación (cm)	Imágenes (frames) utilizadas en la reconstrucción
Un aplicador	30	~1800	30	200
Un aplicador	33	~1980	30	250
Dos aplicadores y maniquí	43	~2580	10 – 60	250
Dos aplicadores y maniquí	33	~1980	10 – 60	250
Tres aplicadores y maniquí	46	~2760	50 - 70	138
Tres aplicadores y maniquí	70	~4140	50 - 70	210

8.4.4 Reconstrucción 3D utilizando el algoritmo *Sfm* implementado en el software COLMAP [13]

Para realizar la reconstrucción 3D de la emulación del tratamiento de hipertermia, se utilizó el software COLMAP. En la Fig. 53 se muestra la configuración utilizada en la ventana de *Automatic reconstruction* para llevar a cabo esta fase. La Tabla 16 muestra los tiempos de reconstrucción de los objetos grabados (ver Tabla 15) en un ordenador con una memoria RAM de 8 GB, un procesador Core i5 novena generación y un sistema operativo de 64 bits. Dentro de la Tabla 16 se encuentra resaltada la información de la reconstrucción utilizada para el desarrollo de la Etapa 3. La Fig. 54

muestra la reconstrucción obtenida de dos aplicadores y el maniquí (resaltados en la Tabla 15 y 16).

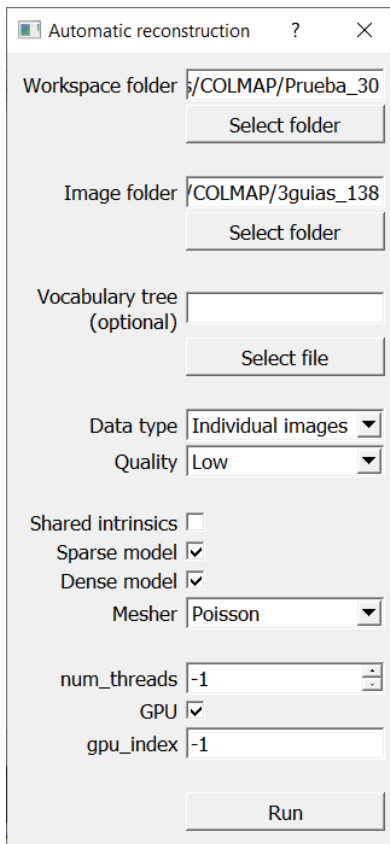


Fig. 53. Configuración utilizada para realizar la reconstrucción 3D

Tabla 16. Datos de los videos grabados de distintos objetos para su futura reconstrucción 3D.

Elementos grabados	Calidad de reconstrucción	Tipo de dato	Tiempo de reconstrucción (minutos)	Resultados obtenidos
Un aplicador	Baja	Frames de video	~ 47	Reconstrucción aceptable, geometría del aplicador completa
Un aplicador	Baja	Frames de video	~ 44	El aplicador reconstruido no

				tuvo su geometría completa
Dos aplicadores y maniquí	Baja	Imágenes individuales	~ 26	Reconstrucción aceptable, geometrías de los aplicadores completas
Dos aplicadores y maniquí	Baja	<i>Frames</i> de video	~ 30	Reconstrucción aceptable, deformación en la superficie de los aplicadores. Aplicadores reconstruidos completos,
Tres aplicadores y maniquí	Baja	Imágenes individuales	~ 20	deformación en las aristas de los aplicadores. Aplicadores reconstruidos incompletos,
Tres aplicadores y maniquí	Baja	Imágenes individuales	~ 22	deformación en la superficie de los elementos reconstruidos.

Se realizaron distintas reconstrucciones con diferentes cantidades de imágenes de la misma grabación, variando los parámetros de *COLMAP*. Sin embargo, un mayor número de imágenes implicaba un mayor tiempo de procesamiento (llegando a tardar 6 horas con 800 imágenes) y esto no aseguraba una buena calidad en la reconstrucción 3D, ya que se presentaron distorsiones en las figuras debido al solapamiento de las imágenes. Por lo tanto, después una cantidad determinada de

pruebas se configuró *COLMAP* tal y como se muestra en la Fig. 53, usando una cantidad de imágenes que van desde las 200 a las 300, ya que con estas especificaciones se obtuvieron tiempos de procesamiento reducidos (>30 min) y reconstrucciones 3D sin deformaciones exageradas ni gran cantidad de elementos aislados.

8.4.5 Edición de la reconstrucción 3D

Como se observa en la Fig. 54, la reconstrucción 3D tiene varios elementos que no son de interés. Por lo tanto, y para facilitar su análisis, se realizó una adecuación de la reconstrucción 3D que va de acuerdo con el diagrama de flujo de la Fig. 22.



Fig. 54. Reconstrucción 3D de los aplicadores y el maniquí obtenida en COLMAP.

8.4.5.1 Eliminación de vértices duplicados, eliminación de elementos aislados y eliminación de elementos no deseados

La Fig. 55 muestra la reconstrucción 3D después de eliminar los vértices duplicados y los elementos aislados. Se muestran ambos pasos en una sola imagen ya que la eliminación de vértices duplicados no se hace evidente en la reconstrucción 3D.



Fig. 55. Reconstrucción 3D eliminando vértices duplicados y elementos aislados.

La Fig. 56 muestra la reconstrucción 3D después de eliminar los elementos no deseados, es decir, la reconstrucción resultante se enfoca directamente en la posición de los aplicadores sobre la pierna.



Fig. 56. Reconstrucción 3D eliminando elementos no deseados.

8.4.5.2 Escalamiento

La Fig. 57 muestra la reconstrucción ya escalada, y como prueba de ello se muestra la dimensión de una de las uniones de los aplicadores; como se mencionó en la sección 7.4.5, esta unión mide 24 mm. Después del escalamiento la reconstrucción obtenida se guarda en formato .stl sin ninguna otra modificación, este archivo es denominado el modelo completo.

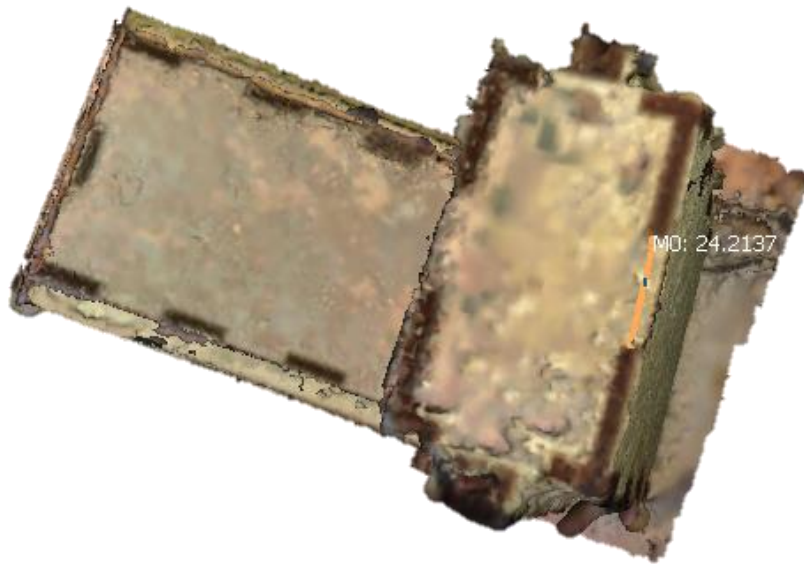


Fig. 57. Reconstrucción 3D escalada, tomando como referencia las uniones entre las caras de los aplicadores.

8.4.5.3 Reducción en el número de vértices del modelo

En la Fig. 58 se muestran distintas vistas de la reconstrucción 3D una vez que se han reducido el número de vértices con el fin de simplificar su tiempo de procesamiento con el algoritmo desarrollado (*coor_recons.mat*). La reconstrucción obtenida se guarda en formato .stl y este se denomina como modelo ideal. El número de vértices final del modelo ideal depende de dos cosas; el número de aplicadores involucrados y la deformación del modelo. Dependiendo del número de aplicadores el número final de vértices no puede ser menor al número de vértices pertenecientes a los aplicadores. Por ejemplo, si en la reconstrucción 3D están involucrados 3 aplicadores el número mínimo de vértices debería ser 24 (8 vértices por cada aplicador). Sin embargo, al realizar las pruebas correspondientes se encontró que reducir el número de vértices del modelo ideal a menor o igual que 24 vértices (siguiendo el ejemplo anterior) se encontró que el modelo ideal pierde la forma de los aplicadores. Es decir, los prismas

rectangulares (aplicadores) se deforman dando paso en algunos casos a figuras piramidales.

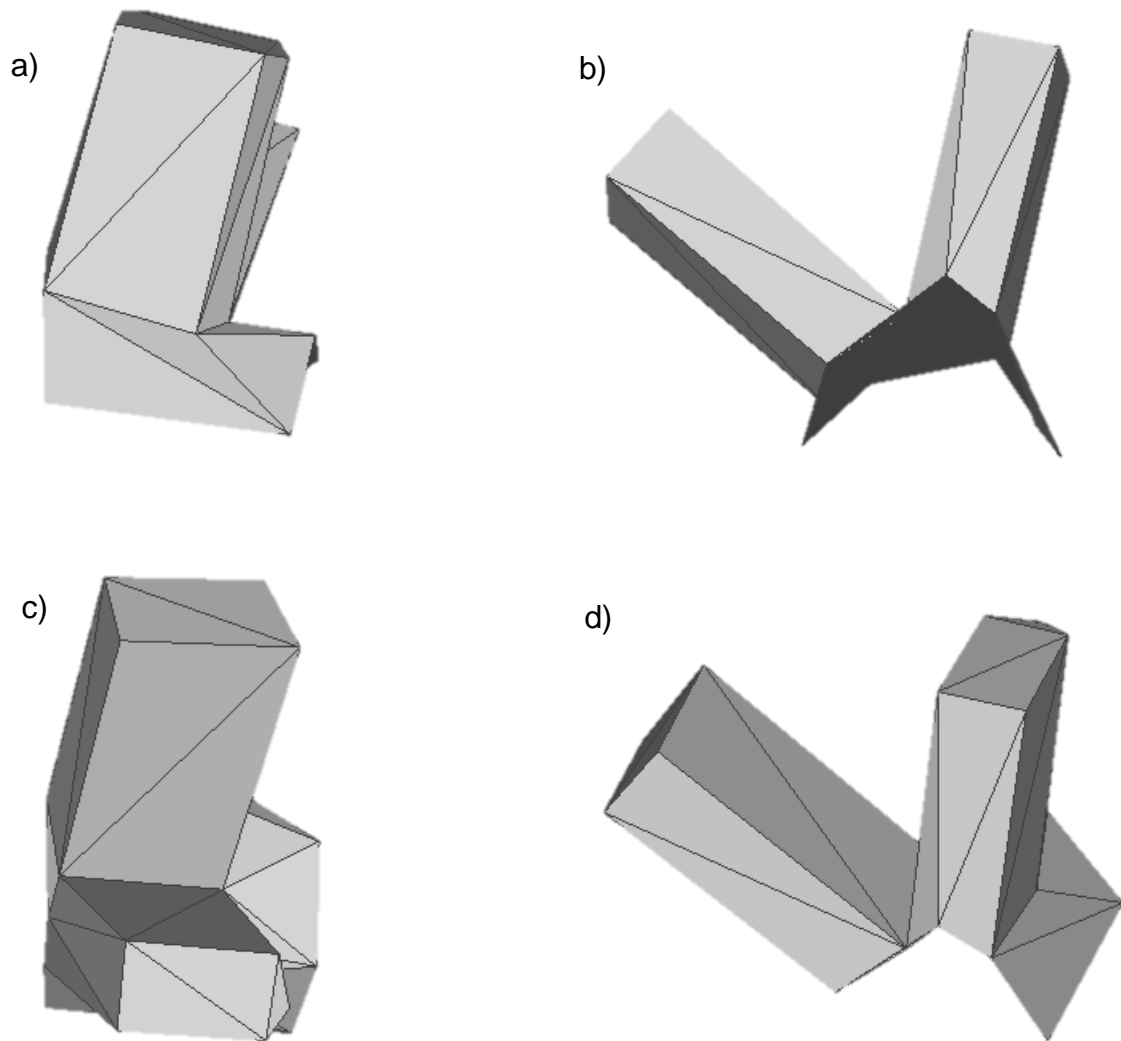


Fig. 58. Vistas del modelo ideal de la reconstrucción 3D de la emulación tratamiento de hipertermia, a) vista frontal, b) vista lateral, c) vista superior, d) vista trimétrica.

8.4.6 Obtención de la posición (x , y , z & θ) de los vértices de los aplicadores reconstruidos

Para la obtención de las coordenadas de los aplicadores, se utilizó el flujo de trabajo descrito en la subsección 7.4.6. El algoritmo *coor_recons.mat* es el encargado de llevar a cabo esta etapa.

8.4.6.1 Despliegue del modelo ideal y detección de los vértices superiores de los aplicadores

La Fig. 59 muestra el modelo ideal desplegado en *Matlab*, al igual que en la Etapa 1 y 2, se utilizó la función *stlread* [40], también se eliminaron los vértices duplicados con la función *unique*, aplicada a la variable *vertices*. Después de que el algoritmo *coor_recons.mat* despliega el modelo ideal, éste solicita al usuario seleccionar dos puntos en el espacio tridimensional cerca de las caras superiores de los aplicadores (entiéndase como caras superiores aquellas mostradas en la Fig. 24). En la Fig. 60 se muestran los puntos seleccionados por el usuario, así como los vértices encontrados en toda la figura y los vértices pertenecientes a las caras superiores (vértices superiores).

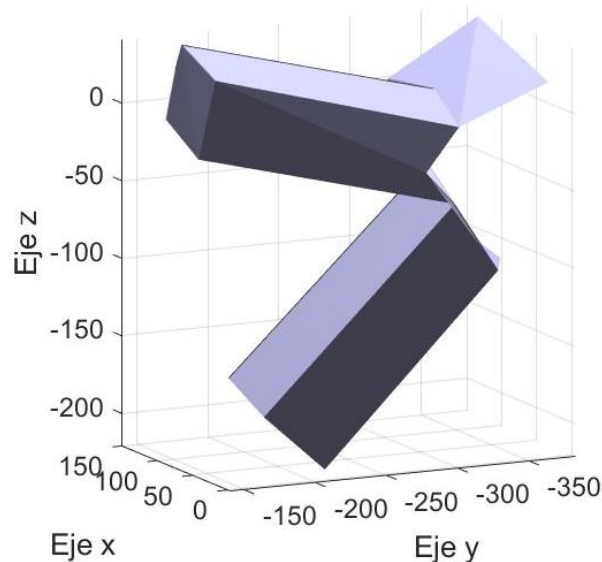


Fig. 59. Despliegue del modelo ideal de los aplicadores reconstruidos.

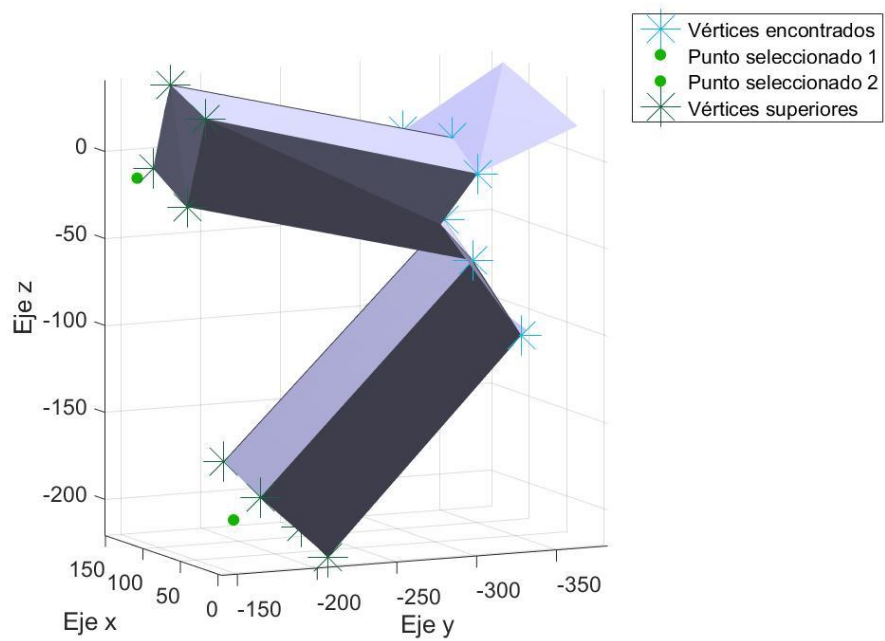


Fig. 60. Identificación de los vértices superiores de los aplicadores reconstruidos.

La razón principal de solicitar al usuario que seleccione puntos en el espacio 3D de las caras superiores de los aplicadores, es que, las pruebas realizadas con una detección automática no arrojaban resultados aceptables. Ya que en la mayoría de los casos no identificaba el total de los vértices deseados (8 en este caso). Entonces con la ayuda del usuario se detectan los 4 vértices más cercanos al punto seleccionado y con esto se agrupan los denominados vértices superiores.

8.4.6.2 Emparejamiento de los vértices superiores entre el modelo ideal y el modelo completo

Una vez que se identificaron los vértices superiores del modelo ideal, es necesario emparejarlos con los vértices presentes en el modelo completo. En la Fig. 61 se muestran los vértices superiores, los vértices emparejados y también se puede observar el perímetro de la cara superior de ambos aplicadores.

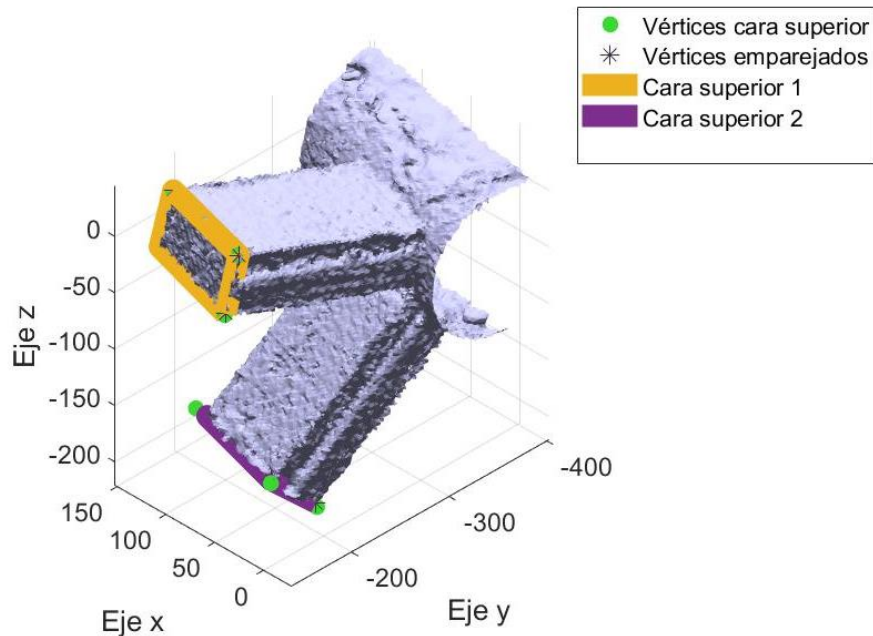


Fig. 61. Vértices emparejados en el modelo completo.

La razón principal para hacer la detección en el modelo ideal y no en el modelo completo directamente es la cantidad de datos analizar y el tiempo de procesamiento que eso lleva. Además de que al analizar el modelo completo, se identifican más de los 4 vértices de la cara superior, con lo cual sería necesario implementar más criterios de discriminación.

8.4.6.3 Estimación de los vértices de la cara inferior de los aplicadores

Después de emparejar los vértices superiores, el siguiente paso es la estimación de los vértices inferiores. En el modelo ideal de la reconstrucción los vértices inferiores no son evidentes, por lo cual no se realiza una detección de los vértices en ese modelo. En el modelo completo estos vértices tampoco son identificables ya que estos se “fusionaron” con al maniquí. Por lo cual la solución propuesta es estimar estos puntos usando los criterios expuestos en la subsección 7.4.6. La Fig. 62 muestra los vértices inferiores estimados, los vértices y las caras superiores de ambos aplicadores.

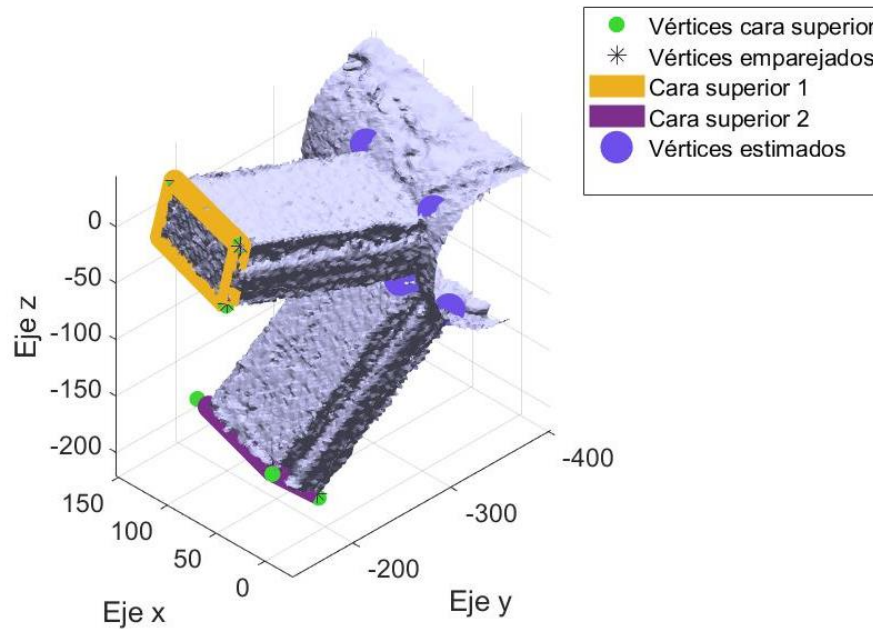


Fig. 62. Despliegue de los vértices inferiores (vértices estimados).

8.4.6.4 Selección del punto de referencia y reubicación de los vértices

Una vez que se tuvieron los vértices completos de cada uno de los aplicadores (vértices superiores e inferiores), el siguiente paso consiste en que el algoritmo *coor_recons.mat* solicita al usuario seleccionar un punto en el espacio tridimensional para reubicar todos los vértices en función de este punto, tal y como se realizó en las Etapas 1 y 2. La Fig. 63 muestra el punto seleccionado y la Fig. 64 muestra los vértices reubicados de acuerdo con el punto seleccionado.

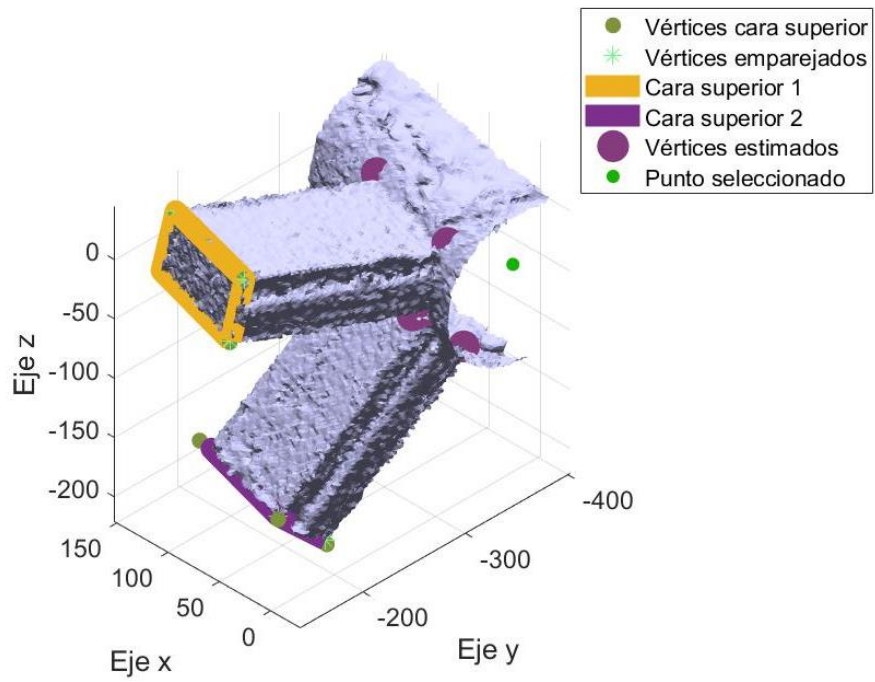


Fig. 63. Punto de referencia seleccionado por el usuario en el espacio tridimensional de la reconstrucción 3D.

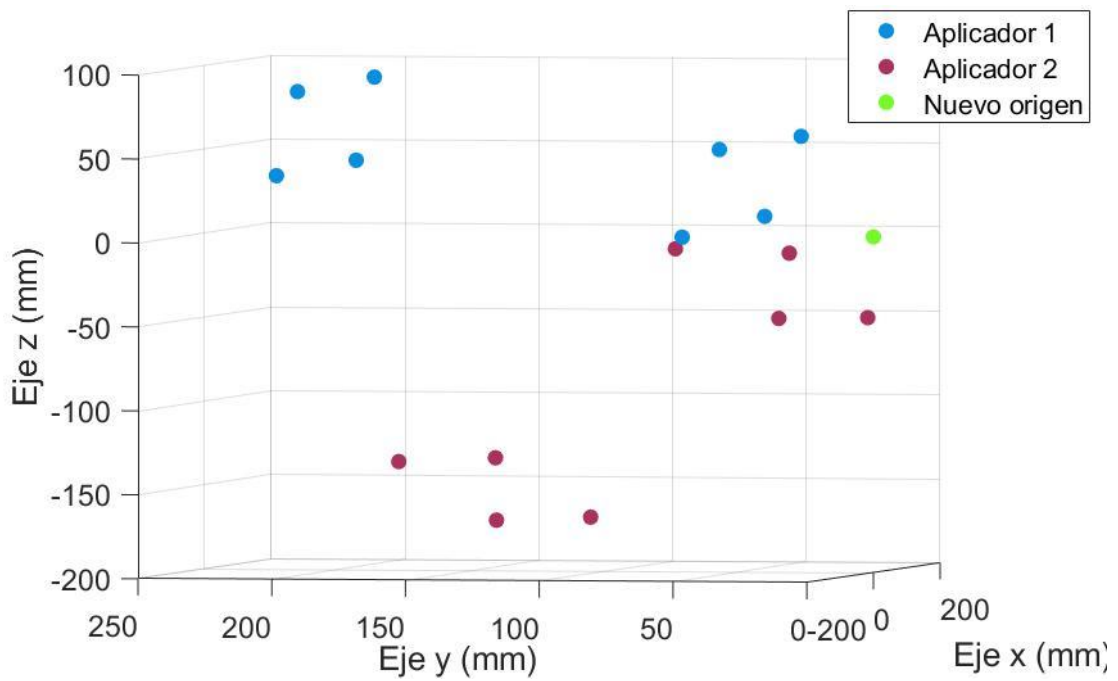


Fig. 64. Vértices reubicados de acuerdo con el punto seleccionado (nuevo origen).

8.4.6.5 Cálculo de los ángulos θ de los aplicadores

Una vez que los vértices de los aplicadores se reubicaron de acuerdo con el punto de referencia seleccionado por el usuario, el siguiente paso consiste en calcular los ángulos en los aplicadores tal y como se describe en la sección 7.2.2.1. La Fig. 65 muestra la variable *theta* desplegada en *Matlab*. Sin embargo, como se mencionó anteriormente esta coordenada (θ) es considerada de forma provisional como dato meramente informativo.

```
theta(:,:,1) =  
      NaN    5.7589    9.6051  
    12.3913    4.5178      NaN  
    13.5697      NaN    6.1675  
  
theta(:,:,2) =  
      NaN    14.2725    47.6269  
    15.6480    0.3670      NaN  
    12.9611      NaN    67.0679
```

Fig. 65. Ángulos de los aplicadores para cada uno de los planos.

8.4.6.6 Exportación de las coordenadas (x, y, z)

Para este paso se realizó el mismo procedimiento que en la Etapa 2, se generó un hexaedro en el software de modelado y se colocaron las coordenadas obtenidas en el algoritmo *coor_recons.mat*. En la Fig. 66 se muestra la comparación entre los vértices reubicados de los aplicadores en *Matlab* y los vértices reconstruidos en el *software* de modelado.

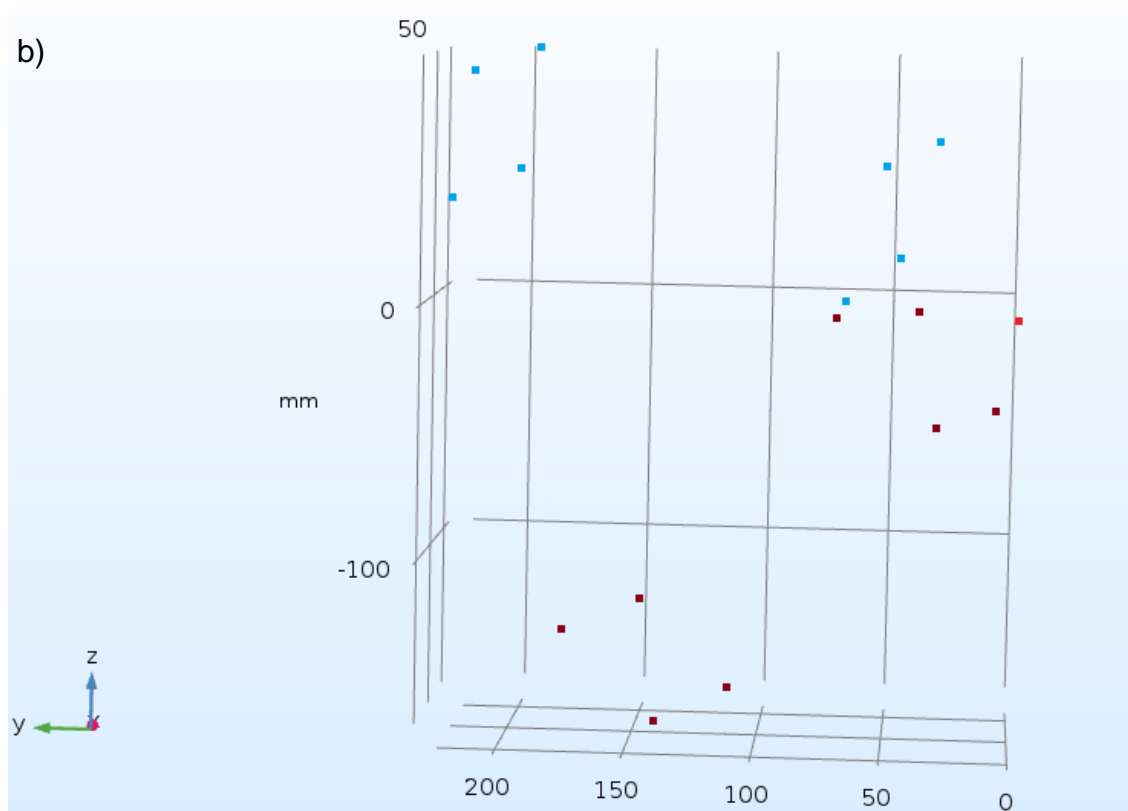
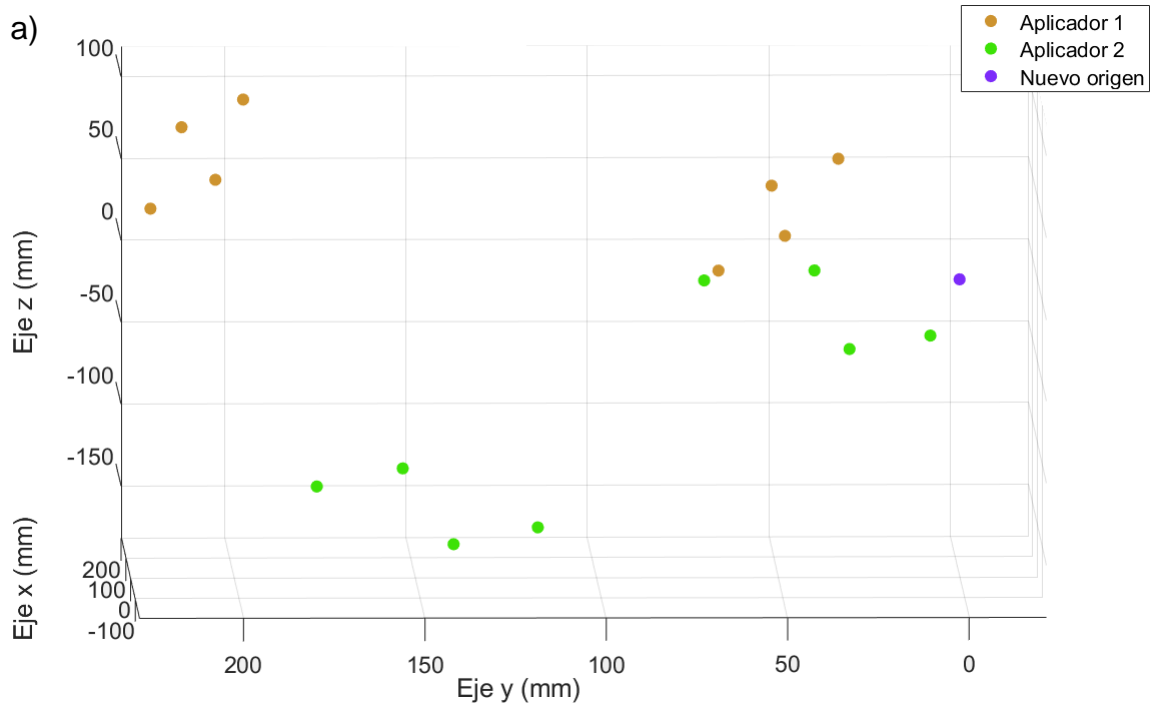


Fig. 66. Comparación entre los vértices reubicados de la reconstrucción 3D en *Matlab* (a) y los vértices desplegados en el *software* de modelado (b).

La Fig. 66 muestra que la orientación de los ejes es la misma en ambos casos, el rango de valores en los ejes difiere un poco por la forma de graficar de cada uno de los *softwares*. Sin embargo, al ser una transcripción de la posición de los vértices, las posiciones en ambos *softwares* son exactamente iguales. La Fig. 67 muestra los aplicadores reconstruidos en el *software* de modelado de acuerdo con las coordenadas de los vértices encontrados previamente.

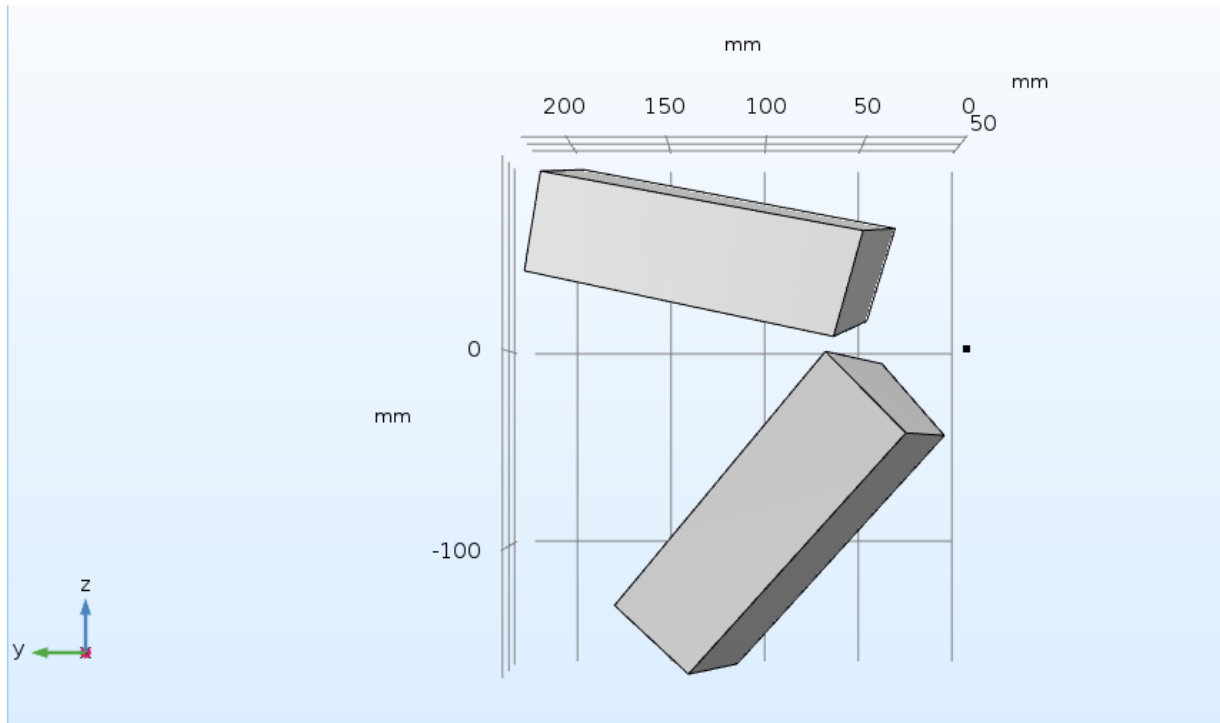


Fig. 27 Reconstrucción 3D en el *software* de modelado de los aplicadores fotografiados.

Como se mencionó previamente, los aplicadores se colocaron a una distancia aproximada de 30 mm entre sí. Para comprobar si esta distancia se mantuvo en la reconstrucción 3D, se calculó la distancia euclidiana entre los vértices más próximos de los aplicadores. La Fig. 67 muestra los vértices de los aplicadores y sus respectivas coordenadas (x , y , z).

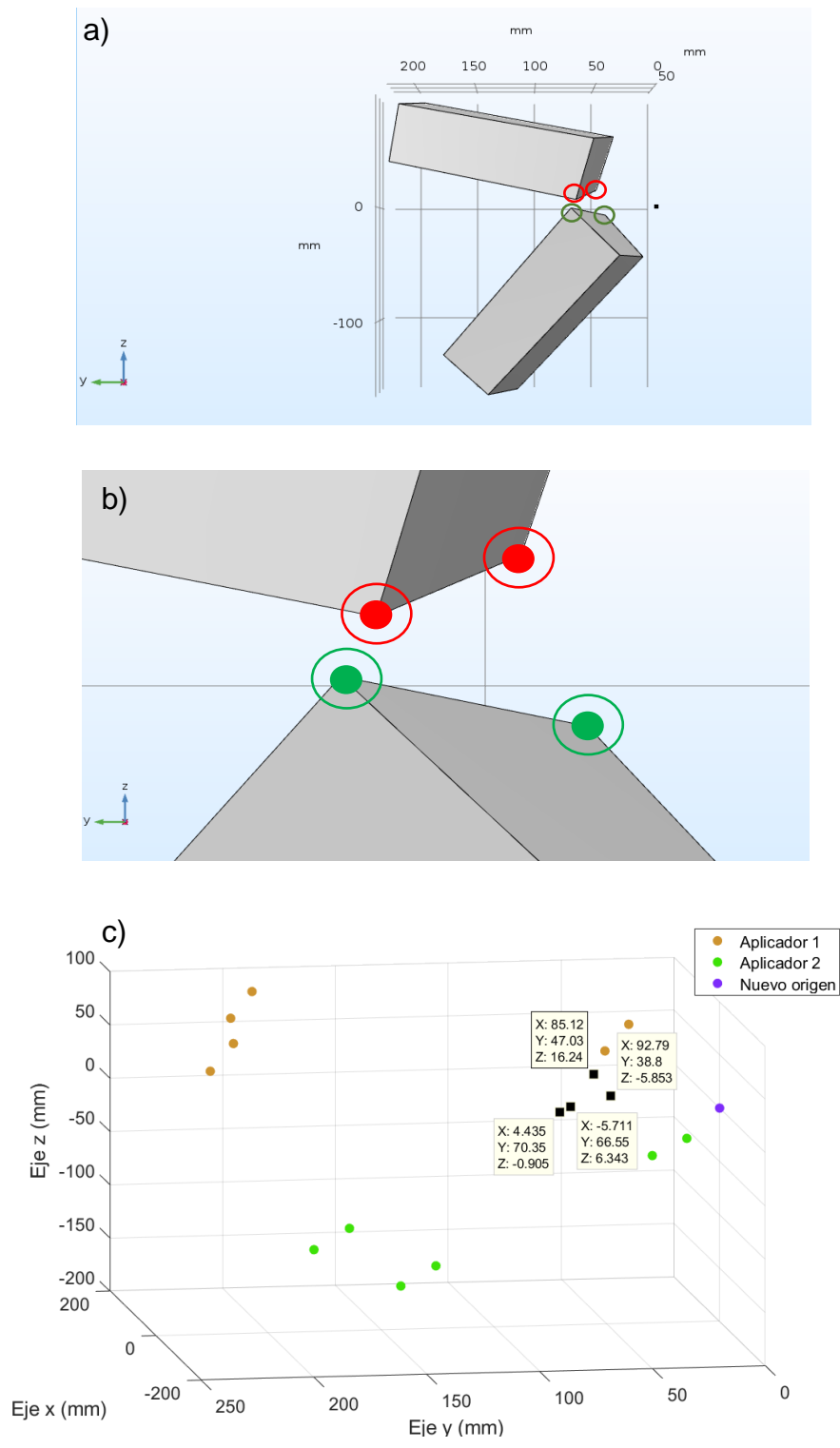


Fig. 68. Vértices utilizados para medir la distancia entre los aplicadores, a) vértices seleccionados en la reconstrucción realizada en el software de modelado, b) acercamiento a los vértices seleccionados, c) coordenadas de los vértices seleccionados

La Tabla 17 muestra la distancia calculada y el porcentaje de error entre la distancia de los vértices seleccionados (Fig. 67c). El porcentaje de error es de 56.55% y 17.35% en los pares de vértices analizados. Esto indica que la estimación de los vértices inferiores (los cuales fueron analizados) tiene un porcentaje de error bastante alto en uno de los casos siendo más del 50%.

Tabla 17. Distancia entre los aplicadores reconstruidos a partir de dos vértices.

Vértice a (mm)			Vértice b (mm)			Distancia calculada (mm)	Distancia real (mm)	% de error
x	y	z	x	y	z			
-5.711	66.55	6.343	4.435	70.35	-0.905	13.0351	30	56.55
85.12	47.03	16.24	92.79	38.8	-5.853	24.7924	30	17.35

9. Discusión

En la subsección 8.1 se comprobó de forma preliminar que el aplicador propuesto por C. Trujillo et al [11] puede ser usado en el tratamiento de tumores en la interfaz músculo-hueso al obtener temperaturas de hipertermia ($\sim 40.5^{\circ}\text{C}$) a una profundidad dentro de los tejidos de aproximadamente 32.35 mm cuando el aplicador es alimentado con una potencia de 100 W durante 90 min. Aunque aún es necesario validar los resultados obtenidos de forma experimental. Tomando en consideración que el espesor del bolus de agua, de la grasa y del músculo fue de 5 mm, 1 mm y 9 mm, respectivamente, para obtener esa temperatura a la profundidad mencionada. A pesar de que, se han realizado modelos similares por diferentes grupos de investigación, por ejemplo; F. Ondrej *et al.* [32] y S. Singh *et al.* [33]. En esos casos, a pesar de que se analizaron las distribuciones de temperatura sobre los tejidos, no se mencionó alguna referencia sobre el SWR, el cual es un parámetro importante para saber qué tan bien se está transfiriendo la energía. En la sección 8.1 se realizó un análisis del SWR cuando los espesores de la grasa y el músculo son variables. En el trabajo presentado por F. Ondrej *et al.*, en el año 2014 [32], se utilizaron más de 5 medios (bolus de agua, piel, grasa, músculo, hueso, tumor, etc.). Sin embargo, para analizar la distribución de temperatura en los medios, el espesor de todos ellos se consideró constante. En el presente trabajo se implementó un estudio paramétrico cambiando los espesores de los tejidos (músculo y grasa). En el caso de [33], aunque se consideraron 5 medios (piel, grasa, tumor, músculo y un bolus de agua), las simulaciones se realizaron utilizando una antena de ranura *microstrip* y también se consideró que el tumor se localiza cerca de la superficie (en la interfaz grasa-músculo).

Una vez comprobada la viabilidad en el uso de los aplicadores propuestos por C. Trujillo *et al.* [11] en el tratamiento de tumores en la interfaz músculo-hueso, en la subsección 8.2 se describen los resultados de la Etapa 1. En esta Etapa, la obtención de las coordenadas (x , y , z & θ) del aplicador es una tarea relativamente sencilla para el algoritmo desarrollado (*ext_coor_apli.mat*). Ya que se está trabajando con un diseño CAD, esto implica que sus dimensiones no tienen gran discrepancia con las medidas

originales, permitiendo usar tolerancias pequeñas (5×10^{-3} mm) para la identificación de vértices.

En el trabajo realizado por A. Tzitzis *et al.* [31], la forma en que obtuvieron las coordenadas (x , y , z) de un *tag* fue a través de la distancia euclidiana. Esta implementación se retomó en la Etapa 1, 2 y 3 para discriminar aquellos vértices cuya distancia euclidiana coincidiera con las dimensiones de los aplicadores. En [31] no está dentro de sus resultados la obtención de ángulo de posición, por lo cual en el presente trabajo se propuso crear un objeto de referencia que comparta con el objeto original un punto en común y partir de esto calcular el ángulo existente entre el objeto original y el de referencia.

En la subsección 8.3 se describen los resultados obtenidos en la Etapa 2. Las principales diferencias entre la Etapa 1 y la Etapa 2 (subsección 8.2 y 8.3 respectivamente) son que en la Etapa 2 analizó un arreglo de 3 aplicadores en lugar de solo 1 (tal y como se hizo en la Etapa 1) y además de la adición de la reconstrucción 3D de los aplicadores en un *software* de modelado. Para llevar a cabo esta reconstrucción se utilizó la información obtenida en el algoritmo *ext_coor_apli.mat* (coordenadas x , y , z de cada uno de los vértices de cada aplicador); sin embargo, los aplicadores reconstruidos son representados en el *software* de modelado por prismas rectangulares, propiciando la deformación de la geometría original de los aplicadores al no considerar las pestañas laterales de cada aplicador, así como el espacio interior de cada uno de ellos.

La Etapa 3 puede ser dividida en dos fases, la fase de la reconstrucción 3D de los elementos involucrados en un tratamiento de hipertermia (aplicadores y maniquí) y la obtención de las coordenadas de los aplicadores de dicha reconstrucción. De la primera fase se puede concluir que con el ordenador disponible para el desarrollo de este trabajo (memoria RAM de 8 GB, un procesador Core i5 novena generación y un sistema operativo de 64 bits) se obtuvieron reconstrucciones 3D sin deformaciones exageradas en la superficie de los elementos a analizar, manejando una calidad baja

en la configuración de *COLMAP*. También se obtuvieron tiempos de procesamiento relativamente pequeños (>30 min). En cuanto a la obtención de las coordenadas de los aplicadores, resultó complicado trabajar con toda la información (vértices) de la reconstrucción del modelo completo, ya que este modelo consta al menos de 40,000 vértices y dadas las condiciones de discriminación se detectaban vértices que no eran de utilidad. Por lo cual, se implementó el uso de un modelo ideal de la reconstrucción para reducir el número de datos a analizar determinar y detectar solo los 8 vértices por aplicador que son de interés. Sin embargo, obtener los 8 vértices por cada aplicador en un solo paso resultó complicado por la geometría de la reconstrucción en sí, ya que una de las caras de los aplicadores (denominada cara inferior) se fusiona con la pierna haciendo que los vértices de esa cara se pierdan. Debido a esto, se propuso dividir la determinación de los 8 vértices de cada aplicador en varias etapas, la primera de ellas involucra la intervención del usuario, quien selecciona un punto en el espacio 3D cercano a la cara superior del aplicador con el fin de facilitar la detección de los 4 vértices superiores. La siguiente etapa corresponde al emparejamiento de los vértices detectados (vértices superiores) del modelo ideal y los vértices del modelo completo, posterior a esto se estiman los vértices inferiores aquellos que se encuentran pegados al maniquí. Una vez que se tienen los 8 vértices por aplicador el flujo de trabajo es similar al usado en la Etapa 2, incluyendo la reconstrucción de los aplicadores en el *software* de modelado.

La diferencia entre el trabajo publicado por Drizdal *et al.* en el 2012 [8] y el presente trabajo es que se obtuvieron tiempos de reconstrucción menores (>30 min). Sin embargo, la cantidad de fotografías utilizadas para realizar la reconstrucción es mayor, mientras que en [8] utilizaron solo 10 fotografías en el presente trabajo se usan 200 – 300 fotografías para obtener una calidad óptima de reconstrucción, además de que el número de objetos (aplicadores) a reconstruir también es menor, considerando que en este trabajo solo se reconstruyeron 2 aplicadores y en [8] se reconstruyeron 6 de ellos. Otro punto importante es que en este trabajo se obtuvo la información detallada de cada uno de los vértices de los aplicadores (x , y , z) en función de un punto de referencia arbitrario que puede ser modificado de acuerdo con las necesidades del

usuario. Como dato adicional, en este trabajo se obtiene la coordenada θ de cada uno de los aplicadores, este dato en específico no se consideró en el trabajo realizado por Drizdal *et al.* [8].

10. Conclusiones y perspectivas

Los resultados obtenidos en la subsección 8.1 cumplen con el primer objetivo particular planteado, el cual es evaluar la factibilidad del aplicador desarrollado por C. Trujillo et al [11]. De este objetivo se tienen como conclusión que es viable el uso del aplicador propuesto en el tratamiento de tumores en la interfaz músculo-hueso, que se encuentran a una profundidad no mayor a 4 mm, utilizando un solo aplicador.

En aquellos objetivos particulares propuestos cuya finalidad es la obtención de un diseño CAD de un aplicador o un arreglo de ellos, se cumplen en las subsecciones 8.2 y 8.3 respectivamente, obteniendo archivos en formato .stl para su análisis en *Matlab*. En estas mismas secciones se obtienen las coordenadas (x , y , z & θ) de dichos diseños, en el caso del diseño CAD de un solo aplicador se obtuvo un error porcentual de 1.6% en la coordenada θ . Para el caso del arreglo de aplicadores se obtuvo un error porcentual de 5×10^{-4} entre la distancia de los aplicadores y la posición calculada.

Con respecto al objetivo general de esta tesis presentado en la sección 5, “obtener la posición (x , y , z & θ) de c/u de los aplicadores propuestos en un tratamiento terapéutico de hipertermia, a través de una reconstrucción 3D, a partir en fotografías bidimensionales de ellos” en este trabajo se cumple con dicho objetivo al conseguir las coordenadas (x , y , z & θ) de un sistema de dos aplicadores reconstruidos virtualmente a través de una serie de fotografías o imágenes proporcionadas por la grabación de un video de este sistema. Las coordenadas (x , y , z) son las únicas utilizadas para generar una reconstrucción 3D en el *software* de modelado de acuerdo con la forma en que se ingresan las coordenadas de cada uno de los aplicadores. La orientación de estos se da de forma automática, haciendo que la obtención de la coordenada θ sea por el momento únicamente informativa.

Como una perspectiva en trabajos futuros se puede evaluar el comportamiento del aplicador cuando se varía la frecuencia de operación (dentro del rango permitido de acuerdo con C. Trujillo *et al.*), poniendo especial atención en la temperatura máxima

alcanzada, profundidad de penetración y la cantidad de energía transferida. También sería conveniente realizar un análisis del comportamiento de un arreglo de aplicadores para inspeccionar la interacción de la energía electromagnética proporcionada por cada guía de onda, buscando alcanzar una mayor profundidad de penetración. También sería necesario analizar la cantidad de energía transferida y las temperaturas alcanzadas en los tejidos.

La reconstrucción obtenida en el *software* de modelado puede ser utilizada para realizar una simulación de la interacción de la energía electromagnética proporcionada por cada una de las guías de onda reconstruidas y conocer sus efectos sobre un sistema multicapa que emule los tejidos del cuerpo humano durante un tratamiento de hipertermia, similar al realizado en la subsección 8.1.

Otra opción de mejora de este trabajo, podría ser la edición o adecuamiento de los *frames* (imágenes) obtenidos del video para agilizar el procesamiento para la reconstrucción 3D o para mejorar la calidad de reconstrucción al reducir o eliminar elementos que no son de interés (como por ejemplo el fondo de la imagen).

En cuanto a la reconstrucción 3D, podrían explorarse otras configuraciones de la reconstrucción automática si se cuenta con ordenadores más potentes (mayor memoria RAM, tarjeta gráfica independiente, procesador más reciente, etc.), con el fin de buscar mejor calidad en las reconstrucciones (i. e. eliminación de distorsiones en la superficie).

Otro aspecto por mejorar es la forma en que se detectan los vértices superiores cuando se trabaja con reconstrucciones 3D, para prescindir de la intervención del usuario. Buscando que la obtención de las coordenadas sea de forma automática. Finalmente, es necesario mejorar la estimación de los vértices inferiores en la reconstrucción 3D del sistema de tratamiento de hipertermia, ya que el porcentaje de error es bastante grande (<50%) en algunos casos. Esto podría llevarse a cabo poniendo condiciones de discriminación más rigurosas reduciendo la tolerancia, por ejemplo. En este trabajo

la estimación de estos vértices se basa en la selección de algún dato (vértice) de la reconstrucción 3D de acuerdo con los criterios mencionados en la sección 7.4.6. Por lo cual, otro camino a seguir podría ser la estimación de los vértices sin tomar en consideración los vértices de la reconstrucción 3D, sino que simplemente que estos vértices cumplan con los criterios de discriminación y con esto crearlos.

11. Resultados académicos

Publicación en congreso internacional como primer autor

D. G. Serrano-Díaz, C. J. Trujillo-Romero, A. Vera, and L. Leija, **“Effect of the Water Bolus and Tissue Thickness over the Heat Distribution Generated by a RF Applicator Used as an Auxiliar to Treat Bone Tumors,”** Pan Am. Heal. Care Exch. PAHCE, vol. 2021-May, no. 2011, 2021, doi: 10.1109/GMEPE/PAHCE50215.2021.9434838.

Referencias

- [1] "Cáncer", Organización Mundial de la Salud, 2020. [Online]. Available: <https://www.who.int/topics/cancer/es/>. [Accessed: 29- Oct- 2020]
- [2] "Cancer - PAHO/WHO | Pan American Health Organization", Paho.org, 2020. [Online]. Available: <https://www.paho.org/en/topics/cancer>. [Accessed: 10- Nov- 2020]
- [3] "Cancer", Who.int. [Online]. Available: https://www.who.int/health-topics/cancer#tab=tab_1. [Accessed: 10- Nov- 2020]
- [4] American Cancer Society. Facts & Figures 2020. American Cancer Society. Atlanta, Ga. 2020.
- [5] "Cáncer de huesos - Síntomas y causas - Mayo Clinic", Mayoclinic.org, 2020. [Online]. Available: <https://www.mayoclinic.org/es-es/diseases-conditions/bone-cancer/symptoms-causes/syc-20350217>. [Accessed: 29- Oct- 2020]
- [6] C. J. Trujillo-Romero, G. Rico-Martínez, and J. Gutiérrez-Martínez, "Thermal ablation: an alternative to bone cancer." *Investig. en Discapac.*, vol. 7, pp. 36–47, 2018.
- [7] C. J. Trujillo-Romero, L. Leija, and A. Vera, "FEM modeling for performance evaluation of an electromagnetic oncology deep hyperthermia applicator when using monopole, inverted T, and plate antennas," *Prog. Electromagn. Res.*, vol. 120, no. January, pp. 99–125, 2011, doi: 10.2528/PIER11071809.
- [8] T. Drizdal, M. M. Paulides, M. Linthorst, and G. C. Van Rhoon, "Reconstruction of applicator positions from multiple-view images for accurate superficial hyperthermia treatment planning," *Phys. Med. Biol.*, vol. 57, no. 9, pp. 2491–2503, 2012, doi: 10.1088/0031-9155/57/9/2491.
- [9] S. Gao, M. Zheng, X. Ren, Y. Tang, and X. Liang, "Local hyperthermia in head and neck cancer: Mechanism, application and advance," *Oncotarget*, vol. 7, no. 35, pp. 57367–57378, 2016, doi: 10.18632/oncotarget.10350.
- [10] H. P. Kok and J. Crezee, "Hyperthermia treatment planning: Clinical application and ongoing research," 14th Eur. Conf. Antennas Propagation, EuCAP 2020, 2020, doi: 10.23919/EuCAP48036.2020.9135602.

- [11] C. Trujillo Romero, "Propuesta de desarrollo de un sistema aplicador de campo electromagnético para terapia en hipertermia oncológica", Doctorado, Centro de investigación y de estudios avanzados del Instituto Politécnico Nacional, 2011.
- [12] H. P. Kok, P. Wust, P. R. Stauffer, F. Bardati, G. C. Van Rhoon, and J. Crezee, "Current state of the art of regional hyperthermia treatment planning: a review," *Radiat. Oncol.*, pp. 1–14, 2015, doi: 10.1186/s13014-015-0503-8.
- [13] J. L. Schonberger and J. M. Frahm, "Structure-from-Motion Revisited," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 4104–4113, 2016, doi: 10.1109/CVPR.2016.445.
- [14] "Understanding Bone Cancer | American Cancer Society", *Cancer.org*, 2021. [Online]. Available: <https://www.cancer.org/cancer/bone-cancer/about/what-is-bone-cancer.html>. [Accessed: 01- Oct- 2021].
- [15] "¿Qué es el cáncer?", *Cancer.org*, 2020. [Online]. Available: <https://www.cancer.org/es/cancer/aspectos-basicos-sobre-el-cancer/que-es-el-cancer.html>. [Accessed: 29- Oct- 2020]
- [16] "Treating Bone Cancer", *Cancer.org*, 2020. [Online]. Available: <https://www.cancer.org/cancer/bone-cancer/treating.html>. [Accessed: 18- Nov- 2020]
- [17] "Hyperthermia in Cancer Treatment", *National Cancer Institute*, 2011. [Online]. Available: <https://www.cancer.gov/about-cancer/treatment/types/surgery/hyperthermia-fact-sheet>. [Accessed: 11- Nov- 2020]
- [18] Dr. Singh, "Hyperthermia", *Slideshare.net*, 2015. [Online]. Available: <https://www.slideshare.net/parneetsingh1217/hyperthermia-46322264>. [Accessed: 18- Nov- 2020]
- [19] D. Deás Yero, F. Gilart González, A. Raizer, "Hipertermia Electromagnética para el Tratamiento del Cáncer," no. January, pp. 33–34, 2013, doi: 10.1007/978-3-642-21198-0.
- [20] G. Baroni, G. Ferrigno, R. Orecchia, and A. Pedotti, "Real-time opto-electronic verification of patient position in breast cancer radiotherapy," *Comput. Aided Surg.*, vol. 5, no. 4, pp. 296–306, 2000, doi: 10.1002/1097-0150(2000)5.

- [21] B. Prasad, J. K. Kim, and S. Kim, "Role of simulations in the treatment planning of radiofrequency hyperthermia therapy in clinics," *J. Oncol.*, vol. 2019, 2019, doi: 10.1155/2019/9685476.
- [22] N. Grandón-Pastén, D. Aracena-Pizarro, and C. L. Tozzi, "Reconstrucción De Objeto 3D a Partir De Imágenes Calibradas," *Ingeniare. Rev. Chil. Ing.*, vol. 15, no. 2, pp. 158–168, 2007, doi: 10.4067/s0718-33052007000200006.
- [23] M. Aharchi and M. Ait Kbir, "A Review on 3D Reconstruction Techniques from 2D Images," pp. 510–522, 2020, doi: 10.1007/978-3-030-37629-1_37.
- [24] Q. Wei, "Converting 2D to 3D: A survey," *Int. Conf. Page*, vol. 7, no. December, p. 37, 2005, [Online]. Available: <http://ict.ewi.tudelft.nl/pub/ben/Research Assignment Qing Qing Wei - A Survey of 2D to 3D Conversion Algorithms.pdf>.
- [25] H. Ham and J. Wesley, "Computer vision based 3D reconstruction : A review," vol. 9, no. 4, pp. 2394–2402, 2019, doi: 10.11591/ijece.v9i4.pp2394-2402.
- [26] T. B. Phan, Di. H. Trinh, D. Lamarque, D. Wolf, and C. Daul, "Dense Optical Flow for the Reconstruction of Weakly Textured and Structured Surfaces: Application to Endoscopy," *Proc. - Int. Conf. Image Process. ICIP*, vol. 2019-September, pp. 310–314, 2019, doi: 10.1109/ICIP.2019.8802948.
- [27] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion," *Acta Numer.*, vol. 26, pp. 305–364, 2017, doi: 10.1017/S096249291700006X.
- [28] R. Tomás, A. Riquelme, M. Cano, and A. Abellán, "Structure from Motion (SfM): una técnica fotogramétrica de bajo coste para la caracterización y monitoreo de macizos rocosos.," *10o Simp. Nac. Ing. Geotécnica, A Coruña, España*, vol. 1, no. 1, pp. 209–216, 2016.
- [29] J. Crezee, R. Zweije, and H. P. Kok, "Loco-regional Hyperthermia Delivery: Patient-specific set-up Procedures for Treatment Optimisation," *14th Eur. Conf. Antennas Propagation, EuCAP 2020*, 2020, doi: 10.23919/EuCAP48036.2020.9135481.
- [30] M. Linthorst, T. Drizdal, H. Joosten, G. C. Van Rhoon, and J. Van Der Zee, "Procedure for creating a three-dimensional (3D) model for superficial

- hyperthermia treatment planning,” *Strahlentherapie und Onkol.*, vol. 187, no. 12, pp. 835–841, 2011, doi: 10.1007/s00066-011-2272-0.
- [31] A. Tzitzis, A. C. Raptopoulos, and T. Yioultsis, “A Real-Time Multi-Antenna SAR-based method for 3D localization of RFID Tags by a Moving Robot,” vol. 7281, no. c, pp. 1–15, 2021, doi: 10.1109/JRFID.2021.3070409.
- [32] F. Ondrej, M. Ilja, V. Lucie, and V. Jan, “Space distribution of SAR and temperature in human body model with tumor using waveguide applicator array,” *Eur. Microw. Week 2014 Connect. Futur. EuMW 2014 - Conf. Proceedings; EuMC 2014 44th Eur. Microw. Conf.*, pp. 484–487, 2014, doi: 10.1109/EuMC.2014.6986476.
- [33] S. Singh, B. Sahu, and S. P. Singh, “Hyperthermia performance of conformal applicator for limb tumor in presence of water bolus,” *2017 Int. Symp. Antennas Propagation, ISAP 2017*, vol. 2017-January, pp. 1–2, 2017, doi: 10.1109/ISANP.2017.8228875.
- [34] J. J. A. Flores-Cuautle et al., “Termoterapia Para Tratamiento del Dolor: Modelado Paramétrico de la Temperatura,” *Congr. Nac. Ing. Biomed.*, no. 1, pp. 269-272%V 4, 2017, [Online]. Available: <http://memorias.somib.org.mx/index.php/memorias/article/view/264>
- [35] N. Johnson, J. Abraham, Z. Helgeson, W. Minkowycz and E. Sparrow, "An Archive of Skin-Layer Thicknesses and Properties and Calculations of Scald Burns With Comparisons to Experimental Observations", *Journal of Thermal Science and Engineering Applications*, vol. 3, no. 1, 2011.
- [36] Hasgall PA, Di Gennaro F, Baumgartner C, Neufeld E, Lloyd B, Gosselin MC, Payne D, Klingeböck A, Kuster N, “IT’IS Database for thermal and electromagnetic parameters of biological tissues,” Version 4.0, May 15, 2018, DOI: 10.13099/VIP21000-04-0.
- [37] "Specific Absorption Rate (SAR) in the Human Brain", COMSOL. [Online]. Available: <https://www.comsol.com/model/specific-absorption-rate-sar-in-the-human-brain-2190>. [Accessed: 03- Aug- 2020]
- [38] S. Jacobsen and P. R. Stauffer, “Multifrequency radiometric determination of temperature profiles in a lossy homogeneous phantom using a dual-mode

- antenna with integral water bolus," IEEE Trans. Microw. Theory Tech., vol. 50, no. 7, pp. 1737–1746, 2002, doi: 10.1109/TMTT.2002.800424.
- [39] A. Morimoto et al., "Association between hand muscle thickness and whole-body skeletal muscle mass in healthy adults: A pilot study," J. Phys. Ther. Sci., vol. 29, no. 9, pp. 1644–1648, 2017, doi: 10.1589/jpts.29.1644.
- [40] Eric Johnson (2021). STL File Reader (<https://www.mathworks.com/matlabcentral/fileexchange/22409-stl-file-reader>), MATLAB Central File Exchange. Retrieved August 12, 2021
- [41] M. España, "Moto G7 Plus | motorola ES - motorola España | Teléfonos Android y Razr", Motorola.es. [Online]. Disponible en: <https://www.motorola.es/smartphones-moto-g7-plus/p>. [Visitado el: 16- Oct- 2020]

Apéndice I

Programa *ext_coor_apli.mat*

```
% Centro de Investigación y Estudios Avanzados del IPN %
% Avance de tesis %
% Daniel Gerardo Serrano Díaz %
% Extracción de coordenadas de los aplicadores %
tic
clear all, close all, clc %limpia el workspace, cierra todas figures abiertas y limpia la
command window

archivo_stl = stlread('prueba1_dioxido - Waveguide_dioxido.STL'); % lee el archivo
en formato .stl

patch(archivo_stl,'FaceColor', [0.8 0.8 1.0], ...
      'EdgeColor', 'none', ...
      'FaceLighting', 'gouraud', ...
      'AmbientStrength', 0.15); % Se agrega luz al objeto reconstruido

camlight('headlight');
material('dull');

axis('image');
hold on, grid on
view([-135 35]);%vista del objeto
vertices_totales=getfield(archivo_stl,'vertices');%obtiene los vertices del archivo .stl
vertices_unicos=unique(vertices_totales,'rows').%almacena los vertices sin repetirlos
xlabel('Eje x (mm)'),ylabel('Eje y (mm)'),zlabel('Eje z (mm)');%etiqueta los ejes
pause();

%distancia entre dos puntos en un plano 3D
%d=sqrt((x1-x2)^2+(y1-y2)^2+(z1-z2)^2)

% dist_ejez=AD;%distancia que buscara el codigo----- d_ejez
tolerancia1=.0005,%tolerancia de la distancia

% AB=42;
% AD=84;
% AF=169;
AB=53;
AD=93;
AF=169.2;
AC=sqrt(AB^2+AD^2);
```

```

AE=sqrt(AF^2+AB^2);
AG=sqrt(AD^2+AF^2+AB^2);
AH=sqrt(AF^2+AD^2);

dist_ejez=AB;

% vertices_unicos_copia=vertices_unicos;%se hace una copia de los vertices
detectados
contador1=1;%contador de la variable que almacena distancia entre los vertices
contador2=1;%contador de las coordenadas de las distancias que son iguales a d,
%en este caso en particular AB

for contador_m1=1:length(vertices_unicos)
    for contador_m2=1:length(vertices_unicos)
x1_discriminacion=vertices_unicos(contador_m1,1);%valor en x del vertice 1 -----
x1_discriminacion
y1_discriminacion=vertices_unicos(contador_m1,2);%valor en y del vertice 1
z1_discriminacion=vertices_unicos(contador_m1,3);%valor en z del vertice 1
x2_discriminacion=vertices_unicos(contador_m2,1);%valor en x del vertice 2
y2_discriminacion=vertices_unicos(contador_m2,2);%valor en y del vertice 2
z2_discriminacion=vertices_unicos(contador_m2,3);%valor en z del vertice 2
dist1_busqueda(contador1)=sqrt((x1_discriminacion-
x2_discriminacion)^2+(y1_discriminacion-y2_discriminacion)^2+(z1_discriminacion-
z2_discriminacion)^2);%calcula de la distancia entre
%los dos vertices
if dist1_busqueda(contador1)>dist_ejez-tolerancia1 &&
dist1_busqueda(contador1)<dist_ejez+tolerancia1%condicional, si d esta dentro del
rango permitido
%de la distancia determinada

    coor_vertices_encontrados1(contador2,:)=vertices_unicos(contador_m1,:);%variable
que almacena las coordenadas del vertice encontrado ---- total_coor
    valor_dist1(contador2)=dist1_busqueda(contador1);%guarda el valor de la
distancia obtenida
    contador2=contador2+1;%aumenta en 1 el contador de la variable que almacena
las coordenadas
end
contador1=contador1+1;%aumenta en 1 el contador de la variable que calcula la
distancia entre los vertices
    end
end
%-----
vertices_originales=plot3(vertices_unicos(:,1),vertices_unicos(:,2),vertices_unicos(:,3))
,'*','Color','m');%grafica todos los vertices en la figura con un *
xlabel('Eje x (mm)'),ylabel('Eje y (mm)'),zlabel('Eje z (mm)');%etiqueta los ejes
hold on,grid on

```



```

vertices_guia=plot3(coor_vertices_encontrados1(:,1),coor_vertices_encontrados1(:,2)
,coor_vertices_encontrados1(:,3),'o','Color','g','MarkerSize',10);%grafica todos los
vertices en la figura con un * ----- total_ver_gui
etiquetas1=legend([vertices_originales,vertices_guia],{'Vértices originales','Vértices
encontrados'});%coloca las etiquetas a los vertices originales y a los detectados

[num_vertices]=size(coor_vertices_encontrados1);%se almacena el tamaño de la
variable coor, en a & b
disp(['Número de vértices detectados: ',num2str(num_vertices(1))])%despliega el
número de vertices detectados

pause();

%ciclo while que espera a que el usuario de clic sobre la última figura
%desplegada
boton_espera = 2;
while boton_espera ~= 0
    disp('De clic en algún lugar del espacio 3D');
    boton_espera = waitforbuttonpress;
end
datos_cursor = get(gca,'CurrentPoint');
punto_medio_cursor=mean(datos_cursor);%obtiene el punto seleccionado con el clic
punto_mediox_cursor=punto_medio_cursor(:,1);%posicion x del punto seleccionado
punto_medioy_cursor=punto_medio_cursor(:,2);%posicion y del punto seleccionado
punto_medioz_cursor=punto_medio_cursor(:,3);%posicion z del punto seleccionado

disp('Coordenadas del punto seleccionado')%despliega las coordenadas obtenidas
disp(['x = ',num2str(punto_mediox_cursor)])
disp(['y = ',num2str(punto_medioy_cursor)])
disp(['z = ',num2str(punto_medioz_cursor)])

grafica_puntomedio=plot3(punto_medio_cursor(:,1),punto_medio_cursor(:,2),punto_
medio_cursor(:,3),'.','Color',[.09,.7,.01],'MarkerSize',25);%grafica todos los vertices en
la figura con un *
etiquetas2=legend([vertices_originales,vertices_guia,grafica_puntomedio],{'Vértices
originales','Vértices encontrados','Punto seleccionado'});%coloca las etiquetas a los
vertices originales y a los detectados
var_control_reubicacion=0;
%ciclo while que controla la recolocación del punto previamente
%seleccionado con el mouse
boton_control_reubicacion=1;
while boton_control_reubicacion~=0
    mensaje_reubicacion='¿Desea recolocar la ubicación del punto seleccionado (Y/N)?
';
    respuesta1=input(mensaje_reubicacion,'s');
    if respuesta1=='Y' || respuesta1=='y'

```

```

    coorx_manual='Introduzca el valor en el eje x: ';
    nueva_coorpx=input(coorx_manual);
    coory_manual='Introduzca el valor en el eje y: ';
    nueva_coorpy=input(coory_manual);
    coorz_manual='Introduzca el valor en el eje z: ';
    nueva_coorpz=input(coorz_manual);
    punto_medio2=[nueva_coorpx,nueva_coorpy,nueva_coorpz];

grafica_puntomedio2=plot3(punto_medio2(:,1),punto_medio2(:,2),punto_medio2(:,3),'
.', 'Color',[.09,.7,.9], 'MarkerSize',25);%grafica todos los vertices en la figura con un *

etiquetas3=legend([vertices_originales,vertices_guia,grafica_puntomedio,grafica_pun
tomedio2],{'Vértices originales','Vértices encontrados','Punto medio','Punto
movido'});%coloca etiqueta en la figura
    var_control_reubicacion=var_control_reubicacion+1;
else
    if var_control_reubicacion~=0
        boton_control_reubicacion=0;
        punto_medio2=punto_medio2;
    else
        boton_control_reubicacion=0;
        punto_medio2=punto_medio_cursor;
    end
end
end

end

for contador_m2=1:num_vertices(1)
    recorrido_vertices_encontrados1=coor_vertices_encontrados1(contador_m2,:,:),%se
    recorren las coordenadas originales en función de la variable j
    coor_reubicadas(contador_m2,:)=recorrido_vertices_encontrados1-
    punto_medio2;%reubica las coordenadas de acuerdo con el punto de referencia
    definido previamente
end

figure
punt_origen=plot3(0,0,0,'.', 'Color','b', 'Markersize',25);%grafica el punto (0,0,0)
hold on,grid on
vertices_reubicados=plot3(coor_reubicadas(:,1),coor_reubicadas(:,2),coor_reubicada
s(:,3),'.', 'Color',[.85,.325,0.0980], 'MarkerSize',25);%grafica todos los vertices en la
figura con un *
axis([-2000 2000 -2000 2000 -2000 2000])%coloca valores maximos y minimos en
los ejes (x,y & z)
etiquetas4=legend([punt_origen,vertices_reubicados],{'Origen del plano','Vértices re-
ubicados'});%coloca etiqueta en los vertices reubicados
xlabel('Eje x (mm)'),ylabel('Eje y (mm)'),zlabel('Eje z (mm)');%etiqueta los ejes

```

```

view([-135 35]);%vista del objeto
% axis ij;%corrige los ejes de y
% pause()
dist_ejez=44.19;%distancia que buscara el codigo
tol1=.005;%tolerancia de la distancia

numero_aplicadores=length(coor_reubicadas)/8;

contador1=1;%contador de la variable que almacena distancia entre los vertices
contador2=1;%contador de las coordenadas de las distancias que son iguales a d,
%en este caso en particular AB
control_matriz_aplicador=1;%variable del numero de aplicador

clear dc val_dis

control_while=0;
% cto=0;

%   salvar=coor_reubicadas;
% for i=1:2

while control_while<numero_aplicadores

    posicion_coordenas=zeros(1,8);
    posicion_coordenas(contador2)=1;
    contador2=contador2+1;

    for contador_m2=2:length(coor_reubicadas)

x1_discriminacion=coor_reubicadas(1,1);%valor en x del vertice 1
y1_discriminacion=coor_reubicadas(1,2);%valor en y del vertice 1
z1_discriminacion=coor_reubicadas(1,3);%valor en z del vertice 1
x2_discriminacion=coor_reubicadas(contador_m2,1);%valor en x del vertice 2
y2_discriminacion=coor_reubicadas(contador_m2,2);%valor en y del vertice 2
z2_discriminacion=coor_reubicadas(contador_m2,3);%valor en z del vertice 2

dist1_busqueda=sqrt((x1_discriminacion-x2_discriminacion)^2+(y1_discriminacion-
y2_discriminacion)^2+(z1_discriminacion-z2_discriminacion)^2);%calculo de la
distancia entre los dos vertices

if dist1_busqueda>AB-tolerancia1 && dist1_busqueda<AB+tolerancia1%AB
    posicion_coordenas(5)=contador_m2;
    contador2=contador2+1;
elseif dist1_busqueda>AC-tol1 && dist1_busqueda<AC+tol1%AC
    posicion_coordenas(6)=contador_m2;

```

```

    contador2=contador2+1;
elseif dist1_busqueda>AD-tolerancia1 && dist1_busqueda<AD+tolerancia1 %AD
    posicion_coordenas(2)=contador_m2;
    contador2=contador2+1;
elseif dist1_busqueda>AE-tol1 && dist1_busqueda<AE+tol1 %AE
    posicion_coordenas(7)=contador_m2;
    contador2=contador2+1;
elseif dist1_busqueda>AF-tolerancia1 && dist1_busqueda<AF+tolerancia1 %AF
    posicion_coordenas(3)=contador_m2;
    contador2=contador2+1;
elseif dist1_busqueda>AG-tol1 && dist1_busqueda<AG+tol1 %AG
    posicion_coordenas(8)=contador_m2;
    contador2=contador2+1;
elseif dist1_busqueda>AH-tol1 && dist1_busqueda<AH+tol1 %AH
    posicion_coordenas(4)=contador_m2;
    contador2=contador2+1;
else
    clear dc
%   l=1;
end
% l=l+1;%aumenta en 1 el contador de la variable que calcula la distancia entre los
vertices
    end
%   posicion_coordenas

matriz_aplicador(:, :, control_matriz_aplicador)=coor_reubicadas(posicion_coordenas, :
);

    control_matriz_aplicador=control_matriz_aplicador+1;

    coor_reubicadas(posicion_coordenas, :)=[];

    clear posicion_coordenas
    contador2=1;
%   l=1;
    control_while=control_while+1;
end
% end
% matriz
% apli(:, :, nu_aplicadores, ap)=n_coor(matriz, :);

clear i j k

contador_m2=1;
contador2=8;

```

```

figure
hold on, grid on
punto_origen=plot3(0,0,0,'.','Color','b','Markersize',25);%grafica el punto (0,0,0)
etiquetas5=legend(punto_origen,{'Origen del plano'});%coloca etiquetas

for contador_m1=1:numero_aplicadores
    hold on

    grafica_aplicadores=plot3(matriz_aplicador(:,1,contador_m1),matriz_aplicador(:,2,con
    tador_m1),matriz_aplicador(:,3,contador_m1),'.','Color',[rand,rand,rand],'MarkerSize',2
    5);%grafica todos los vertices en la figura con un *
    axis([-3000 3000 -3000 3000 -3000 3000])%coloca valores maximos y minimos en
    los ejes (x,y & z)
    % lg(i)=legend(sprintf('Aplicador %d', i))%coloca etiqueta en los vertices
    reubicados
    % lg=legend(apli,'aplicador ')
    xlabel('Eje x (mm)'),ylabel('Eje y (mm)'),zlabel('Eje z (mm)');%etiqueta los ejes
    view([-135 35]);%vista del objeto
end

clear i dc

% limite=num_vertices/numero_aplicadores;

for contador3=1:numero_aplicadores
%Una vez agrupados los vértices en grupos (aplicadores), los vertices se
%ordenan de acuerdo con el valor de z
vertices_ordenados_z=sortrows(matriz_aplicador(:,:,contador3),3);

%Al estar ordenados de acuerdo con su valor z (de menor a mayor), se toma
%el primer vertice y se mide su distancia con el segundo.
distancia_ordenacion=sqrt((vertices_ordenados_z(1,1)-
vertices_ordenados_z(2,1))^2+(vertices_ordenados_z(1,2)-
vertices_ordenados_z(2,2))^2+(vertices_ordenados_z(1,3)-
vertices_ordenados_z(2,3))^2);

%Se busca una coincidencia de la distancia obtenida (AB mm, AD mm ó AF mm)
%dependiendo de la coincidencia encontrada se acomodan los demas vértices
%en sentido horario de acuerdo con el primer vértice. Tanto la cara
%inferior (vertices 1-4) como la cara superior (vertices 5-8)

if distancia_ordenacion>AB-tolerancia1 &&
distancia_ordenacion<AB+tolerancia1%distancia entre el vértices 1 y 2 = AB mm
    posicion_ordenados=zeros(1,8);%vector que almacena la posición de los vertices
    agrupados
    posicion_ordenados(1)=1;%la posición del vertice 1 se mantiene

```

```

    posicion_ordenados(2)=2;%la posición del vertice 2 se mantiene
%   var(contador3)=1;

    for contador_m1=1:6

%Se calcula la distancia entre el vértice 1 y los vertices restantes (3 al 8)
    dist_ejez=sqrt((vertices_ordenados_z(1,1)-
vertices_ordenados_z(contador_m1+2,1))^2+(vertices_ordenados_z(1,2)-
vertices_ordenados_z(contador_m1+2,2))^2+(vertices_ordenados_z(1,3)-
vertices_ordenados_z(contador_m1+2,3))^2);

%Nuevamente se busca la coincidencia entre la distancia calculada y la
%distancia de referencia
if dist_ejez>AD-tolerancia1 && dist_ejez<AD+tolerancia1%vértices AD
    posicion_ordenados(4)=contador_m1+2;
elseif dist_ejez>AC-tol1 && dist_ejez<AC+tol1%vértices AC
    posicion_ordenados(3)=contador_m1+2;
elseif dist_ejez>AF-tolerancia1 && dist_ejez<AF+tolerancia1%vértices AF
    posicion_ordenados(5)=contador_m1+2;
elseif dist_ejez>AE-tol1 && dist_ejez<AE+tol1%vértices AE
    posicion_ordenados(6)=contador_m1+2;
elseif dist_ejez>AH-tol1 && dist_ejez<AH+tol1%vértices AH
    posicion_ordenados(8)=contador_m1+2;
elseif dist_ejez>AG-tol1 && dist_ejez<AG+tol1%vértices AG
    posicion_ordenados(7)=contador_m1+2;
end
    end

elseif distancia_ordenacion>AD-tolerancia1 &&
distancia_ordenacion<AD+tolerancia1
    posicion_ordenados=zeros(1,8);%vector que almacena la posición de los vertices
agrupados
    posicion_ordenados(1)=1;%la posición del vertice 1 se mantiene
    posicion_ordenados(2)=2;%la posición del vertice 2 se mantiene
%   var(contador3)=2;
    for contador_m1=1:6
        %Se calcula la distancia entre el vértice 1 y los vertices restantes (3 al 8)
        dist_ejez=sqrt((vertices_ordenados_z(1,1)-
vertices_ordenados_z(contador_m1+2,1))^2+(vertices_ordenados_z(1,2)-
vertices_ordenados_z(contador_m1+2,2))^2+(vertices_ordenados_z(1,3)-
vertices_ordenados_z(contador_m1+2,3))^2);

%Nuevamente se busca la coincidencia entre la distancia calculada y la
%distancia de referencia
if dist_ejez>AB-tolerancia1 && dist_ejez<AB+tolerancia1%vértices AB
    posicion_ordenados(4)=contador_m1+2;

```

```

elseif dist_ejez>AC-tol1 && dist_ejez<AC+tol1%vértices AC
    posicion_ordenados(3)=contador_m1+2;
elseif dist_ejez>AF-tolerancia1 && dist_ejez<AF+tolerancia1%vértices AF
    posicion_ordenados(5)=contador_m1+2;
elseif dist_ejez>AE-tol1 && dist_ejez<AE+tol1%vértices AE
    posicion_ordenados(8)=contador_m1+2;
elseif dist_ejez>AH-tol1 && dist_ejez<AH+tol1%vértices AH
    posicion_ordenados(6)=contador_m1+2;
elseif dist_ejez>AG-tol1 && dist_ejez<AG+tol1%vértices AG
    posicion_ordenados(7)=contador_m1+2;
end
    end

elseif distancia_ordenacion>AF-tolerancia1 && distancia_ordenacion<AF+tolerancia1
    posicion_ordenados=zeros(1,8);%vector que almacena la posición de los vertices
agrupados
    posicion_ordenados(1)=1;%la posición del vertice 1 se mantiene
    posicion_ordenados(2)=2;%la posición del vertice 2 se mantiene
%    var(contador3)=3;
    for contador_m1=1:6
        %Se calcula la distancia entre el vértice 1 y los vertices restantes (3 al 8)
        dist_ejez=sqrt((vertices_ordenados_z(1,1)-
vertices_ordenados_z(contador_m1+2,1))^2+(vertices_ordenados_z(1,2)-
vertices_ordenados_z(contador_m1+2,2))^2+(vertices_ordenados_z(1,3)-
vertices_ordenados_z(contador_m1+2,3))^2);

%Nuevamente se busca la coincidencia entre la distancia calculada y la
%distancia de referencia
if dist_ejez>AD-tolerancia1 && dist_ejez<AD+tolerancia1%vértices AD
    posicion_ordenados(5)=contador_m1+2;
elseif dist_ejez>AC-tol1 && dist_ejez<AC+tol1%vértices AC
    posicion_ordenados(8)=contador_m1+2;
elseif dist_ejez>AB-tolerancia1 && dist_ejez<AB+tolerancia1%vértices AB
    posicion_ordenados(4)=contador_m1+2;
elseif dist_ejez>AE-tol1 && dist_ejez<AE+tol1%vértices AE
    posicion_ordenados(3)=contador_m1+2;
elseif dist_ejez>AH-tol1 && dist_ejez<AH+tol1%vértices AH
    posicion_ordenados(6)=contador_m1+2;
elseif dist_ejez>AG-tol1 && dist_ejez<AG+tol1%AG
    posicion_ordenados(7)=contador_m1+2;
end

    end

else
    disp('Error punto minimo')

```

end

```
coordenadas_exportacion(:, :, contador3) = vertices_ordenados_z(posicion_ordenados, :);
```

end

```
for contador_m2=1:numero_aplicadores
```

```
    coordenadas_aplicadores = coordenadas_exportacion(:, :, contador_m2);  
    vertice_original = coordenadas_aplicadores(1, :);
```

```
    vertices_referencia(1, 1) = vertice_original(1);  
    vertices_referencia(1, 2) = vertice_original(2);  
    vertices_referencia(1, 3) = vertice_original(3);
```

```
    vertices_referencia(2, 1) = vertice_original(1) + AD;  
    vertices_referencia(2, 2) = vertice_original(2);  
    vertices_referencia(2, 3) = vertice_original(3);
```

```
    vertices_referencia(3, 1) = vertice_original(1);  
    vertices_referencia(3, 2) = vertice_original(2);  
    vertices_referencia(3, 3) = vertice_original(3) + AB;
```

```
    vertices_referencia(4, 1) = vertice_original(1);  
    vertices_referencia(4, 2) = vertice_original(2) + AF;  
    vertices_referencia(4, 3) = vertice_original(3);
```

```
for contador_m1=2:8
```

```
    distancia_vertices = sqrt((coordenadas_aplicadores(1, 1) -  
    coordenadas_aplicadores(contador_m1, 1))^2 + (coordenadas_aplicadores(1, 2) -  
    coordenadas_aplicadores(contador_m1, 2))^2 + (coordenadas_aplicadores(1, 3) -  
    coordenadas_aplicadores(contador_m1, 3))^2);
```

```
    if distancia_vertices > AB - tolerancia1 && distancia_vertices < AB + tolerancia1  
        distancia_ABmm = coordenadas_aplicadores(contador_m1, :);  
    elseif distancia_vertices > AD - tolerancia1 && distancia_vertices < AD + tolerancia1  
        distancia_ADmm = coordenadas_aplicadores(contador_m1, :);  
    elseif distancia_vertices > AF - tolerancia1 && distancia_vertices < AF + tolerancia1  
        distancia_AFmm = coordenadas_aplicadores(contador_m1, :);  
    end
```

end

```
figure, hold on, grid on
```



```

npos=plot3(coordenadas_aplicadores(1:8,1),coordenadas_aplicadores(1:8,2),coor
nadas_aplicadores(1:8,3),'.','Color',[rand,rand,rand],'MarkerSize',25);%grafica todos
los vertices en la figura con un *
vref=plot3(vertices_referencia(:,1),vertices_referencia(:,2),vertices_referencia(:,3),'.','
Color',[rand,rand,rand],'MarkerSize',25);%grafica todos los vertices en la figura con
un *
axis([-1000 1000 -1000 1000 -1000 1000])%coloca valores maximos y minimos en
los ejes (x,y & z)

xlabel('Eje x (mm)'),ylabel('Eje y (mm)'),zlabel('Eje z (mm)');%etiqueta los ejes
view([-135 35]);%vista del objeto

```

```
% plano xy
```

```

vector1=distancia_ABmm(1:2)-vertice_original(1:2);
vector2=vertices_referencia(3,1:2)-vertice_original(1:2);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(1,1)=acosd(producto_puntp/producto_normas);

```

```
% plano xz
```

```

vector1=distancia_ABmm(1:2:3)-vertice_original(1:2:3);
vector2=vertices_referencia(3,1:2:3)-vertice_original(1:2:3);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(1,2)=acosd(producto_puntp/producto_normas);

```

```
% plano yz
```

```

vector1=distancia_ABmm(2:3)-vertice_original(2:3);
vector2=vertices_referencia(3,2:3)-vertice_original(2:3);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(1,3)=acosd(producto_puntp/producto_normas);

```

```
% plano xy
```

```
vector1=distancia_ADmm(1:2)-vertice_original(1:2);
```

```

vector2=vertices_referencia(2,1:2)-vertice_original(1:2);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(2,1)=acosd(producto_puntp/producto_normas);

```

% plano xz

```

vector1=distancia_ADmm(1:2:3)-vertice_original(1:2:3);
vector2=vertices_referencia(2,1:2:3)-vertice_original(1:2:3);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(2,2)=acosd(producto_puntp/producto_normas);

```

% plano yz

```

vector1=distancia_ADmm(2:3)-vertice_original(2:3);
vector2=vertices_referencia(2,2:3)-vertice_original(2:3);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(2,3)=acosd(producto_puntp/producto_normas);

```

% plano xy

```

vector1=distancia_AFmm(1:2)-vertice_original(1:2);
vector2=vertices_referencia(4,1:2)-vertice_original(1:2);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(3,1)=acosd(producto_puntp/producto_normas);

```

% plano xz

```

vector1=distancia_AFmm(1:2:3)-vertice_original(1:2:3);
vector2=vertices_referencia(4,1:2:3)-vertice_original(1:2:3);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(3,2)=acosd(producto_puntp/producto_normas);

```

```

% plano yz
vector1=distancia_AFmm(2:3)-vertice_original(2:3);
vector2=vertices_referencia(4,2:3)-vertice_original(2:3);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(3,3)=acosd(producto_puntp/producto_normas);

theta(:, :, contador_m2)=theta1(:, :);

% an_planoxy=listo
%
% an_planoxz=
% an_planoyz=

end

datos_salida= struct('coordenadas',coordenadas_exportacion,'grados',theta)

toc

```

Apendice II

Programa *ext_coor_apli.mat*

```
%      Centro de Investigación y Estudios Avanzados del IPN      %
%      Avance de tesis      %
%      Daniel Gerardo Serrano Díaz      %
%      Extracción de coordenadas de los aplicadores      %
tic
clear all, close all, clc %limpia el workspace, cierra todas figuras abiertas y limpia la
command window

% disp('Ingrese medidas principales del aplicador')
dim_a=93;%input('Dimension a: ');
dim_b=53;%input('Dimension b: ');
dim_z=169.2;%input('Dimension z: ');

%calculo de distancias de un prisma rectangular con las medidas ingre
%sadas
AB=dim_b;
AD=dim_a;
AF=dim_z;
AC=sqrt(AB^2+AD^2);
AE=sqrt(AF^2+AB^2);
AG=sqrt(AD^2+AF^2+AB^2);
AH=sqrt(AF^2+AD^2);
%%
%-----Despliegue modelo sencillo-----
archivo_stl = stlread('2guias_sim_esc_22_v2.stl'); % lee el archivo en formato .stl
% archivo_stl = stlread('Red_esc_36.stl'); % lee el archivo en formato .stl

patch(archivo_stl,'FaceColor', [0.8 0.8 1.0], ...
      'EdgeColor', 'none', ...
      'FaceLighting', 'gouraud', ...
      'AmbientStrength', 0.15); % Se agrega luz al objeto reconstruido

camlight('headlight');
material('dull');

axis('image');
hold on
view([-135 35]);%vista del objeto
vertices_totales=getfield(archivo_stl,'vertices');%obtiene los vertices del archivo .stl
vertices_unicos=unique(vertices_totales,'rows');%almacena los vertices sin repetirlos
```

```

%distancia entre dos puntos en un plano 3D
%d=sqrt((x1-x2)^2+(y1-y2)^2+(z1-z2)^2)
%%
%-----Deteccion de vértices-----

dist_ejez=AB,%distancia que buscara el codigo----- d_ejez
tolerancia1=5,%tolerancia de la distancia

% vertices_unicos_copia=vertices_unicos;%se hace una copia de los vertices
detectados
contador1=1;%contador de la variable que almacena distancia entre los vertices
contador2=1;%contador de las coordenadas de las distancias que son iguales a d,
%en este caso en particular 42

tam_vertices=length(vertices_unicos);%numero de vertices totales en la geometria
sencilla

for contador_m1=1:tam_vertices
    for contador_m2=contador_m1:tam_vertices
x1_discriminacion=vertices_unicos(contador_m1,1);%valor en x del vertice 1 -----
x1_discriminacion
y1_discriminacion=vertices_unicos(contador_m1,2);%valor en y del vertice 1
z1_discriminacion=vertices_unicos(contador_m1,3);%valor en z del vertice 1
x2_discriminacion=vertices_unicos(contador_m2,1);%valor en x del vertice 2
y2_discriminacion=vertices_unicos(contador_m2,2);%valor en y del vertice 2
z2_discriminacion=vertices_unicos(contador_m2,3);%valor en z del vertice 2
dist1_busqueda(contador1)=sqrt((x1_discriminacion-
x2_discriminacion)^2+(y1_discriminacion-y2_discriminacion)^2+(z1_discriminacion-
z2_discriminacion)^2);%calculo de la distancia entre
%los dos vertices
if dist1_busqueda(contador1)>dist_ejez-tolerancia1 &&
dist1_busqueda(contador1)<dist_ejez+tolerancia1%condicional, si d esta dentro del
rango permitido
%de la distancia determinada

coor_vertices_encontrados1(contador2,:)=vertices_unicos(contador_m1,:);%variable
que almacena las coordenadas del vertice encontrado ---- total_coor

coor_vertices_encontrados1_2(contador2,:)=vertices_unicos(contador_m2,:);%variabl
e que almacena las coordenadas del vertice encontrado ---- total_coor
    valor_dist1(contador2)=dist1_busqueda(contador1);%guarda el valor de la
distancia obtenida
    contador2=contador2+1;%aumenta en 1 el contador de la variable que almacena
las coordenadas

```

```

end
contador1=contador1+1;%aumenta en 1 el contador de la variable que calcula la
distancia entre los vertices
    end

end

coor_vertices_enc_tot=unique([coor_vertices_encontrados1;
coor_vertices_encontrados1_2], 'rows');%vertices totales encontrados;
%%
%-----Despliegue modelo sencillo y vertices-----

%
vertices_originales=plot3(vertices_unicos(:,1),vertices_unicos(:,2),vertices_unicos(:,3)
, '*','Color','m');%grafica todos los vertices en la figura con un *
xlabel('Eje x'),ylabel('Eje y'),zlabel('Eje z');%etiqueta los ejes
hold on,grid on
[num_vertices1]=size(coor_vertices_enc_tot);%se almacena el tamaño de la variable
coor, en a & b
disp(['Número de vértices detectados: ',num2str(num_vertices1(1))])%despliega el
número de vertices detectados
%%
%-----Selección de caras superiores-----
%El usuario debe seleccionar algunos puntos cerca de las caras superiores
%dependiendo del numero de aplicadores que se encuentren en la
%reconstruccion

%ciclo while que espera a que el usuario de clic sobre la última figura
%desplegada
boton_espera = 2;
while boton_espera ~= 0
    disp('De clic en algún lugar del espacio 3D');
    boton_espera = waitforbuttonpress;
end
datos_cursor = get(gca,'CurrentPoint');
punto_medio_cursor=mean(datos_cursor);%obtiene el punto seleccionado con el clic
punto_mediox_cursor=punto_medio_cursor(:,1);%posicion x del punto seleccionado
punto_medioy_cursor=punto_medio_cursor(:,2);%posicion y del punto seleccionado
punto_medioz_cursor=punto_medio_cursor(:,3);%posicion z del punto seleccionado

disp(' ')%despliega las coordenadas obtenidas
disp('Coordenadas del punto seleccionado:')%despliega las coordenadas obtenidas
disp(['x1 = ',num2str(punto_mediox_cursor)])
disp(['y1 = ',num2str(punto_medioy_cursor)])
disp(['z1 = ',num2str(punto_medioz_cursor)])

```

```

pause()

%ciclo while que espera a que el usuario de clic sobre la última figura
%desplegada
boton_espera = 2;
while boton_espera ~= 0
    disp("")
    disp('De clic en algún lugar del espacio 3D');
    boton_espera = waitforbuttonpress;
end
datos_cursor2 = get(gca,'CurrentPoint');
punto_medio2_cursor=mean(datos_cursor2);%obtiene el punto seleccionado con el
clic
punto_mediox2_cursor=punto_medio2_cursor(:,1);%posicion x del punto
seleccionado
punto_medioy2_cursor=punto_medio2_cursor(:,2);%posicion y del punto
seleccionado
punto_medioz2_cursor=punto_medio2_cursor(:,3);%posicion z del punto
seleccionado

% disp(' ')%despliega las coordenadas obtenidas
disp('Coordenadas del punto seleccionado:')%despliega las coordenadas obtenidas
disp(['x2 = ',num2str(punto_mediox2_cursor)])
disp(['y2 = ',num2str(punto_medioy2_cursor)])
disp(['z2 = ',num2str(punto_medioz2_cursor)])

grafica_puntomedio=plot3(punto_medio_cursor(:,1),punto_medio_cursor(:,2),punto_
medio_cursor(:,3),'!','Color',[.09,.7,.01],'MarkerSize',25);%grafica todos los vertices en
la figura con un *
grafica_puntomedio2=plot3(punto_medio2_cursor(:,1),punto_medio2_cursor(:,2),punto
o_medio2_cursor(:,3),'!','Color',[.09,.7,.01],'MarkerSize',25);%grafica todos los
vertices en la figura con un *
vertices_encontrados=plot3(coor_vertices_enc_tot(:,1),coor_vertices_enc_tot(:,2),coo
r_vertices_enc_tot(:,3),'*','Color',[.09,.7,.9],'MarkerSize',25);%grafica todos los
vertices en la figura con un *
etiquetas2=legend([vertices_encontrados,grafica_puntomedio,grafica_puntomedio2],{
'Vértices encontrados','Punto seleccionado 1','Punto seleccionado 2'});%coloca las
etiquetas a los vertices originales y a los detectados
%%
%-----Discriminación vertices de caras superiores-----

tam_ver_enc=size(coor_vertices_enc_tot);%tamaño de los vertices encontrados en la
geometria sencilla
contador1=1;%contador generico

```

```

for contador_m1=1:tam_ver_enc(1)

x1_discriminacion=coor_vertices_enc_tot(contador_m1,1);%valor en x del vertice 1 --
--- x1_discriminacion
y1_discriminacion=coor_vertices_enc_tot(contador_m1,2);%valor en y del vertice 1
z1_discriminacion=coor_vertices_enc_tot(contador_m1,3);%valor en z del vertice 1

dist_punto_ver(contador1,1)=sqrt((punto_mediox_cursor-
x1_discriminacion)^2+(punto_medioy_cursor-
y1_discriminacion)^2+(punto_medioz_cursor-z1_discriminacion)^2);%calculo de la
distancia entre
%el primer punto seleccionado y los vertices encontrados
dist_punto_ver(contador1,2)=coor_vertices_enc_tot(contador_m1,1);%guarda
coordenada x
dist_punto_ver(contador1,3)=coor_vertices_enc_tot(contador_m1,2);%guarda
coordenada y
dist_punto_ver(contador1,4)=coor_vertices_enc_tot(contador_m1,3);%guarda
coordenada z

contador1=contador1+1;%aumenta en 1 el contador1

end

dist_punto_ver=sortrows(dist_punto_ver,1);%Ordena las distancias calculadas de
menor a mayor

caras=dist_punto_ver(1:4,2:4);%guarda los primeros 4 grupos de coordenadas

dist_punto_ver(:,1)=[];%elimina toda la columna de las distancias calculadas
dist_punto_ver(1:4,:)=[];%elimina las primeras cuatro filas que corresponden a las
%coordenadas guardadas

tam_ver_enc=size(dist_punto_ver);%almacena el nuevo tamaño de los vertices
restantes

coor_vertices_enc_tot=dist_punto_ver;%guarda las coordenadas en la variable de las
%coordenadas encontradas

contador1=1;%reinicio de contador generico

for contador_m1=1:tam_ver_enc(1)

x1_discriminacion=coor_vertices_enc_tot(contador_m1,1);%valor en x del vertice 1 --
--- x1_discriminacion
y1_discriminacion=coor_vertices_enc_tot(contador_m1,2);%valor en y del vertice 1
z1_discriminacion=coor_vertices_enc_tot(contador_m1,3);%valor en z del vertice 1

```



```

dist_punto_ver(contador1,1)=sqrt((punto_mediox2_cursor-
x1_discriminacion)^2+(punto_medioy2_cursor-
y1_discriminacion)^2+(punto_medioz2_cursor-z1_discriminacion)^2);%calculo de la
distancia entre
%el segundo punto seleccionado y los vertices restantes
dist_punto_ver(contador1,2)=coor_vertices_enc_tot(contador_m1,1);%almacena
coordenada x
dist_punto_ver(contador1,3)=coor_vertices_enc_tot(contador_m1,2);%almacena
coordenada y
dist_punto_ver(contador1,4)=coor_vertices_enc_tot(contador_m1,3);%almacena
coordenada z

contador1=contador1+1;%aumenta en 1 el contador1

end

dist_punto_ver=sortrows(dist_punto_ver,1);%ordena de menor a mayor las distancias
calculadas

caras=[caras;dist_punto_ver(1:4,2:4)];%almacena los 4 grupos de coordenadas

vertices_superiores=plot3(caras(:,1),caras(:,2),caras(:,3),'*','Color',[rand,rand,rand],'M
arkerSize',25);%grafica todos los vertices en la figura con un *
etiquetas2=legend([vertices_encontrados,grafica_puntomedio,grafica_puntomedio2,
vertices_superiores],{'Vértices encontrados','Punto seleccionado 1','Punto
seleccionado 2','Vértices superiores'});%coloca las etiquetas a los vertices originales
y a los detectados
%%
%-----Despliegue modelo completo-----

% archivo_stl2 = stlread('Red_esc.stl'); % lee el archivo en formato .stl
archivo_stl2 = stlread('2guias_sim_esc_v2.stl'); % lee el archivo en formato .stl

figure,patch(archivo_stl2,'FaceColor', [0.8 0.8 1.0], ...
'EdgeColor', 'none', ...
'FaceLighting', 'gouraud', ...
'AmbientStrength', 0.15); % Se agrega luz al objeto reconstruido

camlight('headlight');
material('dull');

axis('image');
hold on, grid on
view([-135 35]);%vista del objeto

```

```

vertices_totales2=getfield(archivo_stl2,'vertices'),%obtiene los vertices del archivo .stl
vertices_unicos2=unique(vertices_totales2,'rows'),%almacena los vertices sin
repetirlos

%
vertices_originales=plot3(vertices_unicos(:,1),vertices_unicos(:,2),vertices_unicos(:,3)
,'*','Color','m');%grafica todos los vertices en la figura con un *
xlabel('Eje x'),ylabel('Eje y'),zlabel('Eje z');%etiqueta los ejes

%
vertices_guia1=plot3(coor_vertices_encontrados1(:,1),coor_vertices_encontrados1(:,
2),coor_vertices_encontrados1(:,3),'*','Color','b','MarkerSize',20);%grafica todos los
vertices en la figura con un * ----- total_ver_gui
vertices_caras=plot3(caras(:,1),caras(:,2),caras(:,3),'!','Color',[rand,rand,rand],'Marker
Size',30),%grafica todos los vertices en la figura con un * ----- total_ver_gui

%%
%---Emparejamiento de vertices encontrados en modelo sencillo y completo---

%De acuerdo con los vértices encontrados en la geometría sencilla se
%realiza una búsqueda en la geometría completa

tam_caras=size(caras);%tamaño de la variable caras,es decir, el número de vertices
%por cada cara (4,8,12,etc.)
tam_ver_or=size(vertices_unicos2);%número de los vertices totales de la geometría
completa
clear dist1_búsqueda;%limpia la variable dist1_búsqueda
contador1=1;%reinicio del contador1
contador2=1;%reinicio del contador2

for contador_m1=1:tam_caras(1)

x1_discriminacion=caras(contador_m1,1);%valor en x del vertice 1 -----
x1_discriminacion
y1_discriminacion=caras(contador_m1,2);%valor en y del vertice 1
z1_discriminacion=caras(contador_m1,3);%valor en z del vertice 1

for contador_m2=1:tam_ver_or(1)

x2_discriminacion=vertices_unicos2(contador_m2,1);%valor en x del vertice 2
y2_discriminacion=vertices_unicos2(contador_m2,2);%valor en y del vertice 2
z2_discriminacion=vertices_unicos2(contador_m2,3);%valor en z del vertice 2

dist1_búsqueda(contador1)=sqrt((x1_discriminacion-
x2_discriminacion)^2+(y1_discriminacion-y2_discriminacion)^2+(z1_discriminacion-
z2_discriminacion)^2);%cálculo de la distancia entre

```

```

%los dos vertices

dist_caras(contador1,1)=dist1_busqueda(contador1);%almacena la distancia
calculada
dist_caras(contador1,2)=vertices_unicos2(contador_m2,1);%almacena la coordenada
x
dist_caras(contador1,3)=vertices_unicos2(contador_m2,2);%almacena la coordenada
y
dist_caras(contador1,4)=vertices_unicos2(contador_m2,3);%almacena la coordenada
z

contador1=contador1+1;%aumenta en 1 el contador1

end

% contador1=contador1-1;%resta en 1 el valor de contador

[pos_val_min1 pos_val_min2]=min(dist1_busqueda); %busca la distancia minima
entre el vertice del modelo sencillo al modelo completo
% pos_val_min1 almacena el valor encontrado
% pos_val_min2 almacena la posicion del valor encontrado en la matriz
dist1_busqueda

coor_ver_emp(contador2,1:4)=dist_caras(pos_val_min2,1:4);%guarda las
coordenadas y el valor de la distancia

contador2=contador2+1;

clear dis1_busqueda
contador1=1;

end

vertices_caras_emp=plot3(coor_ver_emp(:,2),coor_ver_emp(:,3),coor_ver_emp(:,4),'*
','Color',[rand,rand,rand],'MarkerSize',10);%grafica todos los vertices en la figura con
un * ----- total_ver_gui
etiquetas2=legend([vertices_caras,vertices_caras_emp],{'Vértices cara
superior','Vértices emparejados'});%coloca las etiquetas a los vertices originales y a
los detectados
%%
%----- Ordenar vértices del modelo complejo -----

%Suma los valores de cada una de las coordenadas de cada vertice en
%busqueda del valor menor
ver1_apli(1)=find(min(sum(coor_ver_emp(1:4,2:4),2))==sum(coor_ver_emp(1:4,2:4),2
));

```

```

ver1_apli(2)=find(min(sum(coor_ver_emp(5:8,2:4),2))==sum(coor_ver_emp(5:8,2:4),2
))+4;

% coor_ver_emp
coor_ver_emp([1 ver1_apli(1)],:)=coor_ver_emp([ver1_apli(1) 1],:);
coor_ver_emp([5 ver1_apli(2)],:)=coor_ver_emp([ver1_apli(2) 5],:);
% coor_ver_emp
%%
%----- Ordenar vértices para consol

% ver1=coor_ver_emp(1,2:4);
clear dist1_busqueda

contador2=1;
contador3=0;

for contador_m0=1:4:5

    ver1=coor_ver_emp(contador_m0,2:4);
    dist1_busqueda(1,2:4)=ver1;
    contador1=2;

for contador_m1=2+contador3:4+contador3

    dist1_busqueda(contador1,1)=sqrt((ver1(1)-
    coor_ver_emp(contador_m1,2))^2+(ver1(2)-
    coor_ver_emp(contador_m1,3))^2+(ver1(3)-
    coor_ver_emp(contador_m1,4))^2);%calculo de la distancia entre
%los dos vertices
    dist1_busqueda(contador1,2)= coor_ver_emp(contador_m1,2);
    dist1_busqueda(contador1,3)= coor_ver_emp(contador_m1,3);
    dist1_busqueda(contador1,4)= coor_ver_emp(contador_m1,4);

    contador1=contador1+1;

end

dist1_busqueda;
dist1_busqueda=sortrows(dist1_busqueda,1);%ordena de menor a mayor los vertices
de acuerdo con la distancia de estos con el primer vertice
dist1_busqueda([3 4],:)=dist1_busqueda([4 3],:);%intercambia las filas 3 y 4 para
tener los datos ordenados de tal forma que puedan ser exportados a consol

ver_orde(1+contador3:4+contador3,:)=dist1_busqueda;%guarda los valores de las
coordenadas y la distancia

```

```

clear dist1_busqueda
contador2=contador2+1;
contador1=1;
contador3=contador3+4;

end

graf_lineas(1:4,1:4)=ver_orde(1:4,:);
graf_lineas(6:9,1:4)=ver_orde(5:8,:);
graf_lineas(5,1:4)=ver_orde(1,:);
graf_lineas(10,1:4)=ver_orde(5,:);

% figure,grid on, hold on
% xlabel('Eje x'),ylabel('Eje y'),zlabel('Eje z');%etiqueta los ejes
plot3(graf_lineas(1:5,2),graf_lineas(1:5,3),graf_lineas(1:5,4),'LineWidth',12)
plot3(graf_lineas(6:10,2),graf_lineas(6:10,3),graf_lineas(6:10,4),'LineWidth',12)
%%
%----- Estimación de los vértices complementarios -----

tam_ver_orde=size(ver_orde);

num_apli=tam_ver_orde(1)/4;%numero de aplicadores

clear dist1_busqueda
contador1=1;
contador2=1;
contador3=1;
contador4=1;
contador5=1;

for contador_m1=1:tam_ver_orde(1)

ctrl_nu_apli=tam_ver_orde(1)-((num_apli*4)-4);%control del numero de aplicadores,
siempre multiplo de 4

val_ctrl=contador_m1;%variable de control con el mismo valor del contador_m1

if contador_m1>=ctrl_nu_apli%verifica si el contador_m1 es mayor al control de
numero de aplicadores
    num_apli=num_apli-1; %en caso positivo reduce en 1 el numero de aplicadores, es
decir, termina el primer grupo de vertices
end

val2=val_ctrl+1; %variable 2 cuyo valor es la variable de control mas uno
if val2>ctrl_nu_apli%si su valor es mayor a la variable de control
    val2=val2-4;%reduce su valor en 4 unidades

```

```

else
    val2=val2;%en caso contrario mantiene su valor
end

val3=val_ctrl+3;%variable 3 cuyo valor es la variable de control mas tres
if val3>ctrl_nu_apli%si su valor es mayor a la variable de control
    val3=val3-4;%reduce su valor en 4 unidades
else
    val3=val3;%en caso contrario mantiene su valor
end

x1_dis=ver_orde(val_ctrl,2);%valor en x del vertice 1 ----- x1_discriminacion
y1_dis=ver_orde(val_ctrl,3);%valor en y del vertice 1
z1_dis=ver_orde(val_ctrl,4);%valor en z del vertice 1

x2_dis=ver_orde(val2,2);%valor en x del vertice 2 ----- x1_discriminacion
y2_dis=ver_orde(val2,3);%valor en y del vertice 2
z2_dis=ver_orde(val2,4);%valor en z del vertice 2

x3_dis=ver_orde(val3,2);%valor en x del vertice 2 ----- x1_discriminacion
y3_dis=ver_orde(val3,3);%valor en y del vertice 2
z3_dis=ver_orde(val3,4);%valor en z del vertice 2

% serie(:,contador0)=[val_ctrl val2 val3]%almacena el valor de las variables en una
sola

% contador0=contador0+1;%contador de control de la variable de almacenamiento

for contador_m2=1:tam_ver_or(1)

x1_dif=vertices_unicos2(contador_m2,1);%valor en x del vertice 2
y1_dif=vertices_unicos2(contador_m2,2);%valor en y del vertice 2
z1_dif=vertices_unicos2(contador_m2,3);%valor en z del vertice 2

dist1_búsqueda(contador1)=sqrt((x1_dis-x1_dif)^2+(y1_dis-y1_dif)^2+(z1_dis-
z1_dif)^2);%calculo de la distancia entre
%los dos vertices

% if dist1_búsqueda(contador1)>AF-tolerancia1 &&
dist1_búsqueda(contador1)<AF+tolerancia1

    v_ref1=[x1_dif-x1_dis y1_dif-y1_dis z1_dif-z1_dis];
    v_ref2=[x2_dis-x1_dis y2_dis-y1_dis z2_dis-z1_dis];
    ppunto=sum(v_ref1.*v_ref2);
    n_v1=norm(v_ref1);
    n_v2=norm(v_ref2);

```

```

ang=acosd(ppunto/(n_v1*n_v2));

if ang>=88.5 && ang<=91.5

    coor_est(contador3,:)= [dist1_busqueda(contador1) ang x1_dif y1_dif z1_dif];
    contador3=contador3+1;

end

contador1=contador1+1;

end
% otro=coor_est;

%%%%%%%%%%%%%%
z=find(coor_est(:,1)<=AF+2.5*tolerancia1 & coor_est(:,1)>=AF-2.5*tolerancia1);
coor_est=coor_est(z,:);
%%%%%%%%%%%%%%
% coor_est

tam_fil1=size(coor_est);

for contador_m0=1:tam_fil1(1)

    dist1_busqueda(contador_m0)=sqrt((x1_dis-coor_est(contador_m0,3))^2+(y1_dis-
coor_est(contador_m0,4))^2+(z1_dis-coor_est(contador_m0,5))^2);%calculo de la
distancia entre
    %los dos vertices
    v_ref1=[x3_dis-x1_dis y3_dis-y1_dis z3_dis-z1_dis];
    v_ref2=[coor_est(contador_m0,3)-x1_dis coor_est(contador_m0,4)-y1_dis
coor_est(contador_m0,5)-z1_dis];
    ppunto=sum(v_ref1.*v_ref2);
    n_v1=norm(v_ref1);
    n_v2=norm(v_ref2);
    ang=acosd(ppunto/(n_v1*n_v2));

    angulo(contador_m0)=ang;

%   if dist1_busqueda(contador_m0)<AF+2.5*tolerancia1 &&
dist1_busqueda(contador_m0)>AF-2.5*tolerancia1

end

posiciones=find(angulo>=88.5 & angulo<=91.5);

```

```

    coor_est2=zeros(length(posiciones),5);
    coor_est2(:,1)=dist1_busqueda(posiciones)';
    coor_est2(:,2)=angulo(posiciones)';
    coor_est2(:,3)=coor_est(posiciones,3)';
    coor_est2(:,4)=coor_est(posiciones,4)';
    coor_est2(:,5)=coor_est(posiciones,5)';
    % coor_est2=[dist1_busqueda(posiciones); angulo(posiciones);
    coor_est(posiciones,3); coor_est(posiciones,4); coor_est(posiciones,5);]';

    punto=coor_est2(find(coor_est2(:,1)<AF+1.5*tolerancia1 & coor_est2(:,1)>AF-
    1.5*tolerancia1),:);
    punto(contador5,:)=unique(coor_est2(find(coor_est2(:,1)==max(coor_est2(:,1))),:),'ro
ws');
    [~,mas_cercano]=min(abs(punto(:,1)-AF));

    % punto
    punto=punto(mas_cercano,:);
    % punto=min(punto);
    coor_estm(contador5,:)=punto;
    contador5=contador5+1;
    contador1=1;
    contador2=1;
    contador3=1;
    contador4=1;
    % otro=punto;
    %
    plot3(punto(:,3),punto(:,4),punto(:,5),' ','Color',[rand,rand,rand],'MarkerSize',60);%grafi
ca todos los vertices en la figura con un * ----- total_ver_gui)
clear punto coor_est coor_est2 posiciones dist1_busqueda angulo
% contador5=1;

end

%
plot3(coor_estm(:,3),coor_estm(:,4),coor_estm(:,5),' ','Color',[rand,rand,rand],'Marker
Size',60);%grafica todos los vertices en la figura con un * ----- total_ver_gui)

%%
%-----Agrupación de los vertices superiores con los complementarios-----
contador1=1;

for contador_m0=4:4:tam_ver_orde

    coordenadas_completas(:,:,contador1)=[ver_orde(contador_m0-3:contador_m0,2:4);
    coor_estm(contador_m0-3:contador_m0,3:5)];

```



```

nueva_coorpz=input(coorz_manual);
punto_medio2=[nueva_coorpx,nueva_coorpy,nueva_coorpz];

grafica_puntomedio2=plot3(punto_medio2(:,1),punto_medio2(:,2),punto_medio2(:,3),'
','Color',[.09,.7,.9],'MarkerSize',25);%grafica todos los vertices en la figura con un *
%
var_control_reubicacion=var_control_reubicacion+1;
else
if var_control_reubicacion~=0
boton_control_reubicacion=0;
punto_medio2=punto_medio2;
else
boton_control_reubicacion=0;
punto_medio2=punto_medio_cursor;
end
end
end

end

coor_exportacion=coordenadas_completas-punto_medio2;%reubica las coordenadas
de acuerdo con el punto de referencia definido previamente

plot3(coor_exportacion(:,1,1),coor_exportacion(:,2,1),coor_exportacion(:,3,1),'','Color'
,[rand,rand,rand],'MarkerSize',25)
plot3(coor_exportacion(:,1,2),coor_exportacion(:,2,2),coor_exportacion(:,3,2),'','Color'
,[rand,rand,rand],'MarkerSize',25)
plot3(0,0,0,'','Color',[rand,rand,rand],'MarkerSize',25)

% despliegue de los vertices reubicados y el punto de origen (0,0,0)
figure,hold on,grid on
xlabel('Eje x'),ylabel('Eje y'),zlabel('Eje z');%etiqueta los ejes
plot3(coor_exportacion(:,1,1),coor_exportacion(:,2,1),coor_exportacion(:,3,1),'','Color'
,[rand,rand,rand],'MarkerSize',25)
plot3(coor_exportacion(:,1,2),coor_exportacion(:,2,2),coor_exportacion(:,3,2),'','Color'
,[rand,rand,rand],'MarkerSize',25)
plot3(0,0,0,'','Color',[rand,rand,rand],'MarkerSize',25)

%%
%-----Calculo de angulos de los aplicadores en los planos

num_apli=tam_ver_orde(1)/4;%numero de aplicadores

for contador_m2=1:num_apli

coordenadas_aplicadores=coor_exportacion(:,:,contador_m2);
vertice_original=coordenadas_aplicadores(1,:);

```

```

vertices_referencia(1,1)=vertice_original(1);
vertices_referencia(1,2)=vertice_original(2);
vertices_referencia(1,3)=vertice_original(3);

vertices_referencia(2,1)=vertice_original(1)+AD;
vertices_referencia(2,2)=vertice_original(2);
vertices_referencia(2,3)=vertice_original(3);

vertices_referencia(3,1)=vertice_original(1);
vertices_referencia(3,2)=vertice_original(2);
vertices_referencia(3,3)=vertice_original(3)+AB;

vertices_referencia(4,1)=vertice_original(1);
vertices_referencia(4,2)=vertice_original(2)-AF;
vertices_referencia(4,3)=vertice_original(3);

for contador_m1=2:8

    distancia_vertices=sqrt((coordenadas_aplicadores(1,1)-
coordenadas_aplicadores(contador_m1,1))^2+(coordenadas_aplicadores(1,2)-
coordenadas_aplicadores(contador_m1,2))^2+(coordenadas_aplicadores(1,3)-
coordenadas_aplicadores(contador_m1,3))^2);

    if distancia_vertices>AB-2.5*tolerancia1 && distancia_vertices<AB+2.5*tolerancia1
        distancia_ABmm=coordenadas_aplicadores(contador_m1,:);
    elseif distancia_vertices>AD-2.5*tolerancia1 &&
distancia_vertices<AD+2.5*tolerancia1
        distancia_ADmm=coordenadas_aplicadores(contador_m1,:);
    elseif distancia_vertices>AF-2.5*tolerancia1 &&
distancia_vertices<AF+2.5*tolerancia1
        distancia_AFmm=coordenadas_aplicadores(contador_m1,:);
    end

end

figure,hold on,grid on
npos=plot3(coordenadas_aplicadores(1:8,1),coordenadas_aplicadores(1:8,2),coorde
nadas_aplicadores(1:8,3),'.','Color',[rand,rand,rand],'MarkerSize',25);%grafica todos
los vertices en la figura con un *
vref=plot3(vertices_referencia(:,1),vertices_referencia(:,2),vertices_referencia(:,3),'!','
Color',[rand,rand,rand],'MarkerSize',25);%grafica todos los vertices en la figura con
un *
axis([-1000 1000 -1000 1000 -1000 1000])%coloca valores maximos y minimos en
los ejes (x,y & z)

```

```
% lg(i)=legend(sprintf('Aplicador %d', i))%coloca etiqueta en los vertices reubicados
```

```
% lg=legend(apli,'aplicador ')  
xlabel('Eje x'),ylabel('Eje y'),zlabel('Eje z');%etiqueta los ejes  
view([-135 35]);%vista del objeto
```

```
% plano xy
```

```
vector1=distancia_ABmm(1:2)-vertice_original(1:2);  
vector2=vertices_referencia(3,1:2)-vertice_original(1:2);  
norma_vector1=norm(vector1);  
norma_vector2=norm(vector2);  
producto_puntp=sum(vector1.*vector2);  
producto_normas=norma_vector1*norma_vector2;  
theta1(1,1)=acosd(producto_puntp/producto_normas);
```

```
% plano xz
```

```
vector1=distancia_ABmm(1:2:3)-vertice_original(1:2:3);  
vector2=vertices_referencia(3,1:2:3)-vertice_original(1:2:3);  
norma_vector1=norm(vector1);  
norma_vector2=norm(vector2);  
producto_puntp=sum(vector1.*vector2);  
producto_normas=norma_vector1*norma_vector2;  
theta1(1,2)=acosd(producto_puntp/producto_normas);
```

```
% plano yz
```

```
vector1=distancia_ABmm(2:3)-vertice_original(2:3);  
vector2=vertices_referencia(3,2:3)-vertice_original(2:3);  
norma_vector1=norm(vector1);  
norma_vector2=norm(vector2);  
producto_puntp=sum(vector1.*vector2);  
producto_normas=norma_vector1*norma_vector2;  
theta1(1,3)=acosd(producto_puntp/producto_normas);
```

```
% plano xy
```

```
vector1=distancia_ADmm(1:2)-vertice_original(1:2);  
vector2=vertices_referencia(2,1:2)-vertice_original(1:2);  
norma_vector1=norm(vector1);  
norma_vector2=norm(vector2);  
producto_puntp=sum(vector1.*vector2);  
producto_normas=norma_vector1*norma_vector2;  
theta1(2,1)=acosd(producto_puntp/producto_normas);
```

```
% plano xz
```

```
vector1=distancia_ADmm(1:2:3)-vertice_original(1:2:3);  
vector2=vertices_referencia(2,1:2:3)-vertice_original(1:2:3);  
norma_vector1=norm(vector1);
```

```
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(2,2)=acosd(producto_puntp/producto_normas);
```

% plano yz

```
vector1=distancia_ADmm(2:3)-vertice_original(2:3);
vector2=vertices_referencia(2,2:3)-vertice_original(2:3);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(2,3)=acosd(producto_puntp/producto_normas);
```

% plano xy

```
vector1=distancia_AFmm(1:2)-vertice_original(1:2);
vector2=vertices_referencia(4,1:2)-vertice_original(1:2);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(3,1)=acosd(producto_puntp/producto_normas);
```

% plano xz

```
vector1=distancia_AFmm(1:2:3)-vertice_original(1:2:3);
vector2=vertices_referencia(4,1:2:3)-vertice_original(1:2:3);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(3,2)=acosd(producto_puntp/producto_normas);
```

% plano yz

```
vector1=distancia_AFmm(2:3)-vertice_original(2:3);
vector2=vertices_referencia(4,2:3)-vertice_original(2:3);
norma_vector1=norm(vector1);
norma_vector2=norm(vector2);
producto_puntp=sum(vector1.*vector2);
producto_normas=norma_vector1*norma_vector2;
theta1(3,3)=acosd(producto_puntp/producto_normas);
```

```
theta(:, :, contador_m2)=theta1(:, :, :);
```

```
end
```

```
toc
```