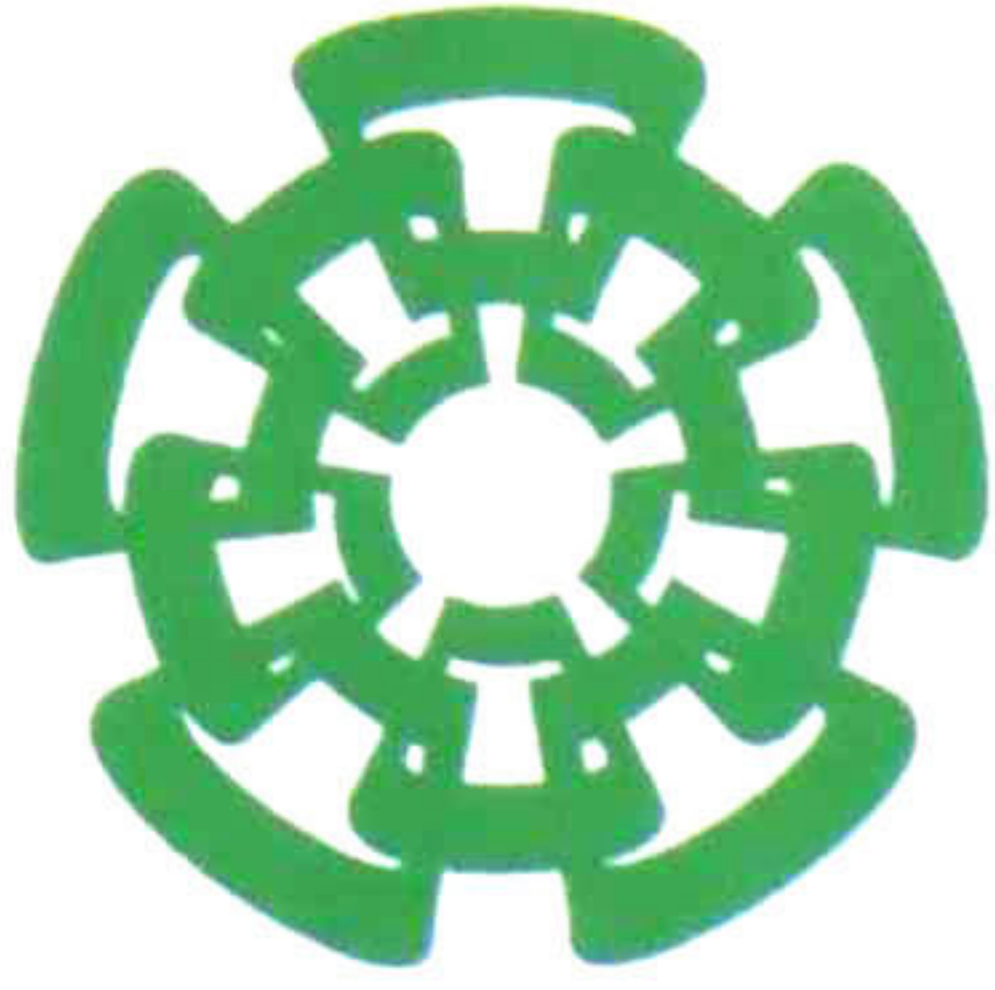




CT-858-551

Jan. 2015



Centro de Investigación y de Estudios Avanzados  
del Instituto Politécnico Nacional  
Unidad Guadalajara

# **Modelo Neuronal de Percepción Táctil y Visual con Aplicaciones Hápticas para Sistemas de Cirugía de Mínima Invasión**

Tesis que presenta:  
**Efraín Castillo Muñiz**

para obtener el grado de:  
**Doctor en Ciencias**

en la especialidad de:  
**Ingeniería Eléctrica**

Director de Tesis  
**Dr. Eduardo José Bayro Corrochano**

**CINVESTAV  
IPN  
ADQUISICION  
LIBROS**

CLASIF.. CT00739  
ADQUIS.. CT-838-SP1  
FECHA: 18-06-2015  
PROCED.. DON: 2015  
\$ \_\_\_\_\_

# **Modelo Neuronal de Percepción Táctil y Visual con Aplicaciones Hápticas para Sistemas de Cirugía de Mínima Invasión**

**Tesis de Doctorado en Ciencias  
Ingeniería Eléctrica**

Por:

**Efraín Castillo Muñiz**

Ingeniero en Comunicaciones y Electrónica

Universidad de Guadalajara 1998-2002

Becario de CONACYT, expediente no. 46666

Director de Tesis

**Dr. Eduardo José Bayro Corrochano**

CINVESTAV del IPN Unidad Guadalajara, Junio de 2014.

*DEDICO ESTA TESIS A MIS BEBES,  
DISCULPEN POR NO SER EL PADRE QUE MERECEAN,  
ESPERO QUE APRENDAN DE MÍ A VIVIR PARA CUMPLIR SUS SUEÑOS.*

## AGRADECIMIENTOS

Primeramente debo agradecer al Prof. Eduardo Bayro por haberme aceptado como su alumno y luego por su infinita paciencia, de corazón gracias. Después de eso debo agradecer a CINVESTAV por ser la institución que es, el lugar que me dio la oportunidad de crecer como persona y como profesional, ojalá y que siga existiendo para que le cambie la vida de manera positiva a muchas personas como yo. También gracias a CONACYT por haberme otorgado las becas que me permitieron estudiar mi licenciatura, maestría y ahora mi doctorado.

La última vez que escribí algo así quise mencionar a cada persona con la que conviví, pero sin querer dejé de mencionar a algunos por lo que no volveré a cometer ese error, debo decir que en mi estancia en CINVESTAV hice muchos amigos que se lo serán para toda la vida, les agradezco a todos las lecciones que me dejaron, fue un placer el tiempo compartido y les aseguro los llevo a todos en mi corazón. También agradezco a todo el personal del CINVESTAV, desde el director, los investigadores, las secretarias, la gente de contabilidad, las personas de apoyo, los intendentes, los oficiales, la gente de la cocina, a todos muchas gracias, sigan haciendo su labor tan bien como hasta hoy.

Finalmente no puedo dejar de mencionar a dos amigos que cambiaron mi vida, Filemón y Ulises, con los que aparte de compartir su amistad compartieron mi sueño de crear una empresa, disculpen por todo lo que los hago pasar, espero que sigamos creciendo en este sueño compartido.

Efraín.

# Índice

<b>LISTA DE FIGURAS .....</b>	<b>3</b>
<b>LISTA DE SÍMBOLOS .....</b>	<b>7</b>
<b>LISTA DE SÍMBOLOS DE ÁLGEBRA GEOMÉTRICA .....</b>	<b>7</b>
<b>1. INTRODUCCIÓN .....</b>	<b>10</b>
1.1 CIRUGÍA DE MÍNIMA INVASIÓN.....	12
1.2 INTERFACES HÁPTICAS.....	15
1.3 CÁMARAS RGB-D.....	18
1.4 OBJETIVOS DEL TRABAJO.....	19
1.5 ESTRUCTURA DEL DOCUMENTO .....	20
<b>2. ANTECEDENTES MATEMÁTICOS .....</b>	<b>21</b>
2.1 ÁLGEBRA GEOMÉTRICA .....	21
<b>3. REDES NEURONALES.....</b>	<b>40</b>
3.1 REDES DE BASE RADIAL.....	41
3.2 REDES NEURONALES GENERALES PARA REGRESIÓN.....	43
3.3 REDES GENERALES ESFÉRICAS PARA REGRESIÓN.....	46
3.3 MODELO NEURONAL TACTO-VISUAL .....	48
<b>4. ESCENARIO DE LABORATORIO: SISTEMA DE CIRUGÍA DE MÍNIMA INVASIÓN.....</b>	<b>54</b>
4.1 BRAZOS ROBOT.....	54
4.2 INTERFACES HÁPTICAS.....	58
4.3 ESCENARIO PARA PRUEBA DEL MODELO .....	60
<b>5. RESULTADOS EXPERIMENTALES Y SIMULACIONES .....</b>	<b>63</b>
5.1 REDES GENERALES ESFÉRICAS PARA REGRESIÓN.....	63
5.2 IMPLEMENTACIÓN DEL MODELO NEURONAL TACTO-VISUAL .....	70
5.3 PRUEBA EN LABORATORIO DEL MODELO NEURONAL TACTO-VISUAL .....	74
<b>6. CONCLUSIONES Y TRABAJO A FUTURO .....</b>	<b>88</b>
6.1 CONCLUSIONES.....	88
6.2 TRABAJO A FUTURO .....	89
<b>7. PUBLICACIONES.....</b>	<b>91</b>
<b>7 ANEXOS.....</b>	<b>93</b>
7.1 COMPARATIVA DE DIFERENTES INTERFACES HÁPTICAS COMERCIALES.....	93
7.2 ROBOTS MOTOMAN.....	96
7.3 CÁMARA RGB -D .....	97
7.4 SENSOR DE FUERZA .....	97
7.5 PROGRAMA EN CLUCALC V4.3.2 .....	98
<b>BIBLIOGRAFÍA.....</b>	<b>104</b>



# Lista de Figuras

Figura 1. Sistema de Cirugía DaVinci desarrollado y comercializado por la empresa INTUITIVE SURGICAL, el cual consiste de una mesa quirúrgica unida a 4 brazos robot, una consola o interfaz de control y una pantalla. ....	12
Figura 2. Esquema de un sistema asistente para cirugías de mínima invasión (tomado de [2] ) .....	13
Figura 3. Se muestra en a) el Titan Surgical System y en b) el Zeus Surgical System.....	14
Figura 4. Interfaz háptica para "tocar" objetos virtuales .....	16
Figura 5. Clasificación de interfaces hápticas.....	17
Figura 6. Interfaz háptica tipo joystick de 6 grados de libertad de SENSABLE Inc (1).....	18
Figura 7. Sensor RGB-D PrimeSense RD 1.09 .....	19
Figura 8. Bivector representado como un plano orientado.....	23
Figura 9. Sentido del bivector .....	23
Figura 10. Proyección estereográfica de un punto en un círculo unitario.....	27
Figura 11. Vectores base y líneas nulas del plano de Minkowski .....	29
Figura 12. El modelo homogéneo para la proyección estereográfica en el caso 1D. ....	30
Figura 13. Esfera construida con centro en $p_e$ y radio $\rho$ .....	31
Figura 14. Plano definido por 3 puntos que yacen en él y el punto en el infinito. ....	32
Figura 15. Círculo formado por tres puntos o la intersección de dos esferas. ....	33
Figura 16. Línea mapeada en la esfera unitaria. ....	34
Figura 17. El par de puntos como la intersección de una línea y una esfera.....	35
Figura 18. Punto generado por su proyección estereográfica en la esfera unitaria.....	36
Figura 19. Reflexión de un punto respecto a un plano. ....	37
Figura 20. Reflexión entre planos paralelos, la traslación .....	38
Figura 21. Reflexión entre planos no paralelos, la rotación.....	39
Figura 22. Estructura de una red de base radial (RBFN) .....	41
Figura 23. Esquema de entrenamiento para una RBFN .....	42

<b>Figura 24. Estructura de una GRNN .....</b>	<b>43</b>
<b>Figura 25. Función desconocida a aproximar.....</b>	<b>45</b>
<b>Figura 26. Kernels propuestos para el entrenamiento de la red .....</b>	<b>46</b>
<b>Figura 27. Se debe garantizar que <math>2b &gt; d_{max}</math> para garantizar el traslape entre círculos.....</b>	<b>48</b>
<b>Figura 28. Objeto tridimensional en forma de elipsoide. ....</b>	<b>48</b>
<b>Figura 29. Modelo discreto de la rigidez de la superficie de un cuerpo tridimensional.....</b>	<b>49</b>
<b>Figura 30. Se muestra un objeto con una protuberancia superficial.....</b>	<b>50</b>
<b>Figura 31. Ejemplo del uso del sphere packing algorithm (tomado de [23])......</b>	<b>51</b>
<b>Figura 32. Modelo Neuronal Tacto Visual.....</b>	<b>53</b>
<b>Figura 33. Se muestra el resultado final que se obtiene al aplicar el Modelo Neuronal Tacto-Visual .....</b>	<b>53</b>
<b>Figura 34. Robot ADEPT 6 600, brazo robot de 6 grados de libertad .....</b>	<b>55</b>
<b>Figura 35. Brazo robot MH5LF de MOTOMAN.....</b>	<b>56</b>
<b>Figura 36. Arreglo de brazos robot propuesto para sistema de cirugía de mínima invasión .....</b>	<b>57</b>
<b>Figura 37. Distribución espacial de sistema de cirugía de mínima invasión .....</b>	<b>58</b>
<b>Figura 38. Omi haptic device, interfaz háptica de 6 grados de libertad de SENSABLE Inc.....</b>	<b>59</b>
<b>Figura 39. Esquema usado para el experimento.....</b>	<b>61</b>
<b>Figura 40. Phantom construido para experimento .....</b>	<b>62</b>
<b>Figura 41. Conjunto de puntos interpolados con la GRNN (arriba) y con la SGRNN (abajo) .....</b>	<b>64</b>
<b>Figura 42. Polinomio propuesto para medir desempeño de la GRNN vs. la SGRN.....</b>	<b>65</b>
<b>Figura 43. Se muestra a la izquierda la interpolación de un polinomio usando un muestreo uniforme de 8 puntos, con la GRNN y con la SGRN, a la derecha lo mismo pero usando un muestreo uniforme de 12 puntos .....</b>	<b>65</b>
<b>Figura 44. Se muestra en (a) el histograma de MSE al interpolar con 20 puntos seleccionados de forma aleatoria 100 veces, en (b) 2 ejemplos de interpolación tomados del conjunto de 100. ....</b>	<b>66</b>
<b>Figura 45. Error cuadrático Medio (MSE) al variar el radio de los círculos en la interpolación.....</b>	<b>67</b>
<b>Figura 46. (a) Malla de un hígado humano, (b) nube de puntos del hígado.....</b>	<b>68</b>

<b>Figura 47. Nubes de puntos del hígado.....</b>	<b>69</b>
<b>Figura 48. Cortes sagitales y axiales del hígado .....</b>	<b>69</b>
<b>Figura 49. Se muestra un corte sagital (a) y un corte axial (b); también se muestra en (c) y (d) la reconstrucción usando la SGRN de los cortes sagital y coronal respectivamente (los puntos rojos representan los obtenidos con la SGRN). .....</b>	<b>70</b>
<b>Figura 50. Reconstrucción de un objeto 2D usando círculos.....</b>	<b>72</b>
<b>Figura 51. Reconstrucción de un objeto tridimensional. Se pueden observar esferas de diferentes radios al centro del objeto .....</b>	<b>72</b>
<b>Figura 52. Se muestra la reconstrucción de un mismo objeto pero con diferentes valores de delta, en a) 0.2, en b) 0.3 y en c) 0.4 .....</b>	<b>73</b>
<b>Figura 53. Se muestra en a) la nube de puntos 3D obtenida originalmente con la cámara y en b) solamente el área de interés que contiene al phantom .....</b>	<b>76</b>
<b>Figura 54. Se muestra la nube de puntos obtenida con la cámara RGB-D del phantom en a) una vista lateral, en b) una vista desde superior. ....</b>	<b>76</b>
<b>Figura 55. Se muestra la nube de puntos obtenida con la cámara RGB-D del phantom (azul) y los puntos "tocados" con el sensor de tacto (rojo) en a) una vista lateral, en b) una vista desde superior .....</b>	<b>77</b>
<b>Figura 56. Vista lateral de la nube de puntos de superficie del phantom (azul) con los puntos .....</b>	<b>78</b>
<b>Figura 57.- Nube de puntos reducida al 10% (azul) y puntos muestra para cálculo de la rigidez (rojo).....</b>	<b>79</b>
<b>Figura 58. Nube de puntos trasladada al origen y normalizada .....</b>	<b>79</b>
<b>Figura 59. Se muestra la reconstrucción de la nube de puntos con esferas tomando diferentes valores de delta, en a) delta=0.15, en b) delta=0.1 y en c) delta=0.05.....</b>	<b>81</b>
<b>Figura 60. Se muestra en (a) la nube de esferas representativa de las superficies, y en b) la nube de puntos reducida (cada punto representa una esfera) .....</b>	<b>82</b>
<b>Figura 61. Se muestra desde la parte de arriba(a) la nube de puntos reducida (rojo) junto con los puntos muestrales donde se conoce la rigidez superficial (azul). En (b) la misma figura desde una perspectiva lateral.....</b>	<b>83</b>
<b>Figura 62. Varias vistas de la función de rigidez superficial obtenida por la red .....</b>	<b>83</b>
<b>Figura 63. Se muestran zonas correspondientes entre el phantom y la función de rigidez superficial .....</b>	<b>84</b>

**Figura 64. Se muestra la nube de esferas que recrea la superficie del phantom, las esferas de color rojo más intenso representan zona de mayor rigidez superficial y viceversa. .... 85**

**Figura 65. Esquema de renderizado háptico..... 86**

**Figura 66. Esquema de renderizado háptico cuando el puntero intersecta más de una esfera. .... 87**

**Figura 67. El efector final de un robot puede ser relacionado con esferas individuales representadas en nuestra red usando el mismo lenguaje geométrico. .... 89**

## Lista de Símbolos

$q \text{ ó } \theta$	Vector de posiciones angulares
$w \text{ ó } \dot{\theta}$	Vector de velocidades angulares
$J \bullet$	Matriz Jacobiana
$\tau$	Esfuerzo de torsión
$\mathbb{R}^n$	Espacio n-dimensional de los reales
$\lim_{x \rightarrow y} z \ x$	Límite de $z \ x$ cuando $x$ tiende a $y$
$x_{ref}$	Señal de referencia para la variable $x$
$\ \bullet\ $	Norma-2 del término $\bullet$
$x_k$	Valor de la variable $x$ en la $k^{\text{ésima}}$ muestra
$T$	Período de muestreo.
$\text{sign } \bullet$	Función signo
$\text{sigm } \bullet$	Función sigmoidea, en este trabajo $\tanh \bullet$
$A^T$	Transpuesta de la matriz $A$
$A^{-1}$	Inversa de la matriz $A$
$A^+$	Pseudo-inversa de la matriz $A$
$\text{rank } A$	Rango de la matriz $A$

## Lista de Símbolos de Álgebra Geométrica

$x_e$	Punto expresado en el espacio euclidiano
$x$	Punto expresado en el espacio conformal
$M$	Motor
$L_x$	Línea $x$ expresada en $G_{4,1}$
$\pi_x$	Plano $x$ expresado en $G_{4,1}$
$R$	Rotor expresado en $G_{4,1}$
$G_{p,q,r}$	Álgebra geométrica con bases $p$ , $q$ y $r$
$\wedge$	Producto exterior en álgebra geométrica
$\sim$	Reversión
$T$	Traslador expresado en $G_{4,1}$
$e_0$	Punto en el origen del espacio conformal
$e_\infty$	Punto en el infinito del espacio conformal

## RESUMEN

Esta tesis se avoca al desarrollo de redes neuronales geométricas útiles para la fusión y procesamiento de datos capturados por cámaras digitales y sensores. Estas redes tienen una variedad de posibles aplicaciones como ser en ingeniería de gráficos, realidad virtual, visión computacional, robótica y procesamiento de imágenes médicas.

Las redes neuronales geométricas se desarrollaron e implementaron usando como lenguaje matemático el Álgebra Geométrica Conformal debido a que en esta Álgebra las esferas se representan como un vector de grado uno facilitando en gran medida su manipulación, además de las grandes ventajas de representar otros objetos geométricos en el mismo sistema matemático.

Como primer aporte se desarrolló y probó una nueva red neuronal geométrica también en el lenguaje del Álgebra Geométrica Conformal que llamamos *Red General Esférica para Regresión*, la cual está basada en la muy conocida *Red Neuronal General para Regresión* (GRNN por sus siglas en inglés).

El segundo y más importante aporte de esta tesis es el desarrollo de un modelo que a partir de una nube de puntos de un objeto y de conocer la rigidez en algunos puntos sobre la superficie del mismo, es capaz de representar de manera conjunta su estructura tridimensional (representación visual) además de la rigidez sobre cualquier punto de su superficie. Esto se logra a través de la representación del objeto con esferas de diferentes radios y que además no se traslapan entre sí. además de una red neuronal que interpola la rigidez del objeto a lo largo y ancho de la superficie siendo capaz de interpolar tales valores de rigidez sobre cada esfera de la representación estructural del objeto, llamamos a este *Modelo Neuronal Tacto-Visual*.

Gracias a la representación lograda con esferas, se muestra que es posible la interacción virtual con el objeto reconstruido a través de un simple renderizado háptico, esto abre grandes posibilidades en el área de realidad virtual en las aplicaciones médicas. La motivación de este trabajo viene de la medicina. el de representar órganos que presentan tumoraciones superficiales visibles o no visibles y que pudieran ayudar al cirujano en la etapa quirúrgica como un complemento de la imagenología médica. Se presenta un experimento de laboratorio donde se comprueba el potencial del modelo y para el cual se montó un escenario que pretende ser la base para la construcción de un sistema de cirugía de mínima invasión de bajo costo.

## ABSTRACT

This thesis focus in the development of geometric neural networks for the fusion and processing of data captured with digital cameras and sensors. These networks have a variety of applications namely in graphics engineering, virtual reality, computer vision, robotics and medical image processing.

The geometric neural networks were developed and implemented using the Conformal Geometric Algebra mathematical language because in this Algebra spheres are represented as a vector of degree one greatly facilitating its manipulation, besides the great advantages of representing other geometric objects in this mathematical framework.

The first contribution is a new geometric neural network developed in the language of Algebra Geometric Conformal that we call Spherical General Regression Network, which is based on the well known General Regression Neural Network (GRNN).

The second and more important contribution of this thesis is the development of a combined network model which uses as input a point cloud of an object and stiffness values at some points on the body surface, based on this data. it is able to represent jointly the three-dimensional structure (visual representation) and in addition interpolate the stiffness at any point on the surface. This is achieved by representing the object with spheres of different radii which not overlap each other. in addition, at the output of the first layer of the network, a second neural network interpolates the object stiffness of the surface. This combined network is able to assign the stiffness value to each sphere of the structural representation of the object. We call this Touch-Visual Neuronal Model.

Thanks to the representation achieved with spheres, it shows that it is possible to interact with the virtual object through a reconstructed single haptic rendering, this opens up great possibilities in the area of virtual reality. The motivation for this work comes from medicine, to represent bodies with visible or non-visible surface tumors to assist the surgeon in surgical stage. A laboratory experiment is presented where the potential of the model is tested and for which an scenario was built that serves as a preliminary prototype for building in near future an inexpensive minimally invasive surgery system.

# 1. Introducción

El ser humano cuenta con cinco sentidos: la visión, el tacto, el oído el olfato y el gusto y es bien sabido que el cerebro es el encargado de coordinar y controlar estas funciones en conjunto con las actividades motrices en los animales, incluyendo al ser humano. Por otro lado, la creación de máquinas inteligentes es un tema que ha fascinado a muchos de los científicos más notables de todos los tiempos, intentando realizar esto a través de diferentes ramas de la ciencia como la biología, la medicina, la electrónica, la mecánica y en tiempos más recientes lo que se ha llamado la cibernética. Ésta última se trata de una disciplina donde se engloban de manera interdisciplinaria varias ramas de la ciencia, estando íntimamente relacionada con la teoría de control y la teoría de sistemas.

Los sistemas bioinspirados son, como su nombre lo indica, sistemas (ya sea desarrollados con software, con hardware o ambos) que intentan emular el comportamiento del ser humano o de algún otro ser vivo en cuanto a su forma de pensar, a su forma de actuar o de procesar la información. Desde hace aproximadamente tres siglos pero de manera mucho más acelerada los últimos cincuenta años el hombre ha creado este tipo de sistemas con dos finalidades principales; entender cada vez mejor los mecanismos biológicos que hacen funcionar



al cerebro y crear sistemas que resuelvan problemas de la vida real y que puedan adaptarse lo mejor a su entorno. en específico actualmente hay un gran esfuerzo por crear este tipo de sistemas para que pueda interactuar de manera natural en entornos humanos [1].

El principal aporte de este trabajo es el de crear un modelo basado en redes neuronales que integra dos sentidos humanos (también presentes en muchos animales), la percepción y el tacto. Estos sentidos son quizás los más usados para la creación de sistemas cibernéticos, pero en general se usan *de manera independiente* en dispositivos inclusive patentados como [2]. Existen ya estudios biológicos muy complejos que intentan entender como el cerebro combina específicamente estos dos sistemas sensoriales [3]. o más recientemente acerca de la fusión de sistemas multisensoriales [4].

La forma como las criaturas vivas representan en su cerebro el mundo externo parece estar definido en un sentido amplio en términos geométricos, lo cual sugiere que los investigadores deben buscar sistemas matemáticos con poderosas características geométricas y algebraicas[5]. Basados en esto, hemos decidido proponer nuestros modelos usando como lenguaje el Algebra Geométrica Conformal.

El desarrollo científico y tecnológico en México es una de las preocupaciones actuales tanto del gobierno como de la población en general. En el ámbito de la medicina, así como muchos otros (industria de los alimentos, manufactura, etc.) somos dependientes de los conocimientos que se generan en otros países, y la mayor parte del tiempo vemos que estos conocimientos nos son vendidos a un precio tan alto, que muy pocas personas pueden acceder a sus beneficios. Aparte del aporte científico ya mencionado. este trabajo pretende sentar las bases para el inicio de la construcción de un sistema de cirugía de mínima invasión de bajo costo, el cual pueda asistir a cirujanos en operaciones de abdomen, principalmente las relacionadas con el riñón. tales como remoción de tumores. obstrucción de vías urinarias, trasplante. entre otros. Aunque este tipo de sistemas ya existe de forma comercial. el precio es tan alto que es casi imposible que sus grandes beneficios lleguen a la mayoría de las personas, en nuestro caso pretendemos que a mediano

plazo este sistema pudiera estar funcionando en clínicas públicas de salud alrededor de este país.

## 1.1 Cirugía de Mínima Invasión

La cirugía de mínima invasión tiene sus inicios en los años 90, después que la empresa Intuitive Surgical [6], usando como base un sistema quirúrgico desarrollado por la armada de los EEUU como un intento para que un médico pudiera operar a distancia a un soldado en el campo de batalla, desarrolló un sistema robotizado que podía ser manipulado por un médico entrenado y que podría ayudar al mismo en algunos tipos de operaciones quirúrgicas, este sistema se muestra en la Figura 1.



Figura 1. Sistema de Cirugía DaVinci desarrollado y comercializado por la empresa INTUITIVE SURGICAL, el cual consiste de una mesa quirúrgica unida a 4 brazos robot, una consola o interfaz de control y una pantalla.

Como se puede observar, el sistema se compone de 4 brazos robots, los cuales pueden ser manipulados por un médico a través de una consola, y cuenta además con una pantalla para visualizar la operación. Desde entonces y a la fecha, el usar este tipo de sistemas como apoyo al trabajo del cirujano ha ido creciendo alrededor del mundo, principalmente en operaciones de abdomen (hígado, riñones, etc.) aunque también se propone ya su uso en neurocirugía [7].

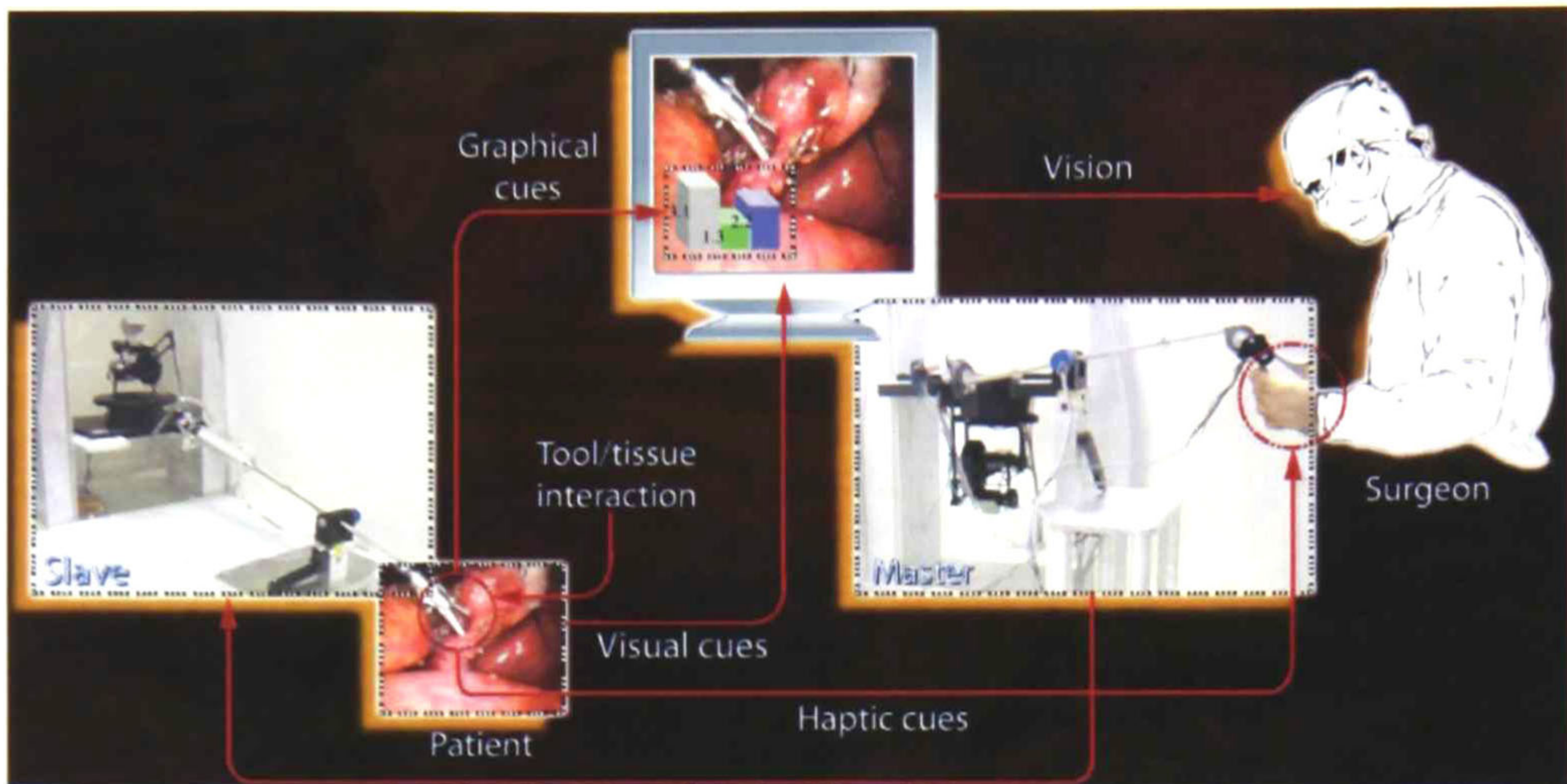


Figura 2. Esquema de un sistema asistente para cirugías de mínima invasión (tomado de [2] )

El principio de la cirugía de mínima invasión es simple, si un médico debe realizar una incisión para introducir sus manos en el cuerpo de un paciente para con ellas explorar, cortar, suturar o cualquier acción que se requiera en la cirugía, tal incisión deberá ser mínimo del tamaño de sus manos, pero si pudiera tener una vista interna del cuerpo, y manos “más pequeñas” para realizar tales acciones, entonces las incisiones serían más pequeñas; lo que se hace en cirugía de mínima invasión es que los brazos robot tienen pequeños efectores finales, lo cuales manipulados por un médico a través de una interfaz puedan realizar dichas tareas (exploración, suturación, etc.) [8]. En el caso de esta interface, normalmente se considera el uso de una interfaz háptica, que es aquella que es capaz de transmitir al médico sensaciones táctiles, esto se logra colocando un sensor de fuerza en el brazo robot y después interpretando la información que envía este sensor a la computadora y procesando tal señal para enviarla a la interfaz aquí mencionada. La parte de la visión se reemplaza por una pequeña cámara endoscópica, la cual se proyecta a una pantalla y con esto el médico puede ver dentro del cuerpo del paciente.

Otros sistemas similares han sido creados después de la aparición del *DaVinci*, por ejemplo el *Titan Surgical System*, de la empresa Titan Medical Inc y el *Zeus Surgical System* de la empresa Computer Motion

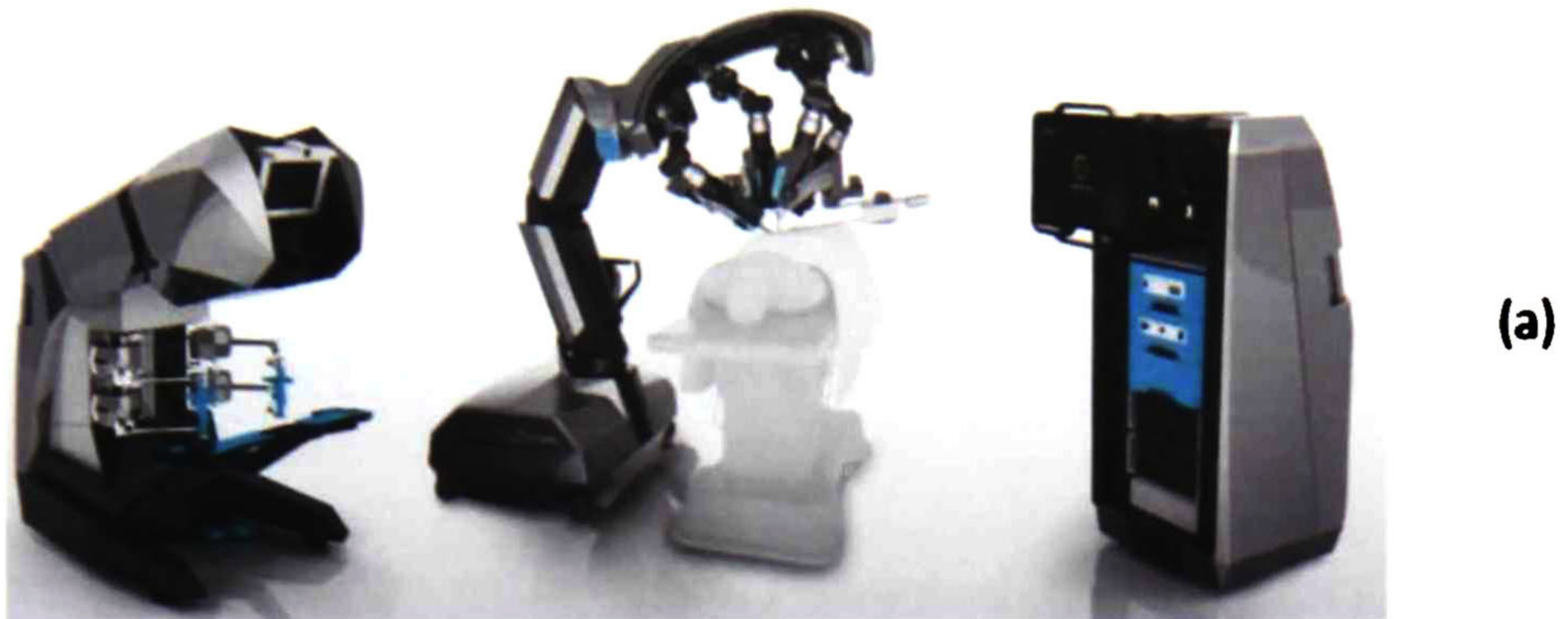


Figura 3. Se muestra en a) el Titan Surgical System y en b) el Zeus Surgical System

Se reportan las siguientes ventajas cuando un paciente se somete a una operación usando cirugía de mínima invasión contra la cirugía tradicional [9]:

*Menor dolor posterior a la cirugía:* permite a los pacientes levantarse y caminar a las pocas horas de la cirugía.

*Estancia hospitalaria corta:* si no hay complicaciones, el paciente puede requerir de uno a tres días en el hospital, después partir a su casa y continuar allí la convalecencia.

*Menor tasa de infección:* los tejidos internos del cuerpo no son expuestos al aire ambiental, a diferencia de la cirugía abierta. Además, la video-

magnificación permite manejarlos en forma más precisa y delicada, protegiendo órganos vitales.

*Reincorporación a la actividad normal:* laboral y de la casa, al ser todo mucho más benigno, se vuelve a la vida diaria más rápidamente.

*Mejora el resultado estético:* las cicatrices son más pequeñas y menos visibles.

Se ha criticado el uso de la cirugía robótica al afirmar que en general la naturaleza ha dotado al ser humano de extraordinarias características que difícilmente podrían ser superadas por máquinas, en algunos casos se ha demostrado que algunas máquinas tienen interfaces que son extremadamente complejas y poco naturales de usar, en cuyo caso la posible ventaja que tendría el uso de robots sería superado por la desventaja del tiempo de entrenamiento de los médicos en su uso, es por ello que la filosofía actual en el desarrollo de tales sistemas no tiene el afán de sustituir a los seres humanos, si no de *complementar* sus habilidades con dispositivos tecnológicos, es decir, integrar lo mejor de las capacidades que ofrece tanto el ser humano como las máquinas, pero ya se han propuesto sistemas que incluso superan por exactitud a la habilidad del cirujano[7].

Otro aspecto que se debe resaltar en cuanto al uso de la cirugía de mínima invasión, es que este tipo de sistemas asistentes para el médico tienen un costo económico altísimo, haciendo muy difícil que sea adquirido por hospitales de escasos recursos.

## **1.2 Interfaces hápticas**

Como se mencionó antes, el control de los brazos robóticos por el cirujano se realiza a través de un sistema teleoperado maestro-esclavo de fuerza-reflectiva [8], esto se logra gracias al uso de una interfaz háptica.

Se entiende por una interfaz háptica a cualquier interfaz que es capaz de transmitir sensaciones táctiles al usuario (fuerza, vibración, calor, etc.). El término háptico viene de la palabra griega hapteshai que significa tocar [10]. Este tipo de interfaces se comenzaron a conceptualizar y después a construir durante la década de los 70 y los 80, y se usaron para diferentes aplicaciones en robótica y videojuegos principalmente. Después, en la década de los 90, estas se comenzaron a usar en los dispositivos de cirugía robótica.

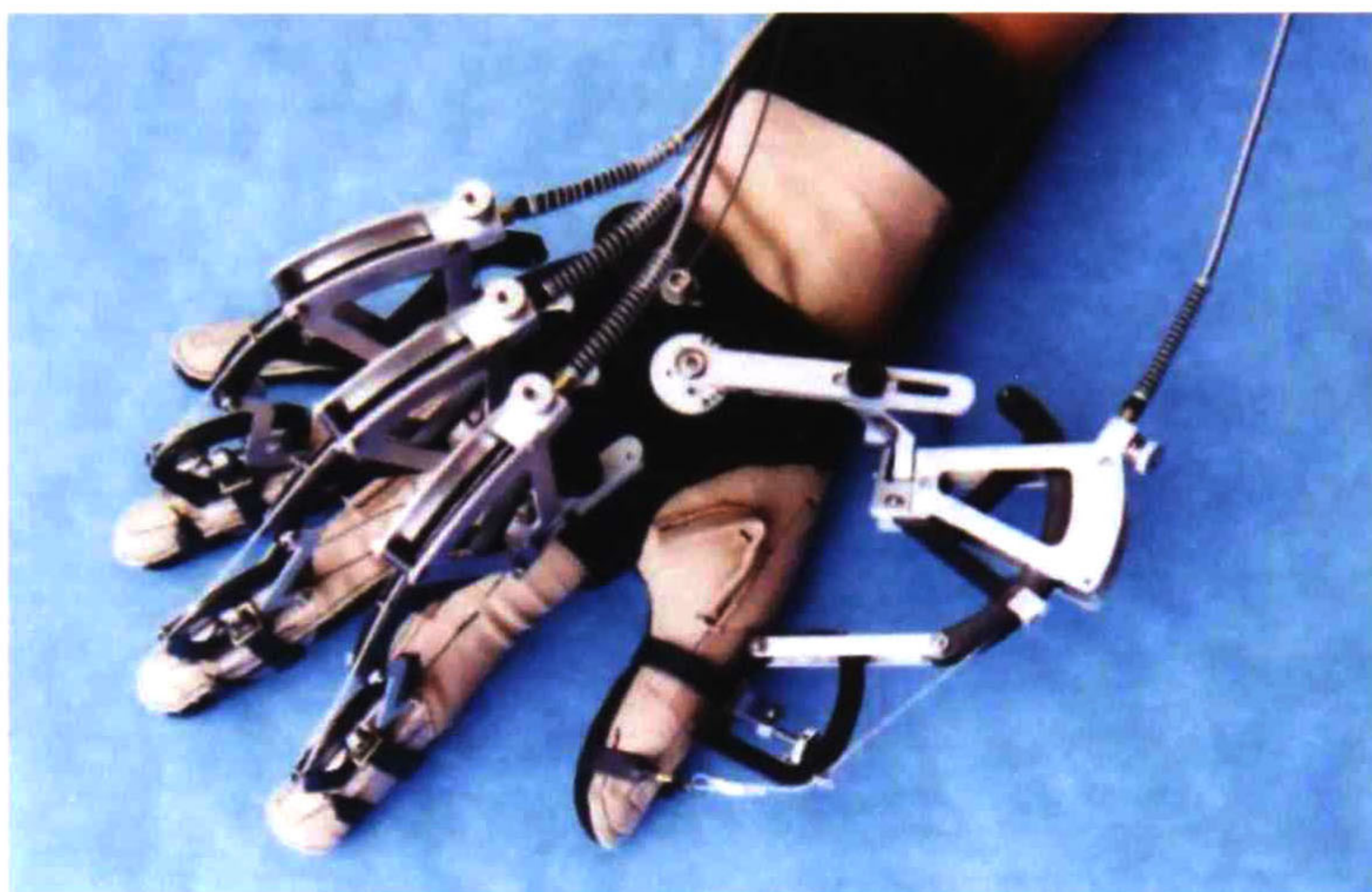


Figura 4. Interfaz háptica para "tocar" objetos virtuales

Las interfaces hápticas se pueden clasificar en función del tipo de actuadores utilizados, la escala en generación de fuerza, su portabilidad o soporte, pudiendo ser interfaces de escritorio, suspendidas en el techo, fijos a la pared o portátiles [11], tal como se puede ver en la Figura 5.

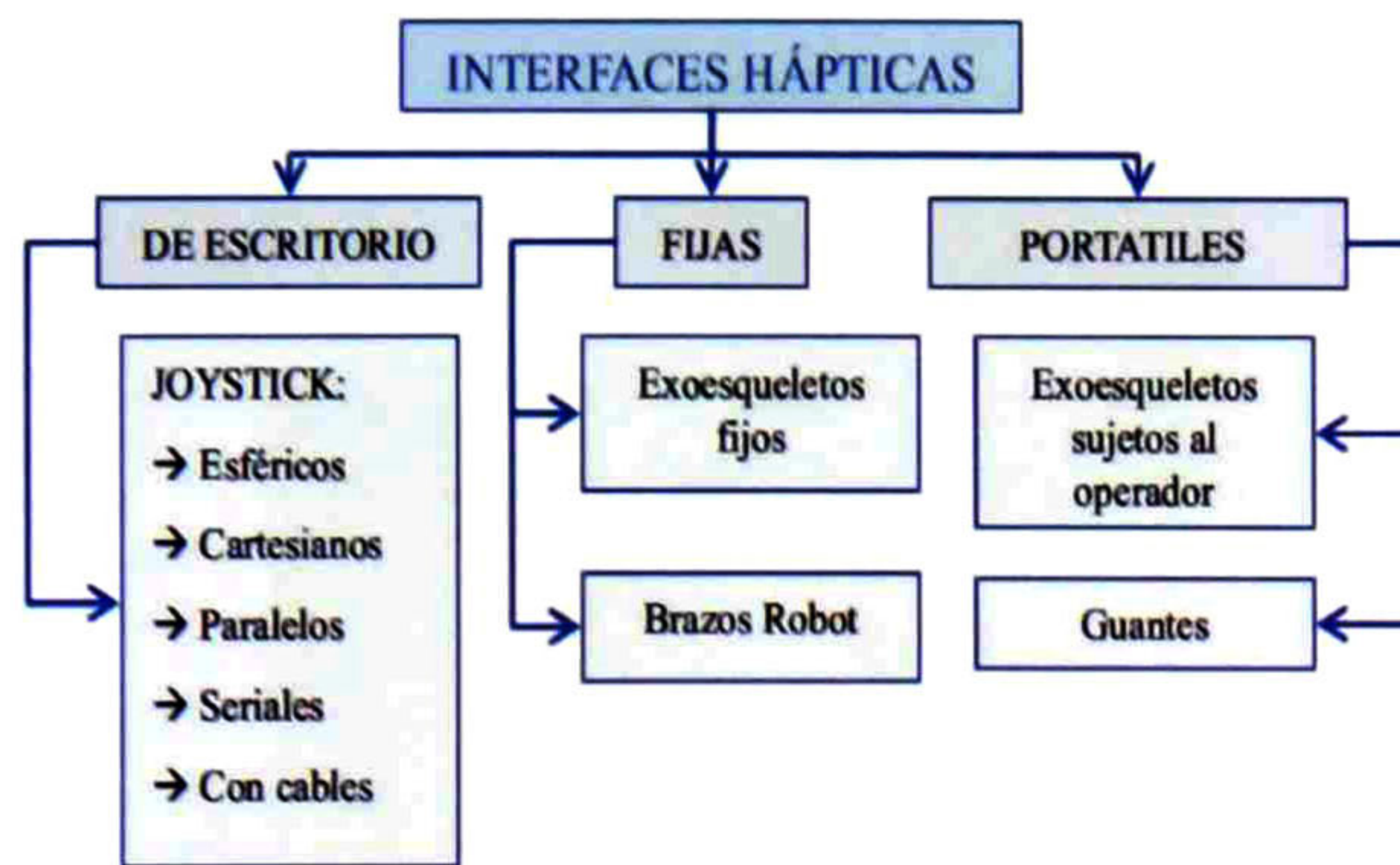


Figura 5. Clasificación de interfaces hápticas

La tecnología háptica se usa comúnmente para la creación de ambientes de realidad virtual, ya que permite que el usuario actúe en ambientes virtuales simulados. En el caso de sistemas médicos, estas interfaces se utilizan con 2 fines distintos, para los sistemas de entrenamiento quirúrgico, y para los sistemas de cirugía de mínima invasión. En el caso de los sistemas de entrenamiento, los médicos que están recibiendo alguna especialización en cirugía son enfrentados a escenarios virtuales en una computadora, la cual recrea las sensaciones táctiles que percibiría el médico en una cirugía real, y el cirujano en entrenamiento logra tales percepciones a través de la interfaz háptica. En el caso del sistema de mínima invasión, se usa interfaz similar a la que se muestra en la Figura 6. Esta interfaz cuenta con 6 grados de libertad, y puede ser usada como un joystick para un brazo robot, el cual deberá contar con un sensor de fuerza para poder transmitir al cirujano las sensaciones táctiles que el robot percibe.



Figura 6. Interfaz háptica tipo joystick de 6 grados de libertad de SENSABLE Inc (1).

### 1.3 Cámaras RGB-D

Las cámaras de color, normalmente conocidas como RGB (Red Green Blue) comunes son capaces de capturar imágenes bidimensionales del espacio. En 2009 se la empresa israelí PrimeSense lanzó una nueva tecnología capaz de asociar a cada pixel de la imagen una profundidad espacial, con lo cual se puede tener una nube de puntos del espacio tridimensional. Esta tecnología fue usada por Microsoft para crear Kinect, el primer sensor RGB-D del mundo usado para videojuegos. En 2013 la empresa PrimeSense fue adquirida por Apple inc. Actualmente esta tecnología se está usando en áreas como televisión, dispositivos móviles, computadoras de escritorio, displays interactivos, robótica, cuidado de la salud, entre otros. En la Figura 7 se muestra una foto del PrimeSense RD 1.09, sensor RGB-D usado en este trabajo.





Figura 7. Sensor RGB-D PrimeSense RD 1.09

## 1.4 Objetivos del trabajo

El principal objetivo de este trabajo es el de plantear y probar un modelo basado en redes neuronales que permita a un sistema cibernético mezclar percepción táctil junto con percepción visual, de tal manera que dicho sistema pueda representar objetos tridimensionales, tanto en sus características visuales (geometría espacial) como en sus características mecánicas superficiales (rigidez). Esto nos motivó a utilizar **álgebra geométrica conformal**, el cual es un marco teórico que nos permite modelar problemas geométricos de manera simple, compacta e intuitiva para sistemas de tiempo real.

Desde un punto de vista técnico, este trabajo se plantea el objetivo de iniciar la construcción de un sistema de cirugía de mínima invasión para asistir a cirujanos en operaciones de abdomen, específicamente para operaciones de riñón tales como trasplante, obstrucción de vías urinarias, entre otras, con la condición que sea de bajo costo. Aquí se plantea la configuración de un sistema que requiere la selección de los brazos robots más apropiados, su colocación espacial y la selección de las interfaces hápticas adecuadas.

## 1.5 Estructura del Documento

El documento está organizado de la siguiente manera: el *capítulo 2* define el marco matemático en el cual se realiza todo este trabajo, en el *capítulo 3* presentamos el marco teórico sobre redes neuronales y se detalla la propuesta de una nueva red neuronal geométrica la cual **llamamos red neuronal esférica para regresión**, se detalla también el **modelo neuronal tacto-visual** que consideramos como el principal aporte de este trabajo; en el *capítulo 4* se plantea un esquema inicial de lo que será el sistema de cirugía de mínima invasión, además se describe su uso para probar los modelos teóricos propuestos en esta tesis, en el *capítulo 5* se muestran los resultados experimentales del modelo neuronal tacto-visual, además se muestran simulaciones del uso de la red neuronal esférica para regresión y se realiza un comparativo con las redes tradicionales y, finalmente, en el *capítulo 6* se expresan las conclusiones y el posible seguimiento que se le puede dar a este trabajo doctoral.

## 2. Antecedentes matemáticos

En este capítulo se presentarán los fundamentos matemáticos de tópicos requeridos en el desarrollo del trabajo aquí presentado.

### 2.1 Álgebra Geométrica

En 1844, el matemático alemán, Hermann Gunther Grassmann [1809-1877], publicó en su “Teoría de la expansión lineal, una nueva rama de las matemáticas” (del alemán *Die Lineale Ausdehnungslehre: ein neuer Zweig der Mathematik*), la primera edición de su cálculo geométrico que ofrecía una manera original de describir operaciones geométricas. En este trabajo se presenta una extensión del álgebra euclidiana, apareciendo el *producto exterior* como el principal producto geométrico de esta teoría. Con esta operación, el producto de vectores generaba áreas, volúmenes y objetos de altas dimensiones.

El álgebra geométrica, también conocida como *álgebra de Clifford* (William K. Clifford [1845-1879]), es una herramienta que mediante una serie de operaciones y propiedades, extiende el sistema de números reales y complejos hacia el cálculo vectorial incluyendo el productora exterior [13]. En la década de 1960, esta álgebra es reformulada por David Hestenes planteando las álgebras geométricas; las cuales

son una herramienta para unificar los conceptos de geometría con el álgebra y la representación geométrica, y facilitar su uso en física e ingeniería [14].

Una de las principales características del álgebra de Clifford (o álgebra geométrica), es que nos permite representar entidades de orden mayor con una simbología compacta, y poder operarlas de forma lineal. Las líneas, planos o esferas son ejemplos de entidades de orden mayor y son representadas como únicos elementos del álgebra de Clifford.

## El Producto Clifford de Vectores en $\mathbb{R}^2$

Sería útil tener una multiplicación de vectores que satisfagan los mismos axiomas de multiplicación de los números reales (distributividad, asociatividad y conmutatividad) y que la norma de estos vectores se preserve en dicha multiplicación  $|ab| = |a||b|$ , donde  $a$  y  $b$  son vectores. A esta multiplicación anticonmutativa de vectores la llamamos *producto Clifford*.

Tomando dos vectores con base ortonormal  $e_1 = [1,0]^T$  y  $e_2 = [0,1]^T$  en el espacio vectorial  $\mathbb{R}^2$ . La magnitud del vector  $r = xe_1 + ye_2$  está dada por  $|r| = \sqrt{x^2 + y^2}$ . Si el vector  $r$  es elevado al cuadrado utilizando el producto Clifford, la expresión sería  $\pi r = r^2 = |r|^2$ .

Usando la forma de coordenadas, introduciremos el producto de vectores de tal forma que

$$(xe_1 + ye_2)^2 = x^2 + y^2$$

Utilizando la regla distributiva sin asumir la conmutabilidad obtenemos

$$x^2e_1^2 + y^2e_2^2 + xy(e_1e_2 + e_2e_1) = x^2 + y^2$$

con esto podemos definir el producto de vectores (el producto Clifford).

**Definición 2.1.** Sean  $a = a_1e_1 + a_2e_2, b = b_1e_1 + b_2e_2 \in \mathbb{R}^2$  dos vectores con una base ortonormal  $e_1, e_2$ . la multiplicación utilizando el producto Clifford de estos vectores queda de la siguiente forma:

$$ab = a_1b_1 + a_2b_2 + (a_1b_2 - a_2b_1)e_1e_2$$

Esto se satisface si los vectores de la base obedecen las reglas de multiplicación

$$e_1^2 = e_2^2 = 1, \quad e_1e_2 = -e_2e_1 \text{ o equivalentemente } |e_1| = |e_2| = 1, \quad e_1 \perp e_2.$$

Usamos la propiedad asociativa para calcular

$$e_1 e_2^2 = e_1 e_2 e_1 e_2 = e_1 e_2 e_1 e_2 = -e_1^2 e_2^2 = -1.$$

Como el cuadrado del elemento  $e_1 e_2$  es negativo, se sigue que no es un escalar ni un vector, sino una nueva clase de unidad llamado *bivector*, representando el área del plano orientado (ver Figura 8) del cuadrado con los lados  $e_1$  y  $e_2$ . Escribiremos  $e_{12} = e_1 e_2$ .

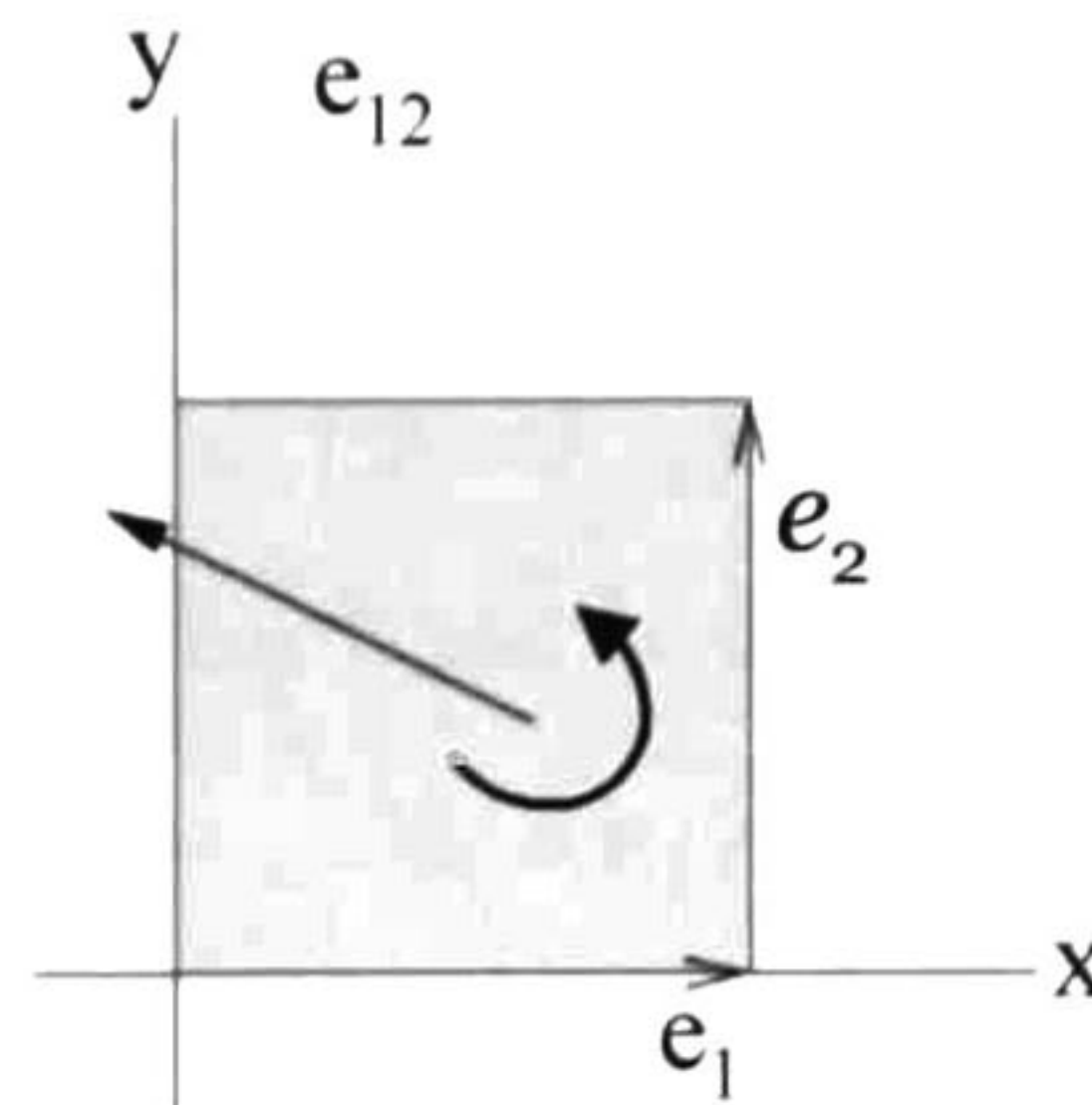


Figura 8. Bivector representado como un plano orientado.

## Producto Exterior

Analizando el producto de los vectores  $a = a_1 e_1 + a_2 e_2$  y  $b = b_1 e_1 + b_2 e_2$ , podemos identificar dos partes

$$a \cdot b = a_1 b_1 + a_2 b_2 \Rightarrow \text{El producto escalar 'a punto b'}, \text{ y} \quad (2.1.1)$$

$$a \wedge b = (a_1 b_2 - a_2 b_1) e_{12} \Rightarrow \text{El producto exterior 'a wedge b'}. \quad (2.1.2)$$

El bivector  $a \wedge b$  representa el segmento del plano orientado del paralelogramo con lados  $a$  y  $b$ . El área de este paralelogramo es  $|a_1 b_2 - a_2 b_1|$  y también representa la magnitud del bivector.



Figura 9. Sentido del bivector

El bivector  $a \wedge b$  y  $b \wedge a$  tienen la misma magnitud pero con sentidos de rotación opuestos (ver Figura 9). Esto puede ser expresado simplemente escribiendo

$$\mathbf{a} \wedge \mathbf{b} = -\mathbf{b} \wedge \mathbf{a} \quad (2.1.3)$$

Notamos que el producto Clifford de los dos vectores es la suma de su parte escalar  $\mathbf{a}_1\mathbf{b}_1 + \mathbf{a}_2\mathbf{b}_2$  y bivectorial  $(\mathbf{a}_1\mathbf{b}_2 - \mathbf{a}_2\mathbf{b}_1)$ . El producto Clifford, así como el producto exterior y producto interior, puede ser extendido a mayores dimensiones.

### Álgebra Geométrica *n-dimensional*

Un espacio vectorial *n-dimensional* puede ser expandido usando las bases ortonormales de sus vectores  $\mathbf{e}_i \quad i=1, \dots, n$  tal que  $\mathbf{e}_i\mathbf{e}_j = \mathbf{e}_{ij}$ . Esto nos lleva a una base que expande un espacio vectorial lineal que corresponde al álgebra geométrica  $G_n$ . Los elementos generados por dicha base son

$$1, \mathbf{e}_i, \mathbf{e}_i \wedge \mathbf{e}_j, \mathbf{e}_i \wedge \mathbf{e}_j \wedge \mathbf{e}_k, \dots, I \quad (2.1.4)$$

donde  $I = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \dots \wedge \mathbf{e}_n$ .

En general un álgebra geométrica  $G_{p,q,r} \quad p,q,r \in \mathbb{N}$  es un espacio lineal de dimensión  $2^n$  con  $n = p + q + r$  con sub-espacios a los que llamaremos **blades** (tomado del idioma inglés), y donde  $p, q$  y  $r$  corresponden al número de vectores base que cuadran a  $+1, -1$  y  $0$ , respectivamente. La notación  $G_{p,q,r}$  es usada para determinar la métrica del espacio en cuestión.

Para aclarar lo anterior, sean  $\mathbf{e}_i, \mathbf{e}_j \in G_{p,q,r}$  dos vectores de la base ortonormal de un espacio vectorial. Entonces, el producto geométrico de los vectores de la base nos da la base del álgebra geométrica, la cual se define como

$$\mathbf{e}_i\mathbf{e}_j = \begin{cases} 1 & \text{para } i = j \in 1, \dots, p \\ -1 & \text{para } i = j \in p+1, \dots, q \\ 0 & \text{para } i = j \in q+1, \dots, r \\ \mathbf{e}_{ij} = -\mathbf{e}_{ji} & \text{para } i \neq j \end{cases} \quad (2.1.5)$$

El producto geométrico (o Clifford) de dos vectores se puede representar de la siguiente forma

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b}. \quad (2.1.6)$$

La regla conmutativa  $\mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a}$ , junto con la regla anti-conmutativa  $\mathbf{a} \wedge \mathbf{b} = -\mathbf{b} \wedge \mathbf{a}$ , nos permite inferir

$$\mathbf{ba} = \mathbf{a} \cdot \mathbf{b} - \mathbf{a} \wedge \mathbf{b} \quad (2.1.7)$$

Sumando y restando las ecuaciones (2.1.6) y (2.1.7), podemos definir

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} &= \frac{1}{2} \mathbf{ab} + \mathbf{ba} \\ \mathbf{a} \wedge \mathbf{b} &= \frac{1}{2} \mathbf{ab} - \mathbf{ba} \end{aligned} \quad (2.1.8)$$

El producto exterior puede ser extendido a dimensiones mayores, por ejemplo el producto  $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$  es un *trivector*, y en general el *producto exterior* de  $k$  vectores es un  $k$ -vector o *k-blade*; dicha cantidad decimos que tiene grado  $k$ . El producto interior de 1-blade con un  $k$ -blade se puede generalizar por medio de

$$\mathbf{a} \cdot \mathbf{B}_k = \frac{1}{2} \mathbf{aB}_k + (-1)^{k+1} \mathbf{B}_k \mathbf{a} \quad (2.1.9)$$

Un *multivector* es una combinación lineal de vectores (o blades) de distintos grados, y se dice que es un multivector homogéneo si todos los blades que lo componen son del mismo grado.

Cada multivector  $M \in G_n$  puede ser escrito en su forma expandida como

$$M = \sum_{k=0}^n \langle M \rangle_k \quad (2.1.10)$$

donde  $\langle M \rangle_k$  es la parte de  $M$  que tiene grado  $k$ . Note que  $\langle M \rangle_0 \in \mathbb{R}$ .

### Propiedades de Multivectores

Un elemento  $M \in G_n$  es invertible si existe otro elemento  $N \in G_n$  tal que  $MN = NM = 1$ , donde  $N$  (si existe) es único, es llamado el inverso de  $M$  y se denota por  $M^{-1}$ . Una condición para la existencia del inverso de  $M$  es que este sea no nulo i.e.  $M^2 \neq 0$ .

Por asociatividad y multi-linealidad, el producto exterior se extiende a cualquier número finito de multivectores y a escalares, con la siguiente propiedad

$$\lambda \wedge M = M \wedge \lambda = \lambda M, \text{ para } \lambda \in \mathbb{R}, M \in G_n \quad (2.1.11)$$

y para el producto interior tenemos

$$\lambda \cdot M = M \cdot \lambda = 0, \text{ para } \lambda \in \mathbb{R}, M \in G_n. \quad (2.1.12)$$

Para cualquier blade  $A$  y  $B$  con grados  $r$  y  $s$  diferente de cero, tenemos

$$A \cdot B = \langle AB \rangle_{|r-s|} \quad (2.1.13)$$

$$A \wedge B = \langle AB \rangle_{r+s} \quad (2.1.14)$$

El *blade* de más alto grado es el *n-blade*  $I = e_1 \wedge \dots \wedge e_n$  y es llamado *pseudoescalar*. Un álgebra geométrica se llama *no degenerada* si la magnitud del pseudoescalar es diferente de cero. Los pseudoescalares son indexados de acuerdo al álgebra a la que pertenecen, por ejemplo  $I_E \in G_3$  (álgebra euclidiana) y  $I_C \in G_{4,1}$  (álgebra conformal).

El dual de un multivector  $M \in G_n$  es denotado por

$$M^* = MI_n^{-1} \quad (2.1.15)$$

y por lo tanto el dual de un *r-blade* es un *(n-r)-blade*.

La reversión de un *s-blade*  $A = a_1 \wedge \dots \wedge a_s$  se denota como

$$A = a_1 \wedge \dots \wedge a_s \quad \tilde{A} = a_s \wedge a_{s-1} \wedge \dots \wedge a_2 \wedge a_1 \quad (2.1.16)$$

y generalizando para cualquier multivector se obtiene

$$\langle M \rangle_i = -1^{i(i-1)/2} \langle M \rangle_i, \text{ para } M \in G_n, 0 \leq i \leq n. \quad (2.1.17)$$

Dado  $D = ABC$ , donde  $A, B, C, D \in G_n$ , su reversión está dada por

$$D = ABC = CBA. \quad (2.1.18)$$

La involución principal de  $G_n$  (también llamada *involución de grado*) es definida como

$$\langle M \rangle_i = -1^i \langle M \rangle_i. \quad (2.1.19)$$

La función exponencial de un multivector  $M$  puede ser expresada en series de expansión como

$$e^M = \sum_{k=0}^{\infty} \frac{M^k}{k!} \quad (2.1.20)$$

## Álgebra Geométrica Conformal

El contar con un álgebra que permite modelar entidades geométricas (puntos, líneas, planos, esferas, círculos, etc.) facilita el operar dichas entidades y sus relaciones geométricas de manera simple y compacta. El *álgebra geométrica conformal* posee dichas características y se relaciona con el espacio Euclidiano mediante una proyección estereográfica. Utilizaremos el álgebra geométrica  $G_{4,1}$  para trabajar con la geometría conformal y mostraremos como el espacio  $\mathbb{R}^3$  es representado en un espacio  $\mathbb{R}^{4,1}$  donde los superíndices denotan una métrica pseudo-euclidiana. Este espacio está definido por las bases:  $e_i$  (vectores),  $e_{ij}$



(bivectores) con  $i, j = 1, \dots, 5$  ; de las cuales  $e_{23}, e_{31}$  y  $e_{12}$  corresponden a las bases de los *cuaterniones de Hamilton*.

### Mapeo Conformal

Este resulta de hacer una proyección estereográfica y una homogenización de las coordenadas. Tomemos el ejemplo de mapear el espacio euclidiano  $\mathbb{R}^1$  con base  $e_1$  a  $G_{2,1}$  con bases  $p = e_1, e_+$  y  $q = e_-$ .

Sea  $x_e \in \mathbb{R}^1$ , entonces su representación conformal correspondiente  $x \in G_{2,1}$  (ver la Figura 10) se obtiene como la intersección de la línea  $f x = -\frac{x}{x_e} + 1$  con el círculo unitario  $f x^2 + x^2 = 1$ , de donde obtenemos

$$x^2 - 2xx_e + x^2x_e^2 = 0. \tag{2.1.21}$$

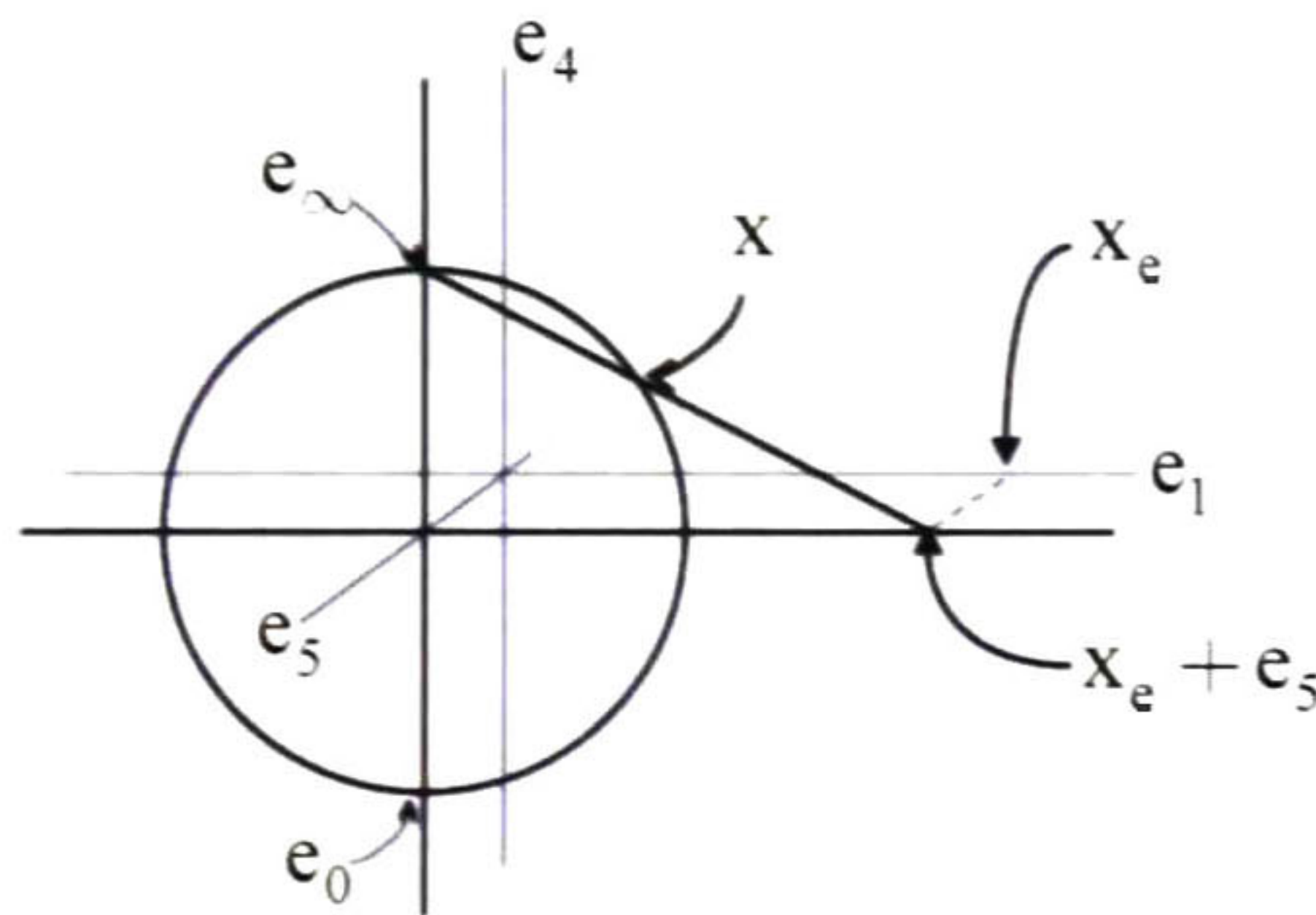


Figura 10. Proyección estereográfica de un punto en un círculo unitario

Esta ecuación tiene 2 soluciones

$$x = 0, \quad x = 2 \frac{x_e}{x_e^2 + 1}, \tag{2.1.22}$$

si evaluamos en la solución no trivial obtenemos el punto de intersección como

$$P = \left( \frac{2x_e}{x_e^2 + 1}, \frac{x_e^2 - 1}{x_e^2 + 1} \right), \tag{2.1.23}$$

y agregando la coordenada homogénea ( $e_-$ ) tenemos

$$x = \frac{2x_e}{x_e^2 + 1} e_1 + \frac{x_e^2 - 1}{x_e^2 + 1} e_+ + e_- \tag{2.1.24}$$

para obtener, finalmente, la *representación conformal* de  $x_e$ . De (2.1.24) se pueden obtener las representaciones conformales de los *puntos en el infinito* y el *origen*.

Para el punto en el infinito, denotado por  $e_\infty$ , se aplica el límite  $x_e \rightarrow \infty$  en (2.1.24) de donde obtenemos

$$\begin{aligned} e_\infty &= \lim_{x_e \rightarrow \infty} x \\ &= \lim_{x_e \rightarrow \infty} \left( \frac{2x_e}{x_e^2 + 1} e_1 + \frac{x_e^2 - 1}{x_e^2 + 1} e_+ + e_- \right), \\ &= \lim_{x_e \rightarrow \infty} \left( \frac{1}{x_e} e_1 + e_+ + e_- \right), \\ &= e_+ + e_-. \end{aligned} \tag{2.1.25}$$

Para el origen, denotado por  $e_0$ , aplicamos el límite  $x_e \rightarrow 0$  para obtener

$$\begin{aligned} e_\infty &= \lim_{x_e \rightarrow 0} x \\ &= \lim_{x_e \rightarrow 0} \left( \frac{2x_e}{x_e^2 + 1} e_1 + \frac{x_e^2 - 1}{x_e^2 + 1} e_+ + e_- \right), \\ &= -e_+ + e_- \end{aligned} \tag{2.1.26}$$

Los productos del punto en el infinito y el origen se definen como  $e_\infty \cdot e_0 = -2$ ,  $e_\infty \wedge e_0 = 2e_+ \wedge e_-$ . Con fines de simplificación algebraica, estos productos se normalizan a

$$\begin{aligned} e_\infty \cdot e_0 &= -1 \\ e_\infty \wedge e_0 &= e_+ \wedge e_- \end{aligned} \tag{2.1.27}$$

donde  $e_0 = -e_+ + e_- / 2$ . Note que se redefinió a  $e_0$ , pero esto es válido ya que es un vector homogéneo, y por lo tanto  $e_0 = ke_0$ ,  $k \in \mathbb{R}$ . Si definimos los vectores nulos como

$$e_- = \frac{1}{2}e_\infty + e_0, \quad e_+ = \frac{1}{2}e_\infty - e_0, \tag{2.1.28}$$

se puede representar a la transformación conformal usando estos vectores nulos con la forma

$$\begin{aligned} x &= \frac{2x_e}{x_e^2 + 1} + \frac{x_e^2 - 1}{x_e^2 + 1} \left( \frac{1}{2}e_\infty - e_0 \right) + \left( \frac{1}{2}e_\infty + e_0 \right) \\ x &= \frac{2x_e}{x_e^2 + 1} + \frac{x_e^2 e_\infty}{x_e^2 + 1} + \frac{2e_0}{x_e^2 + 1} \end{aligned} \tag{2.1.29}$$

para finalmente obtener

$$x = x_e + \frac{1}{2}x_e^2 e_\infty + e_0. \tag{2.1.30}$$

## Espacio de Minkowski $\mathbb{R}^{1,1}$

Este espacio es generado por la *base ortonormal*  $e_+, e_-$  o la *base nula*  $e_\sim, e_0$ . El pseudoescalar unitario se define como

$$E = e_0 \wedge e_\sim = e_+ \wedge e_- \quad (2.1.31)$$

y tiene las siguientes propiedades

$$\begin{aligned} E^2 = 1, \quad \tilde{E} = -E, \quad E e_\sim = -e_\sim, \quad e_0 E = e_0, \quad e_+ E = e_-, \\ e_- E = e_+, \quad e_+ e_\sim = 1 + E, \quad e_- e_\sim = -1 - E, \quad e_\sim \wedge e_- = E, \quad e_+ \cdot e_\sim = 1. \end{aligned} \quad (2.1.32)$$

La base vectorial y líneas nulas de  $G_{1,1}$  son mostradas en la Figura 11, donde el área sombreada representa el pseudo-escalar unitario  $E$ .

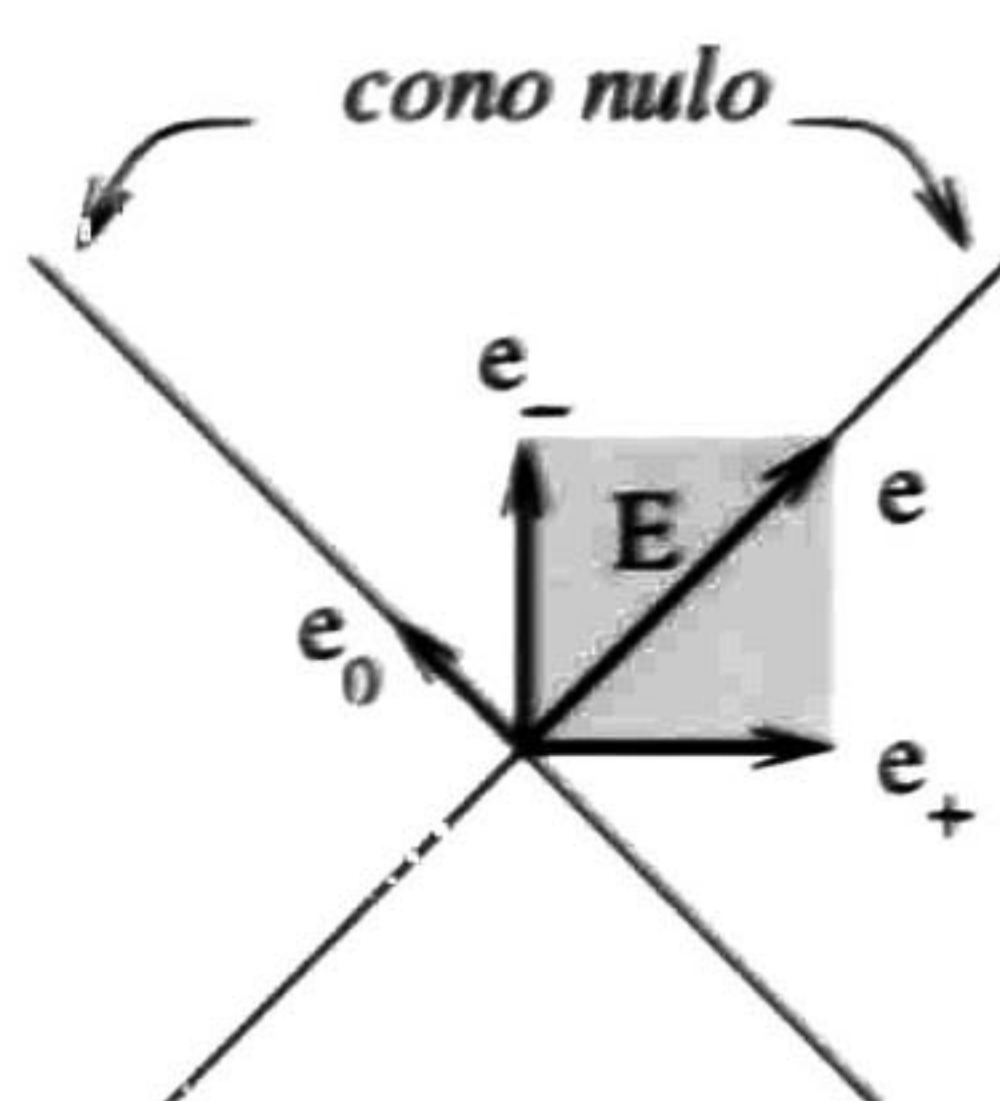


Figura 11. Vectores base y líneas nulas del plano de Minkowski

## Definición del Álgebra Geométrica Conformal

El rol que juega el plano de Minkowski de generar vectores nulos y expandir el espacio vectorial Euclidiano  $\mathbb{R}^n$  a  $\mathbb{R}^{n+1,1}$ , resulta en el álgebra geométrica conformal  $G_{n+1,1}$ . El conjunto  $N^{n+1}$  de todos los vectores nulos en  $\mathbb{R}^n$  es llamado *cono nulo*, y el conjunto de todos los puntos generalizados homogéneos de grado superior es llamado *horósfera*. En la Figura 12 se muestra la visualización del modelo homogéneo para la proyección estereográfica para el caso de 1D. Todos los puntos estereográficos proyectados yacen en el cono, que es el cono nulo relacionado con el plano de Minkowski. También se notan el *hiperplano*  $P e_\sim, e_0$  y la horósfera del álgebra conformal.

Un vector en este espacio puede ser representado como una descomposición ortogonal

$$a = P_E a + P_E^\perp a \quad (2.1.33)$$

donde los operadores de proyección  $P_E$  y  $P_E^\perp$  están dados por

$$\begin{aligned}
P_E a &= a \cdot E \quad E = \alpha e_0 + \beta e_\infty \in \mathbb{R}^{1,1} \\
P_E^\perp a &= a \cdot E^* \quad \tilde{E}^* = a \wedge E \quad E = a_e \in \mathbb{R}^n
\end{aligned}
\tag{2.1.34}$$

El espacio vectorial conformal derivado de  $\mathbb{R}^3$  es denotado por  $\mathbb{R}^{4,1}$ . Su base vectorial está dada por  $e_1, e_2, e_3, e_+, e_-$  ó por  $e_1, e_2, e_3, e_0, e_\infty$ . El álgebra correspondiente  $G_{4,1}$  contiene  $2^5 = 32$  elementos. El pseudo-escalar unitario se denota por  $I_c = e_{123+-} = e_{1230\infty} = I_E E$ , donde  $I_E$  es el pseudo-escalar de  $G_3$ .

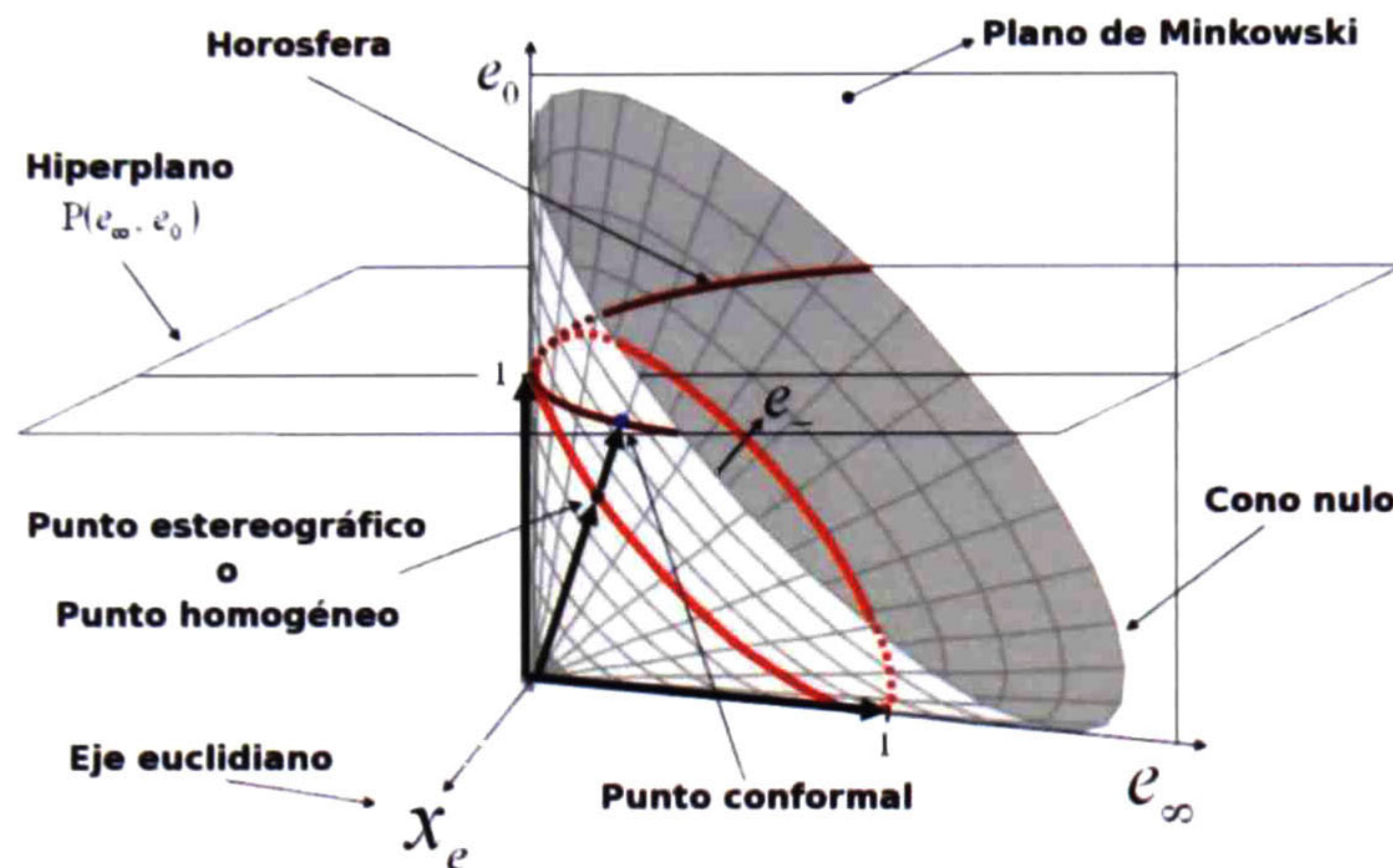


Figura 12. El modelo homogéneo para la proyección estereográfica en el caso 1D.

## Entidades Geométricas Conformales

Una de las grandes ventajas del *álgebra geométrica conformal* es la de representar de manera simple y compacta entidades geométricas, así como ofrecer las herramientas algebraicas necesarias para obtener relaciones geométricas entre ellas. En las siguientes secciones se describirán las principales entidades geométricas conformales y como se generan.

### Esfera

La esfera es la entidad geométrica de cálculo básica para esta álgebra, ya que por medio de ella se pueden generar todas las demás entidades.

Existen muchas formas de construir una esfera: definiendo su centro y radio, con cuatro puntos que yacen en ella, definiendo su círculo ecuatorial, sus dos polos, entre otras. Se obtendrá la *ecuación conformal* de la esfera a partir de la *ecuación*

euclidiana de la misma con centro en  $p_e$  y radio  $\rho$  como se muestra en la Figura 13

$$x_e - p_e^2 = \rho^2 \quad (2.1.35)$$

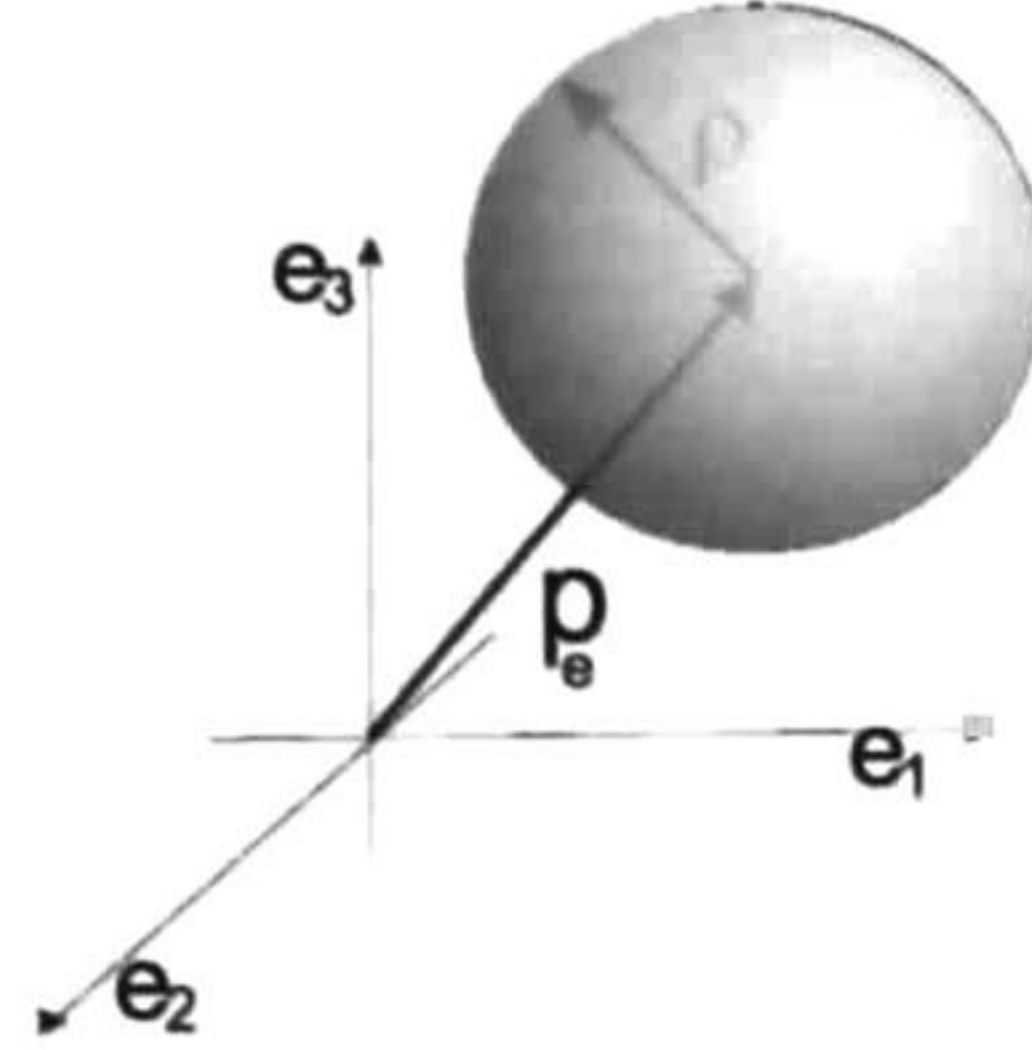


Figura 13. Esfera construida con centro en  $p_e$  y radio  $\rho$

Desarrollando obtenemos

$$\begin{aligned} x_e^2 - 2x_e \cdot p_e + p_e^2 &= \rho^2 \\ \frac{x_e^2}{2} - x_e \cdot p_e + \frac{p_e^2 - \rho^2}{2} &= 0 \quad (2.1.36) \\ \underbrace{\left( x_e + \frac{x_e^2}{2} e_\infty + e_0 \right)}_x \cdot \underbrace{\left( p_e + \frac{p_e^2 - \rho^2}{2} e_\infty + e_0 \right)}_s &= 0 \end{aligned}$$

de donde se obtiene la ecuación conformal de la esfera como

$$s = p_e + \frac{p_e^2 - \rho^2}{2} e_\infty + e_0. \quad (2.1.37)$$

Se puede observar que es un *1-vector* o vector de grado 1, mientras que la relación  $x \cdot s = 0$  determina cuando un punto yace sobre la esfera. Simplificando (2.1.37) obtenemos

$$s = p - \frac{1}{2} \rho^2 e_\infty \quad (2.1.38)$$

y las siguientes relaciones

$$\rho^2 = s \cdot s, \quad p = s + \frac{1}{2} s \cdot s e_\infty \quad (2.1.39)$$

También es posible definir la esfera con cuatro puntos que yacen en ella; esto se logra mediante el producto exterior o *wedge* de dichos puntos para definir la representación dual de la esfera como

$$s^* = x_1 \wedge x_2 \wedge x_3 \wedge x_4. \quad (2.1.40)$$

Cualquier otro punto  $x_i$  que yace sobre la esfera se puede definir en términos de los cuatro que la definieron, por lo que se obtiene la relación

$$s^* \wedge x_i = 0 \quad (2.1.41)$$

la cual nos ofrece otra forma de corroborar si un punto  $x_i$  yace sobre una esfera  $s$ .

## Plano

Un plano se puede imaginar como una esfera de radio infinito (ver Figura 14), por lo que, para obtener la ecuación del plano, solo es necesario intercambiar uno de los puntos de la esfera por  $e_\infty$ .

$$\pi^* = x_1 \wedge x_2 \wedge x_3 \wedge e_\infty \quad (2.1.42)$$

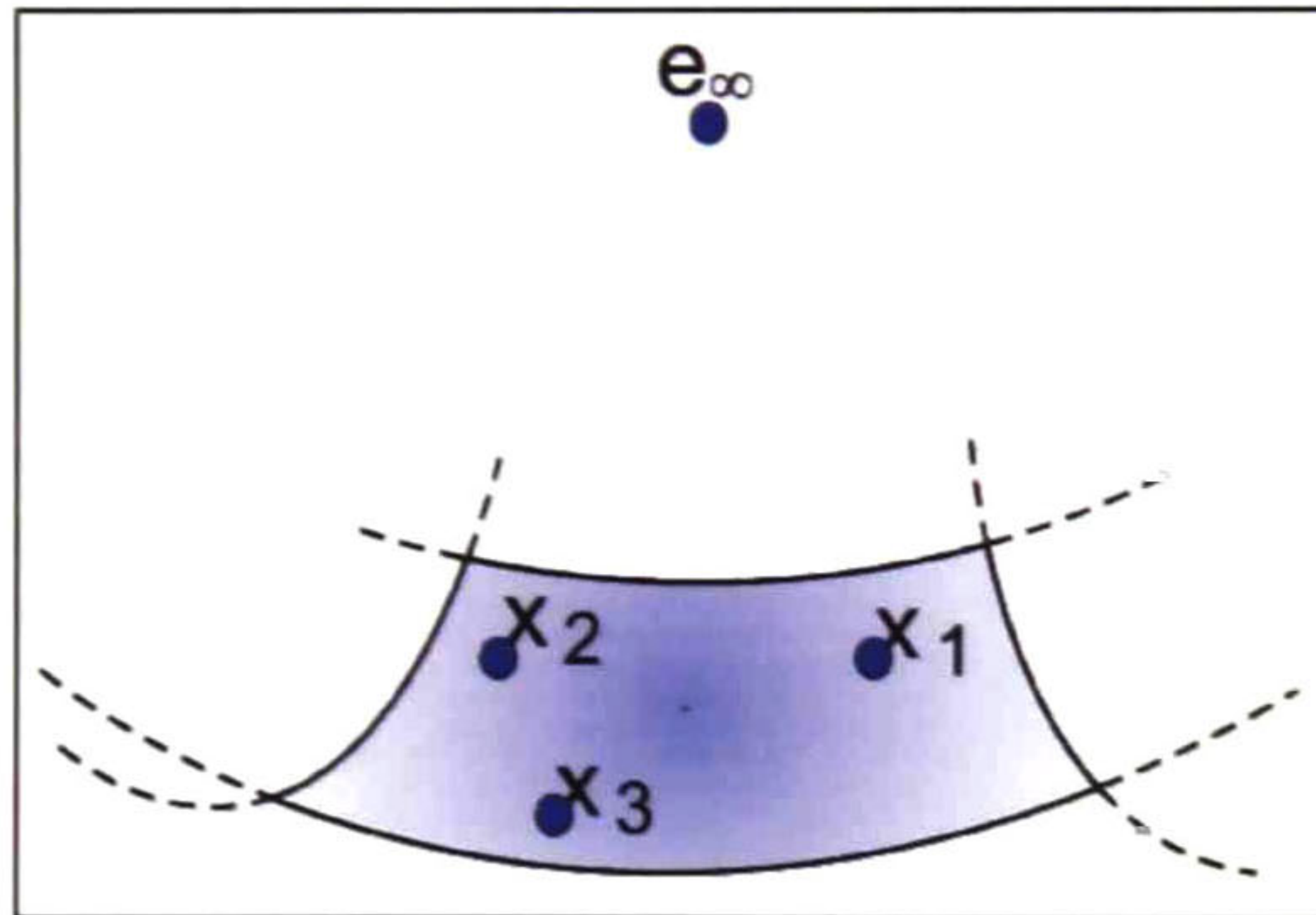


Figura 14. Plano definido por 3 puntos que yacen en él y el punto en el infinito.

Definiendo  $x_{1e} \wedge x_{2e} = -x_{1e} \wedge x_{1e} - x_{2e}$  se puede re-escribir la ecuación anterior como

$$x_2 \wedge x_1 \wedge x_2 \wedge e_\infty = x_{3e} \wedge x_{1e} \wedge x_{2e} \wedge e_\infty + x_{2e} - x_{1e} \wedge x_{3e} - x_{1e} \quad (2.1.43)$$

de donde podemos obtener

$$\pi^* = d \mathbf{I}_E e_\infty + N E \quad (2.1.44)$$

donde  $d$  es la *distancia de Hesse* (distancia más corta del origen  $e_0$  al plano  $\pi$ ) y se define como

$$d = \frac{x_{1e} \wedge x_{2e} \wedge x_{3e}}{|x_{2e} - x_{1e} \wedge x_{3e} - x_{1e}|}, \quad (2.1.45)$$

y  $N = x_{2e} - x_{1e} \wedge x_{3e} - x_{1e} / |x_{2e} - x_{1e} \wedge x_{3e} - x_{1e}|$  es el bivector unitario normal a  $\pi$

De (2.1.44) se puede obtener la representación estándar del plano como

$$\begin{aligned}\pi &= \pi^* l_c \\ &= de_\infty + n\end{aligned}\tag{2.1.46}$$

donde  $n = Nl_E = Ne_{123}$  es el vector de grado uno normal a  $\pi$ .

## Círculo

El círculo se puede obtener por tres puntos no colineales que yacen en él

$$z^* = x_1 \wedge x_2 \wedge x_3,\tag{2.1.47}$$

ó por la intersección de dos esferas

$$z = s_1 \wedge s_2\tag{2.1.48}$$

como se muestra en la Figura 15 Esto es realmente muy práctico, ya que el círculo es simplemente un trivector o vector de grado 3 en álgebra conformal, en comparación con las 3 ecuaciones paramétricas necesarias en álgebra euclidiana para representar un círculo en  $\mathbb{R}^3$ .

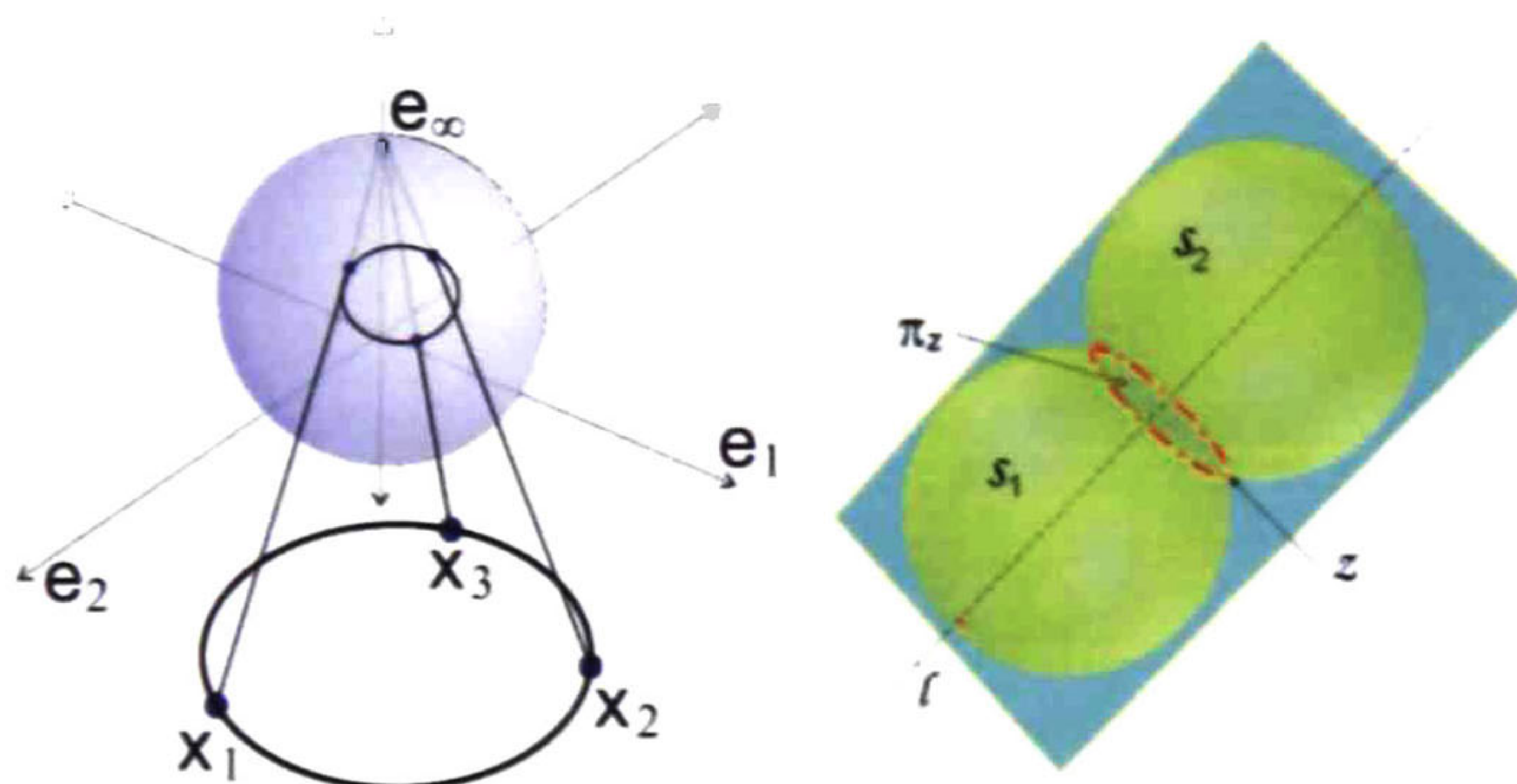


Figura 15. Círculo formado por tres puntos o la intersección de dos esferas.

Similar al caso de la esfera, cualquier punto  $x$  que yace sobre el círculo cumplirá la siguiente relación

$$x \cdot z = 0\tag{2.1.49}$$

Se puede calcular el plano donde un círculo yace de la siguiente forma

$$\pi_z^* = z^* \wedge e_\infty,\tag{2.1.50}$$

y la línea del eje del círculo como

$$l^* = z \wedge e_\infty,\tag{2.1.51}$$

por lo tanto, la intersección de los elementos anteriores nos permite definir el centro del círculo de la siguiente manera

$$c = l \cap \pi_z = l^* \cdot \pi_z. \quad (2.1.52)$$

## Línea

Así como el plano se puede imaginar como una esfera de radio infinito, la línea puede conceptualizarse como un círculo de radio infinito (véase Figura 16). De esta forma, y utilizando (2.1.42), obtenemos

$$\begin{aligned} L^* &= x_1 \wedge x_2 \wedge e_\infty \\ &= x_{1e} \wedge x_{2e} \wedge e_\infty + x_{2e} - x_{1e} E \\ &= e_\infty m + nE \end{aligned} \quad (2.1.53)$$

donde  $n$  es el vector de dirección de la línea y  $m$  su momento.

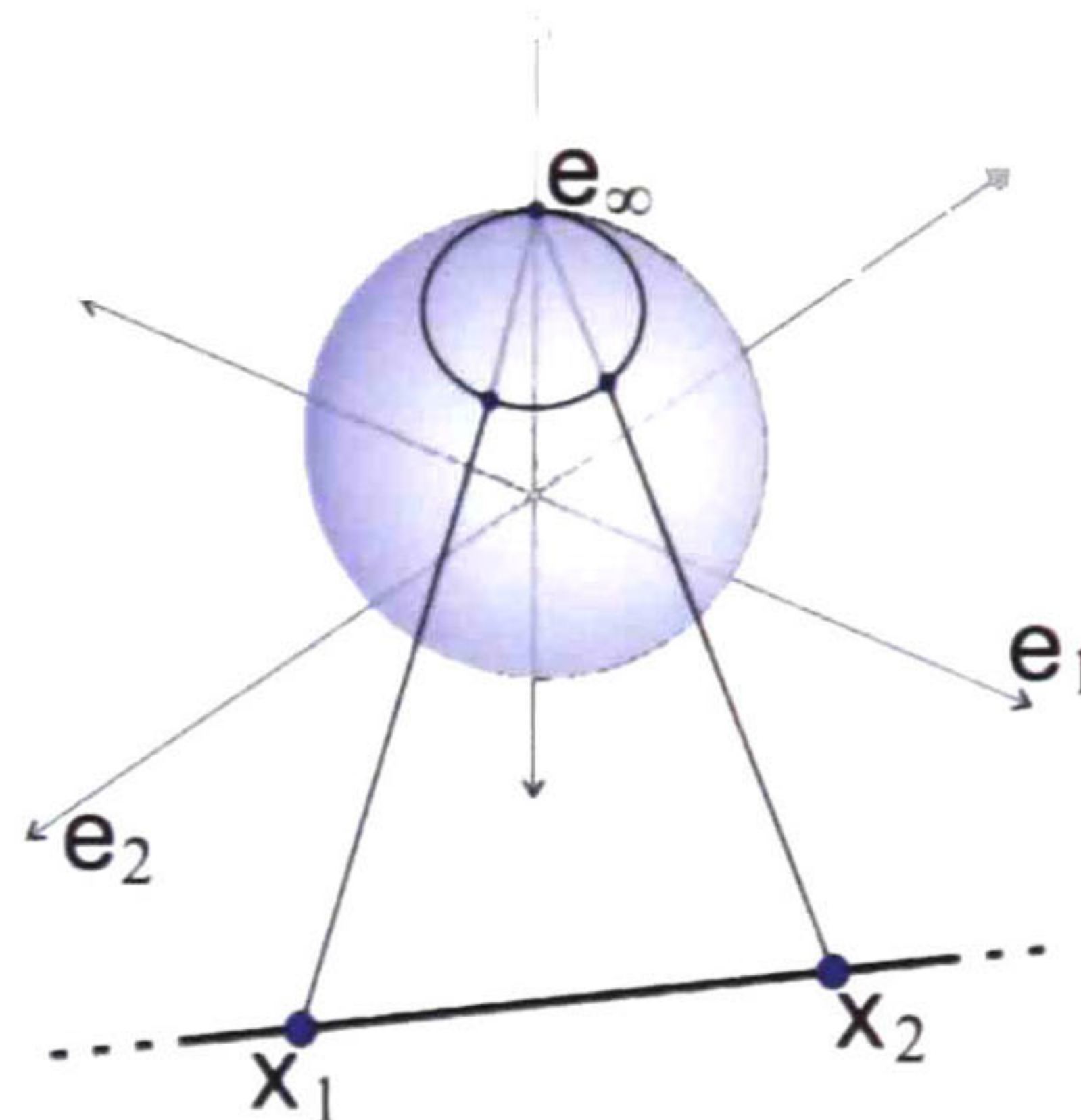


Figura 16. Línea mapeada en la esfera unitaria.

Esta entidad también puede ser generada usando un punto  $x_1$  que yace sobre ella y la dirección de la misma como

$$L^* = x_1 \wedge n \wedge e_\infty. \quad (2.1.54)$$



## Par de Puntos

Si tomamos la parte bivectorial del producto de dos puntos,  $a$  y  $b$ , obtenemos una entidad conocida como *par de puntos*. Dicha entidad posee información sobre la localización de los 2 puntos, y está definida como

$$PP^* = a \wedge b. \quad (2.1.55)$$

Otra forma de obtener el par de puntos es como la intersección de una línea con una esfera (ver Figura 17), lo que resulta en

$$PP^* = sL^* \quad (2.1.56)$$

Esta expresión alternativa ofrece la ventaja de poder extraer uno de los puntos que conforman el par de puntos utilizando el centro de la esfera  $c$  y trasladándolo ambos lados una distancia igual al radio de la esfera  $\rho$ .

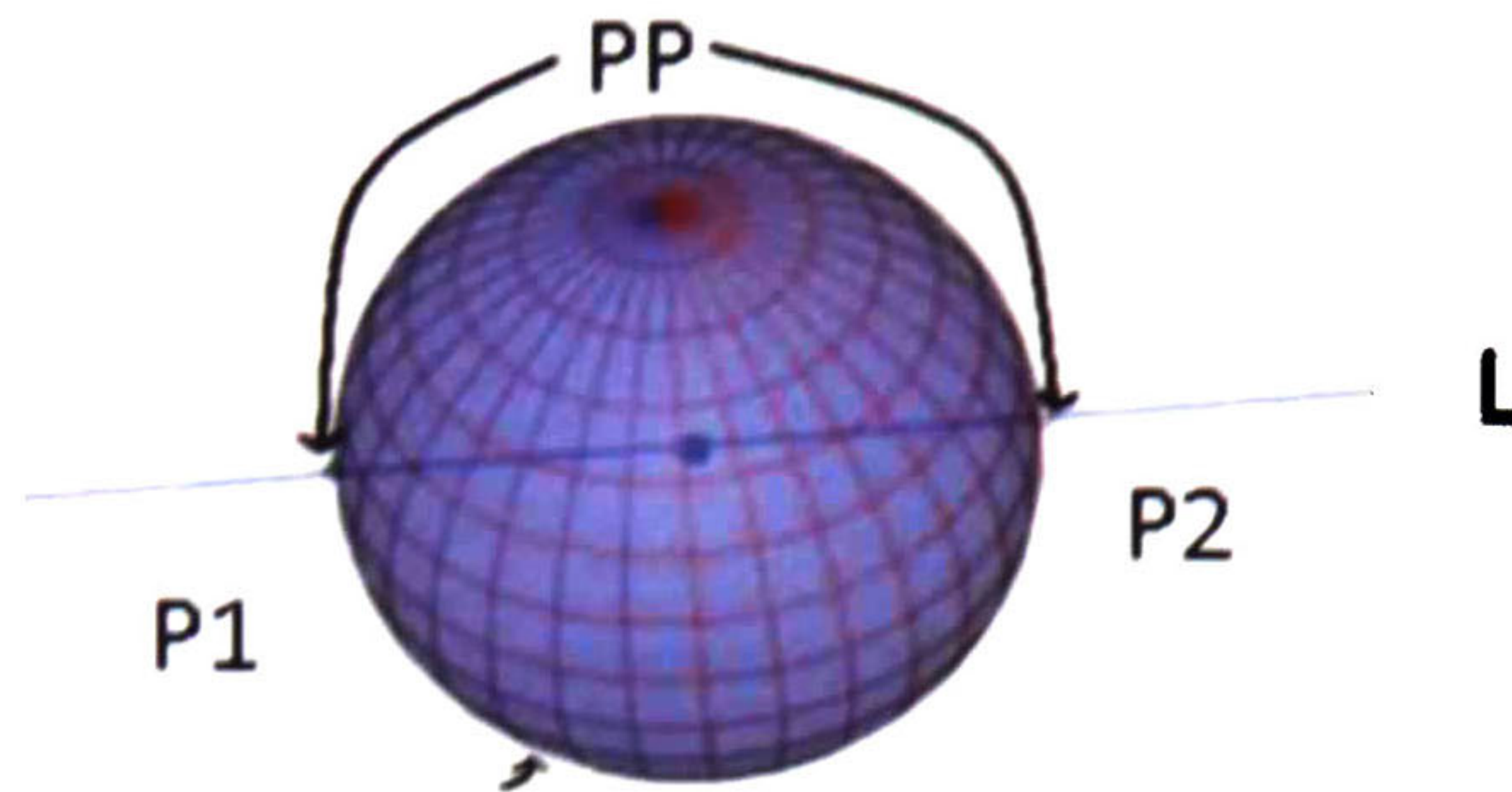


Figura 17. El par de puntos como la intersección de una línea y una esfera

De (2.1.56) obtenemos la esfera, el radio y su centro como

$$s = \frac{PP^*}{L^*} \Rightarrow \rho^2 = s \cdot s, \quad (2.1.57)$$

$$c = s + \frac{1}{2} \rho e_\alpha,$$

y construimos el *trasladador*  $d$  (dicha transformación conformal se verá más adelante) como el vector unitario de la dirección de la línea, lo que resulta en

$$T = 1 + \frac{1}{2} r d e_\alpha, \quad (2.1.58)$$

y así recuperar los puntos como

$$\begin{aligned} p_1 &= TcT^{-1}, \\ p_2 &= T^{-1}cT. \end{aligned} \tag{2.1.59}$$

## Punto

Esta es la entidad geométrica más básica de todas y está en dualidad con la entidad de mayor grado del álgebra que es la esfera. De hecho, el punto puede pensarse como una esfera con radio cero.

Estrictamente hablando, los puntos conformales son obtenidos mediante la proyección estereográfica de los puntos euclidianos sobre la esfera de radio unitario con centro en el origen (ver Figura 18) y se definen como

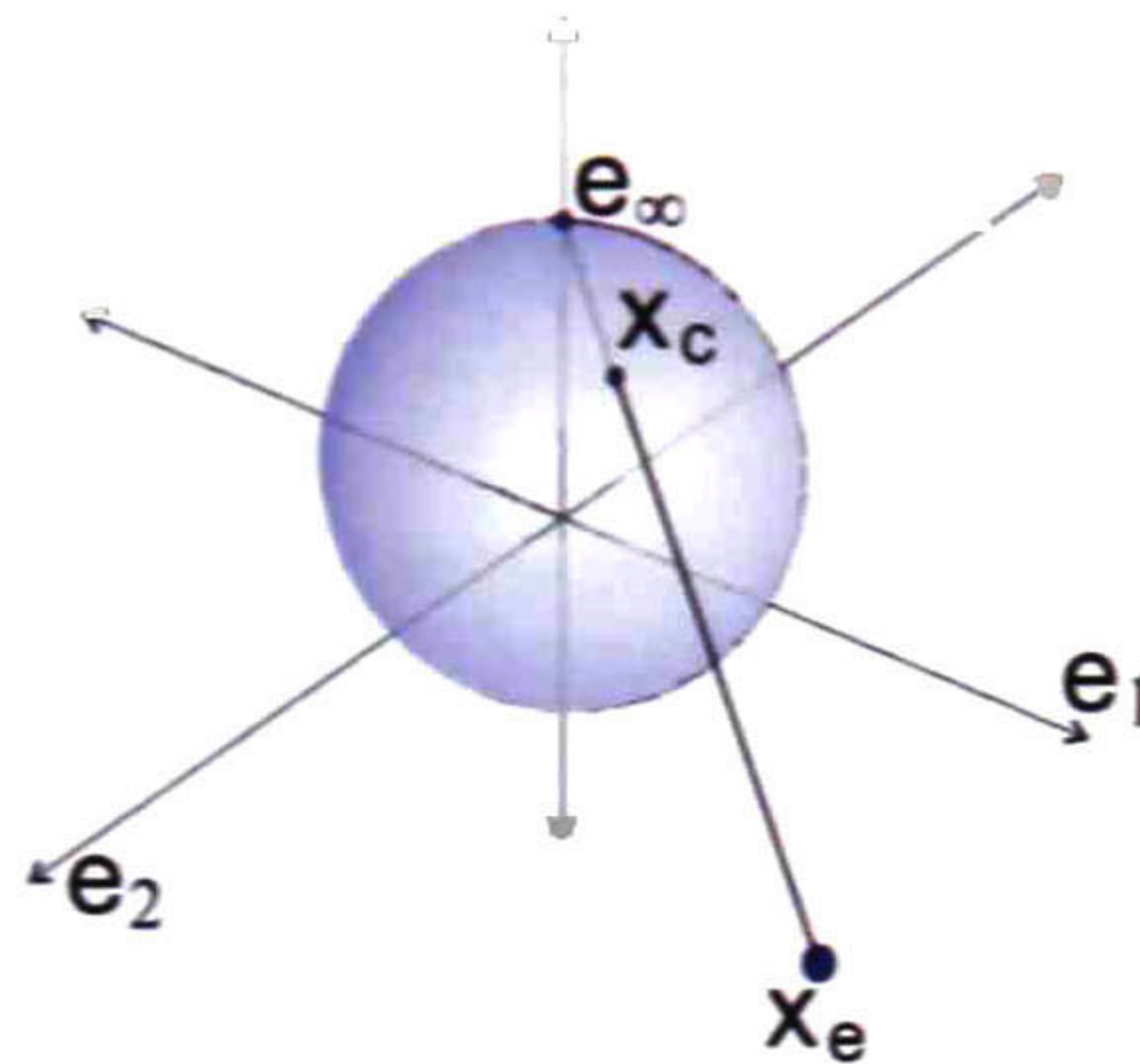


Figura 18. Punto generado por su proyección estereográfica en la esfera unitaria.

$$x = x_e + \frac{1}{2}x_e^2 e_\infty + e_0. \tag{2.1.60}$$

En esta álgebra todos los puntos son elementos del cono nulo, y poseen las siguientes propiedades

$$x \in G_{n+1,1} \mid x^2 = 0, x \cdot e_\infty = -1 \tag{2.1.61}$$

Existe otra forma de representar el mismo punto, la cual se conoce como la forma dual, y consiste en un cuatrivector generado por la intersección de 4 esferas.

$$s^* = x_1 \wedge x_2 \wedge x_3 \wedge x_4 \Leftrightarrow x^* = s_1 \wedge s_2 \wedge s_3 \wedge s_4 \tag{2.1.62}$$

## Reflexión

La reflexión de un punto  $x$  respecto a un plano  $\pi$  es igual al mismo punto menos dos veces la distancia dirigida del plano al punto (ver Figura 19), esto es

$$x' = x - 2 \pi \cdot x \pi^{-1}$$

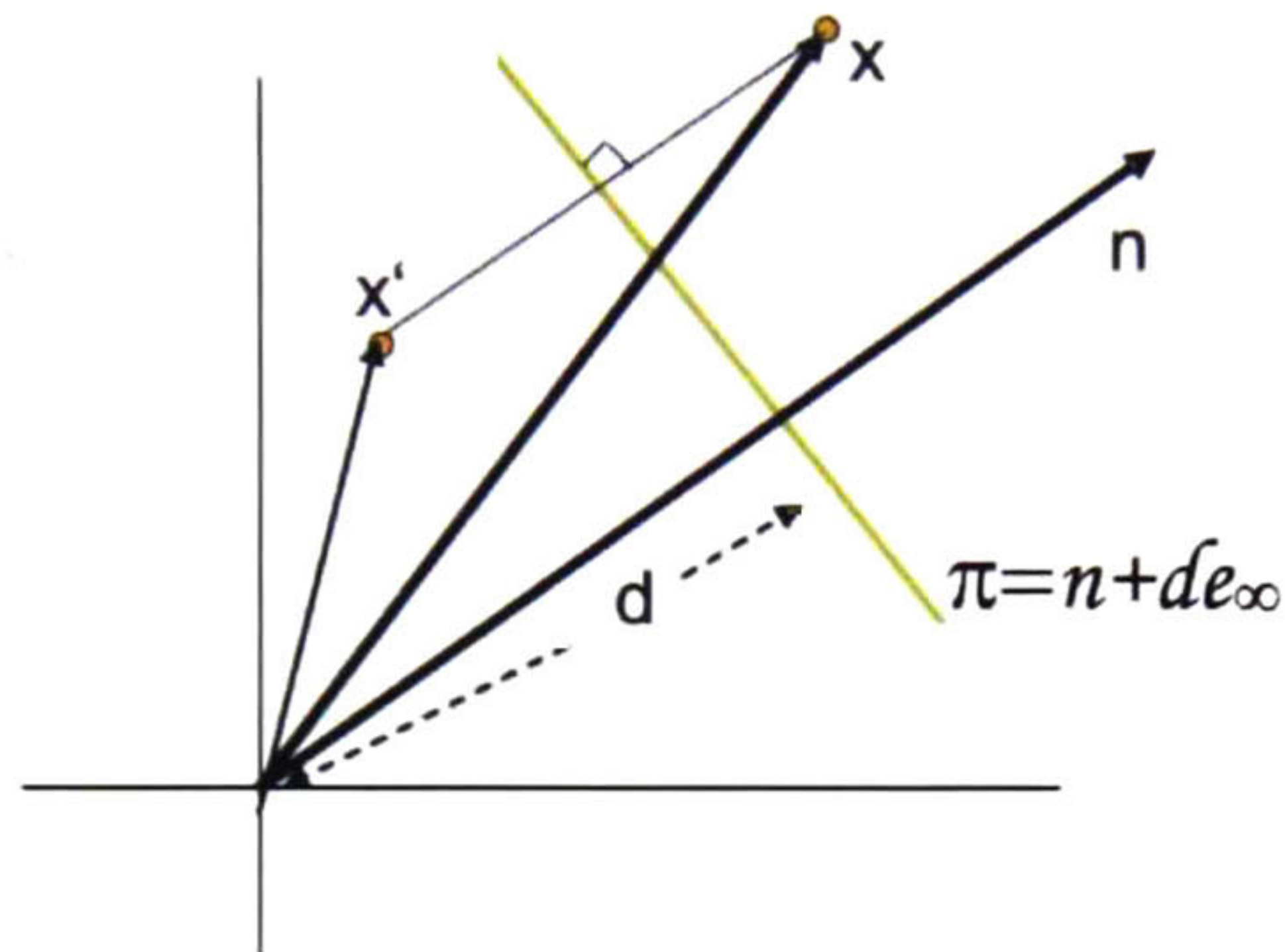


Figura 19. Reflexión de un punto respecto a un plano.

Para simplificar esta expresión, recordemos la propiedad  $ab + ba = 2 b \cdot a$  entonces podemos reescribir a la reflexión como:

$$\begin{aligned} x' &= x - \pi x - x \pi \pi^{-1} \\ &= x - \pi x \pi^{-1} - x \pi \pi^{-1} \\ &= -\pi x \pi^{-1} \end{aligned} \tag{2.1.63}$$

En general, para cualquier entidad geométrica  $Q$ , la reflexión con respecto al plano  $\pi$  está dada por  $Q' = \pi x \pi^{-1}$ .

## Traslación

Esta se puede conceptualizar como dos reflexiones consecutivas respecto a dos planos paralelos (ver Figura 20).

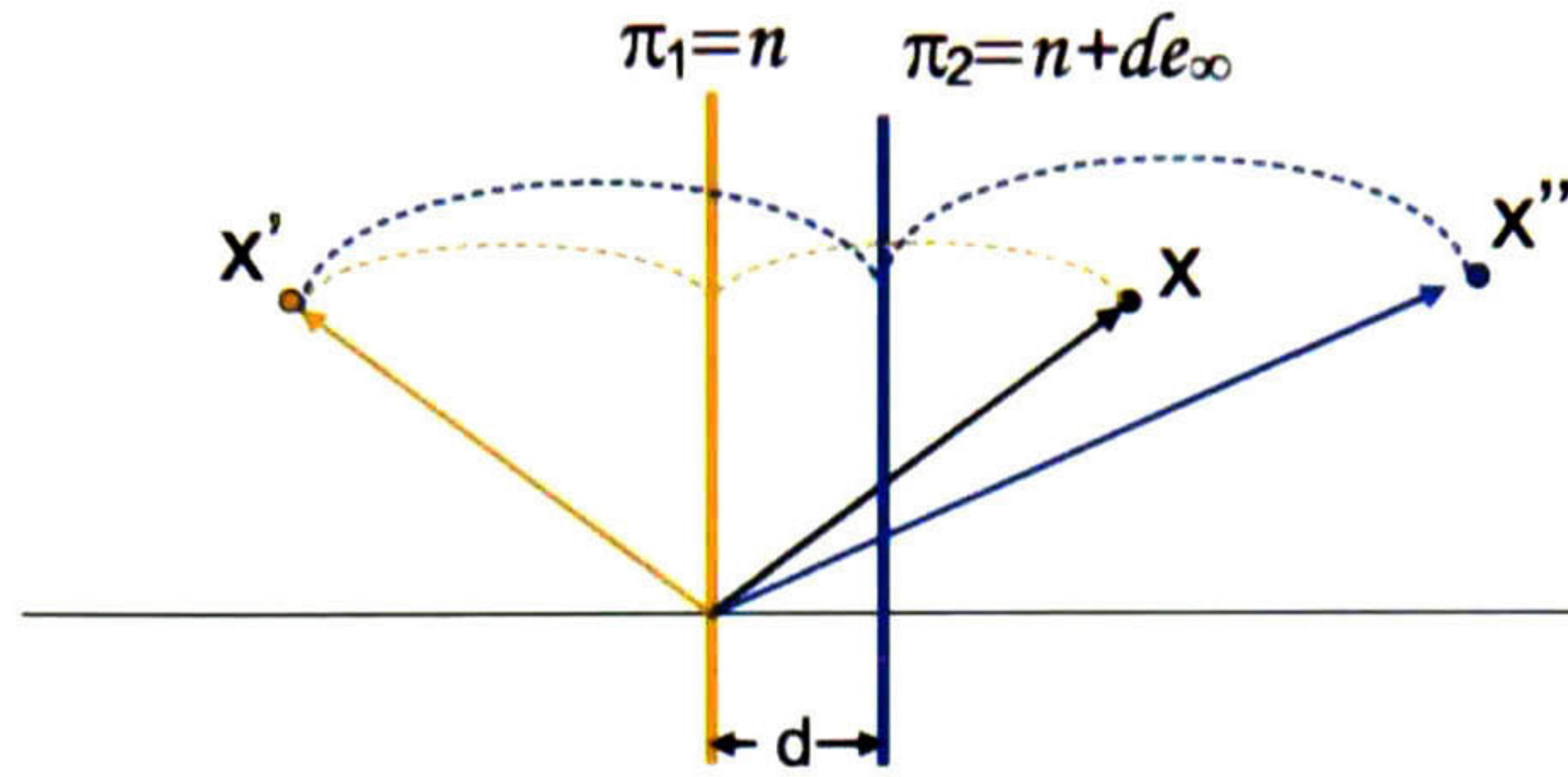


Figura 20. Reflexión entre planos paralelos, la traslación

Sin pérdida de generalidad, se puede considerar que los planos están normalizados (normal de magnitud uno) y donde uno de ellos pasa por el origen. Entonces, la traslación se representa como

$$Q' = \underbrace{\pi_2 \pi_1}_{T_a} Q \underbrace{\pi_1^{-1} \pi_2^{-1}}_{T_a} \quad (2.1.64)$$

con

$$\begin{aligned} T_a &= \pi_2 \pi_1, \\ &= n + de_\infty \quad n + 0e_\infty \\ &= 1 + \frac{1}{2} e_\infty a = e^{e_\infty a/2} \end{aligned} \quad (2.1.65)$$

donde  $a = 2dn$  y  $\|n\| = 1$ . La distancia de traslación es dos veces la distancia entre los planos.

Los trasladadores cuya dirección son los vectores euclidianos base  $e_1, e_2, e_3$  se obtienen como

$$T_{e_1} = e^{-re_1 e_\infty/2}, \quad T_{e_2} = e^{-re_2 e_\infty/2}, \quad T_{e_3} = e^{-re_3 e_\infty/2}, \quad (2.1.66)$$

donde  $r$  es la distancia de traslación.

## Rotación

Se puede describir una rotación por medio de dos reflexiones consecutivas a través de dos planos  $\pi_1$  y  $\pi_2$  que se cruzan en el punto  $c_e$  (ver Figura 21). Supongamos que el punto en común es el origen, entonces los planos tienen la forma  $\pi = n$  y la rotación está dada por

$$Q' = \underbrace{\pi_2 \pi_1}_{R_\theta} Q \underbrace{\pi_1^{-1} \pi_2^{-1}}_{R_\theta} \quad (2.1.67)$$

donde

$$R_\theta = \pi_2 \pi_1 = n_2 \cdot n_1 + n_2 \wedge n_1 \quad (2.1.68)$$

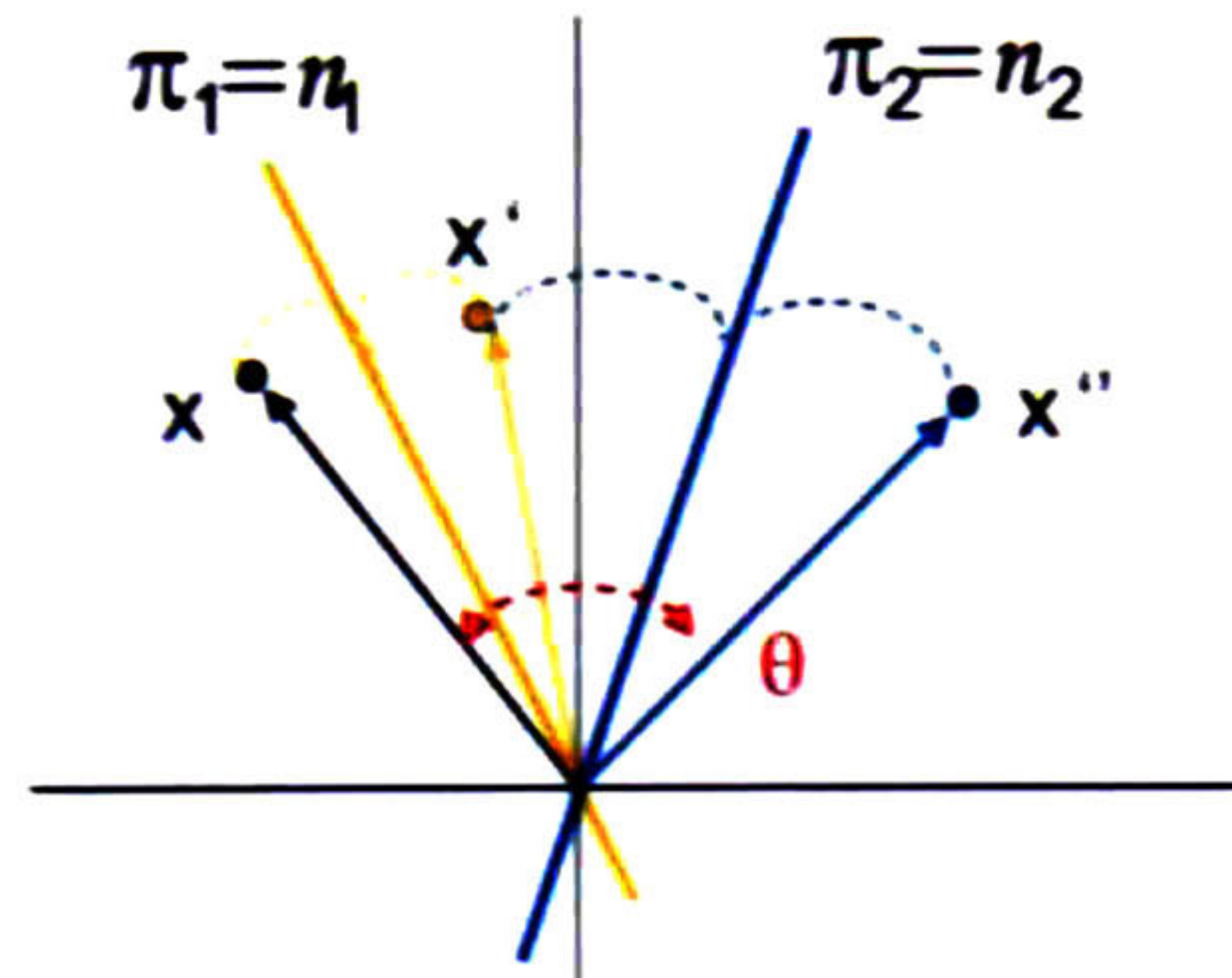


Figura 21. Reflexión entre planos no paralelos, la rotación

Recordando las fórmulas de ángulos entre dos vectores

$$a \cdot b = |a||b| \cos \alpha, \quad a \wedge b = |a||b| \sin \alpha, \quad (2.1.69)$$

es posible re-escribir el rotor de la siguiente manera

$$R_\theta = \cos\left(\frac{\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right)l = e^{-\frac{\theta}{2}l}, \quad (2.1.70)$$

donde  $l = n_1 \wedge n_2$  y  $\theta = 2\alpha$  (dos veces el ángulo entre los planos).

Los rotores cuya dirección son los vectores euclidianos base  $e_1, e_2, e_3$  se obtienen como

$$R_{e_1} = e^{-\theta e_{23}/2}, \quad R_{e_2} = e^{-\theta e_{31}/2}, \quad R_{e_3} = e^{-\theta e_{12}/2}, \quad (2.1.71)$$

donde  $\theta$  es el ángulo de rotación.

### 3. Redes Neuronales

La investigación en redes neuronales artificiales tuvo su origen en 1943, con el trabajo de Warren McCulloch and Walter Pitts [15], los cuales propusieron un modelo simple que podría asemejar el comportamiento de una neurona natural, este modelo es llamado neurona artificial, el cual a la fecha sigue siendo la base para construir las redes neuronales artificiales. Desde entonces y a la fecha, nuevas y más sofisticadas propuestas han sido hechas [16], además que el interés por la investigación en este campo, tanto para su desarrollo en sí mismo así como las posibles aplicaciones ha crecido notablemente.

Un tipo especial de estas redes neuronales fue propuesto por Eduardo Bayro-Corrochano y Sven Buchholds usando el Algebra Geométrica Conformal [5], y las llamaron redes neuronales geométricas. En este trabajo se proponen un nuevo tipo de red, basada en la muy conocida Red Neuronal General para Regresión (conocida como *GRNN* por sus siglas en inglés *General Regression Neural Network*).

### 3.1 Redes de base radial

Las redes de base radial o RBFN son un tipo de redes que calculan su salida en función de la distancia a un centro, su estructura básica se muestra en la **Figura 22**.

Estructura de una red de base radial (RBFN).

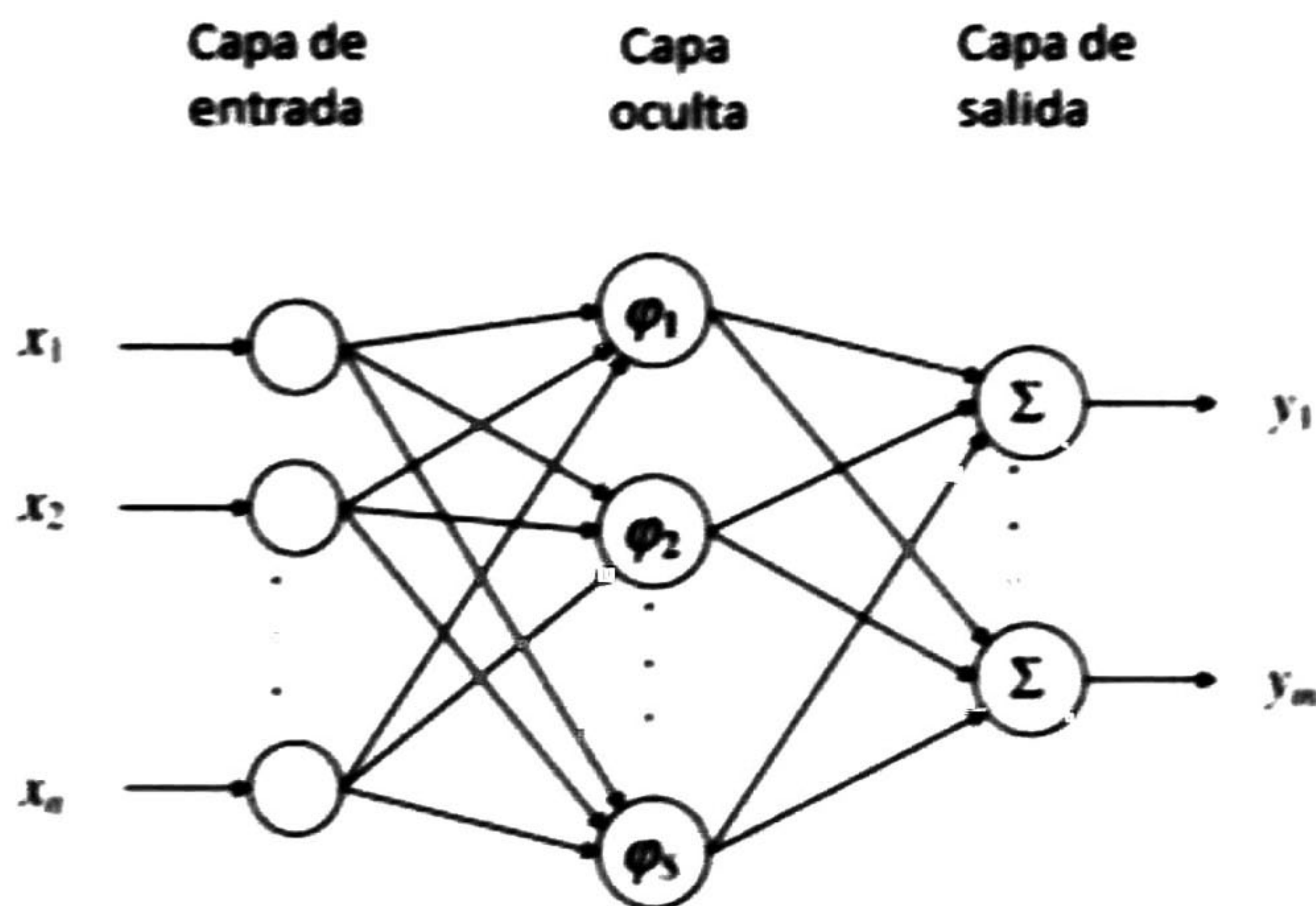


Figura 22. Estructura de una red de base radial (RBFN)

Típicamente esta red tiene una capa de entrada, una capa oculta con una cierta función de activación y una capa de salida. La función de activación debe ser una *función de base radial  $f$* , esto es, el valor de la función se calcula en términos de la distancia entre un vector de entrada  $X$  y un centro  $C$ , en otras palabras:

$$f(X) = f(\| X - C \|). \quad (3.0.1)$$

La función de base radial más comúnmente usada en la literatura es la función Gaussiana.

$$f(X) = e^{-(\epsilon d)^2} \quad (3.0.2)$$

donde  $d=||X-C||$ . En 1990 se demostró que la RBFN usando esta función, o *kernel*, es un aproximador universal [10]. Algunos otros ejemplos clásicos de este tipo de funciones de base radial son:

multicuádrica:

$$f(x) = \frac{\sqrt{d^2 + c^2}}{c}$$

multicuadrática inversa:

$$f(x) = \frac{c}{\sqrt{d^2 + c^2}}$$

spline poliarmónico:

$$f(X) = d^k, \quad k = 1, 3, 5, \dots$$

$$f(X) = d^k \ln(d), \quad k = 2, 4, 6, \dots$$

entre otras. Como ya se mencionó, este tipo de red es un aproximador universal, y esto se logra a través de un proceso de entrenamiento supervisado, tal como se muestra en la Figura 23.

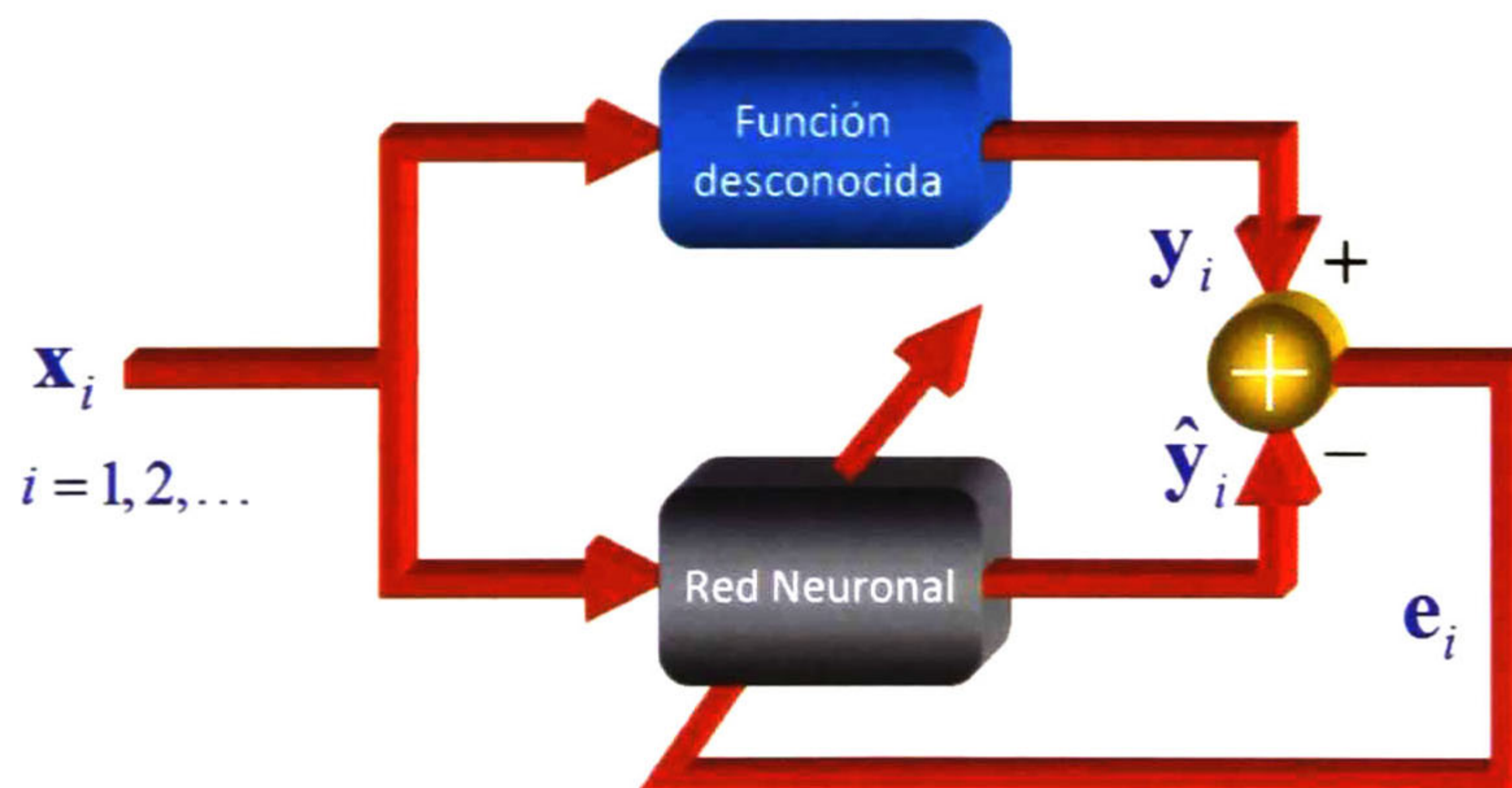


Figura 23. Esquema de entrenamiento para una RBFN



### 3.2 Redes Neuronales Generales para Regresión

Las redes neuronales generales para regresión (más conocidas como GRNN por sus siglas en inglés, General Regression Neural Networks), las cuales fueron propuestas por Donald Specht [17], son redes neuronales que en ocasiones se consideran una de las llamadas redes probabilistas debido a que su estructura es idéntica, pero que tienen una diferencia fundamental entre sí; mientras que las redes probabilistas se usan para clasificación, las GRNN se usan para aproximación de funciones. Se considera que éstas son la versión en redes de lo que se hace llamar en estadística *Nadaraya-Watson kernel regression* [18].

Este tipo de redes son muy usadas para aplicaciones en tiempo real ya que el algoritmo de entrenamiento es de una sola época y suelen converger con unas pocas muestras. por ejemplo se han usado para el ahorro en consumo de energía [19] y para predicción espacial [20]. Su estructura se muestra en la Figura 24.

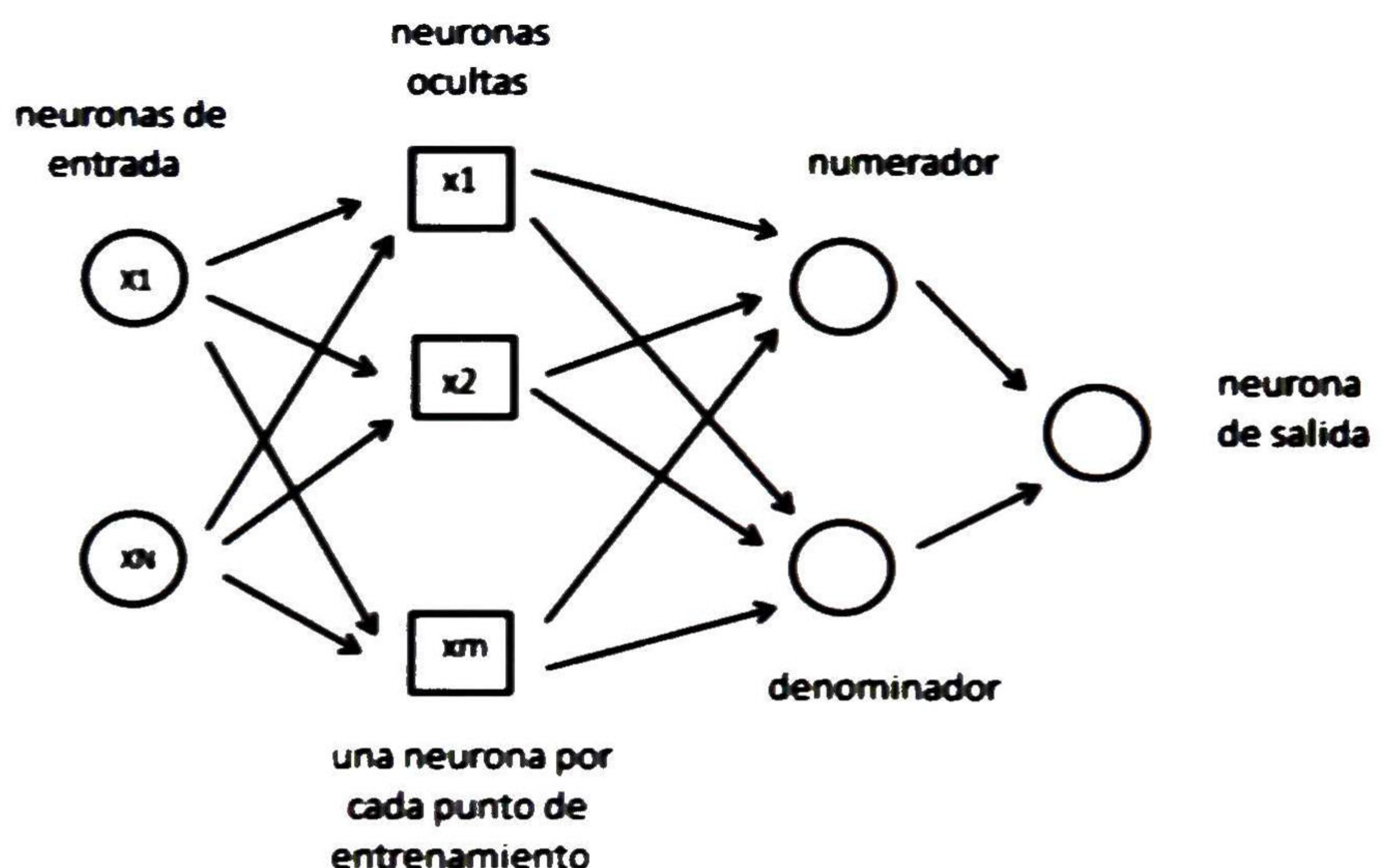


Figura 24. Estructura de una GRNN

Como se puede observar, consta de una capa de entrada, una capa de neuronas ocultas que como el caso de las muy conocidas Redes de Base Radial (RBFN por sus siglas en inglés) y de igual forma debe tener un kernel correspondiente a una función de base radial, de ahí se calcula una razón la cual da la salida de la red. Igual que en las RBFN, el kernel más usado en la literatura es la función Gaussiana (3.1.2). Si llamamos  $G$  a la salida de la red, de acuerdo a la estructura presentada en la Figura 24 tenemos que esta viene dada como:

$$G(x) = \frac{\sum_i w_i f(b \| x - c_i \|^2)}{\sum_i f(b \| x - c_i \|^2)} \quad (3.0.3)$$

donde  $x$  es cualquier punto del dominio,  $f(\bullet)$  es la función Gaussiana,  $w_i \in W$  siendo  $W$  el llamado vector de pesos, el cual generalmente se selecciona inicialmente de manera aleatoria y  $c_i \in C$ , donde  $C$  es el vector de centros (observe la gran similitud con las RBFN).

Veamos un ejemplo de su uso, dada una función desconocida y un vector de entrada  $X$ . el proceso de entrenamiento consiste en que la red cambiando algunos parámetros minimiza la diferencia entre su propia salida y la salida de la función desconocida, es decir, la red *copia* o *aproxima* de alguna manera la función desconocida. Vamos a explicar esto de una forma más detallada. supongamos que tenemos la función en una dimensión que se muestra en la Figura 25, en la cual los puntos rojos representan el vector con el cual se pretende aproximar tal función.

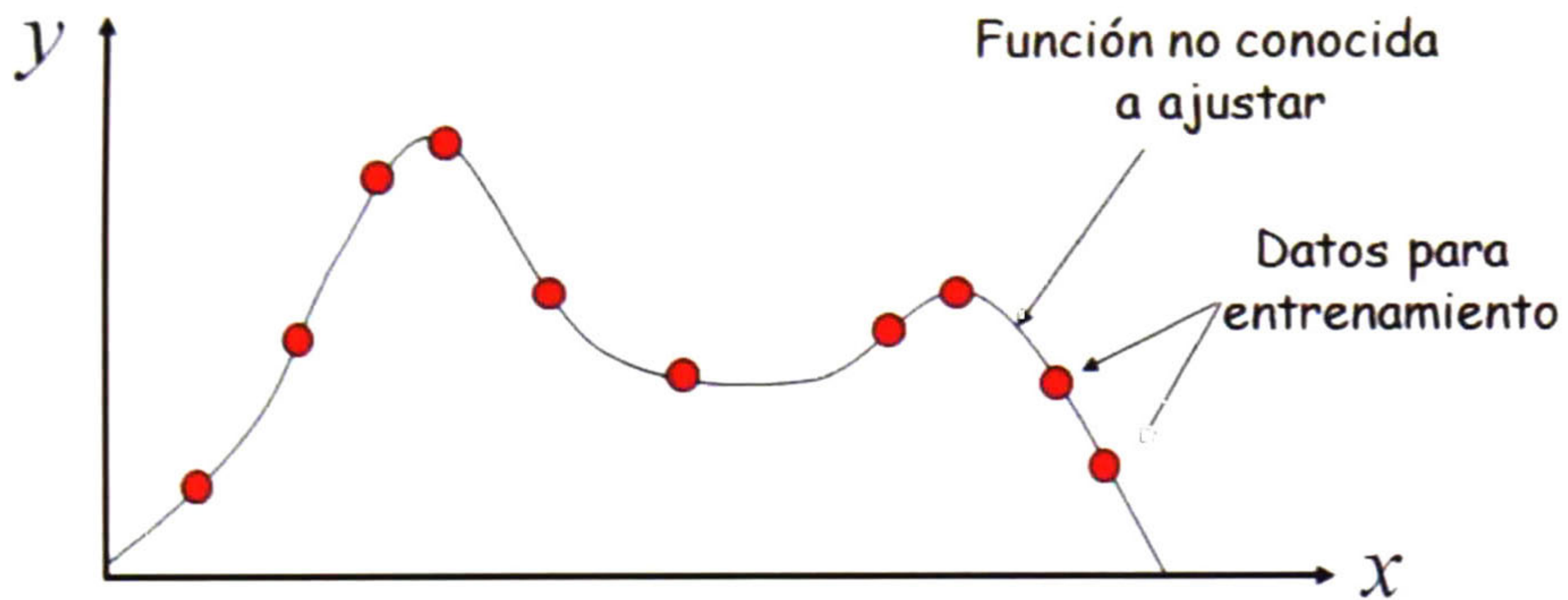


Figura 25. Función desconocida a aproximar.

Durante el proceso de entrenamiento, la red sitúa una serie de funciones o kernels a lo largo del eje  $x$ , cada una de ellas con diferente centro, la cantidad de estas funciones puede fijarse desde antes del proceso de entrenamiento o puede fijarse una cantidad pequeña de las mismas y en el proceso de entrenamiento se incrementa este número hasta que se alcanza una cota de error de la diferencia entre la red y la función a aproximar. Vamos a suponer que usamos la función Gaussiana (aunque se podría usar cualquier otra de las funciones de base radial), como se muestra en la Figura 26 . Si llamamos  $F$  a la función desconocida, y  $\hat{F}$  a la salida de la red, tenemos que:

$$\hat{F}(x) = \sum_i w_i f(\|x - c_i\|) \quad (3.0.4)$$

donde  $x$  es cualquier punto del dominio,  $f(\bullet)$  es la función Gaussiana,  $w_i \in W$  siendo  $W$  la matriz de vectores de pesos, la cual generalmente se selecciona inicialmente de manera aleatoria y  $c_i \in C$ , donde  $C$  es el vector de centros. Durante el proceso de entrenamiento se cambian los valores de  $C$  (optimización no lineal) y los valores de  $W$  (optimización lineal) de tal manera que  $|F - \hat{F}| \rightarrow 0$

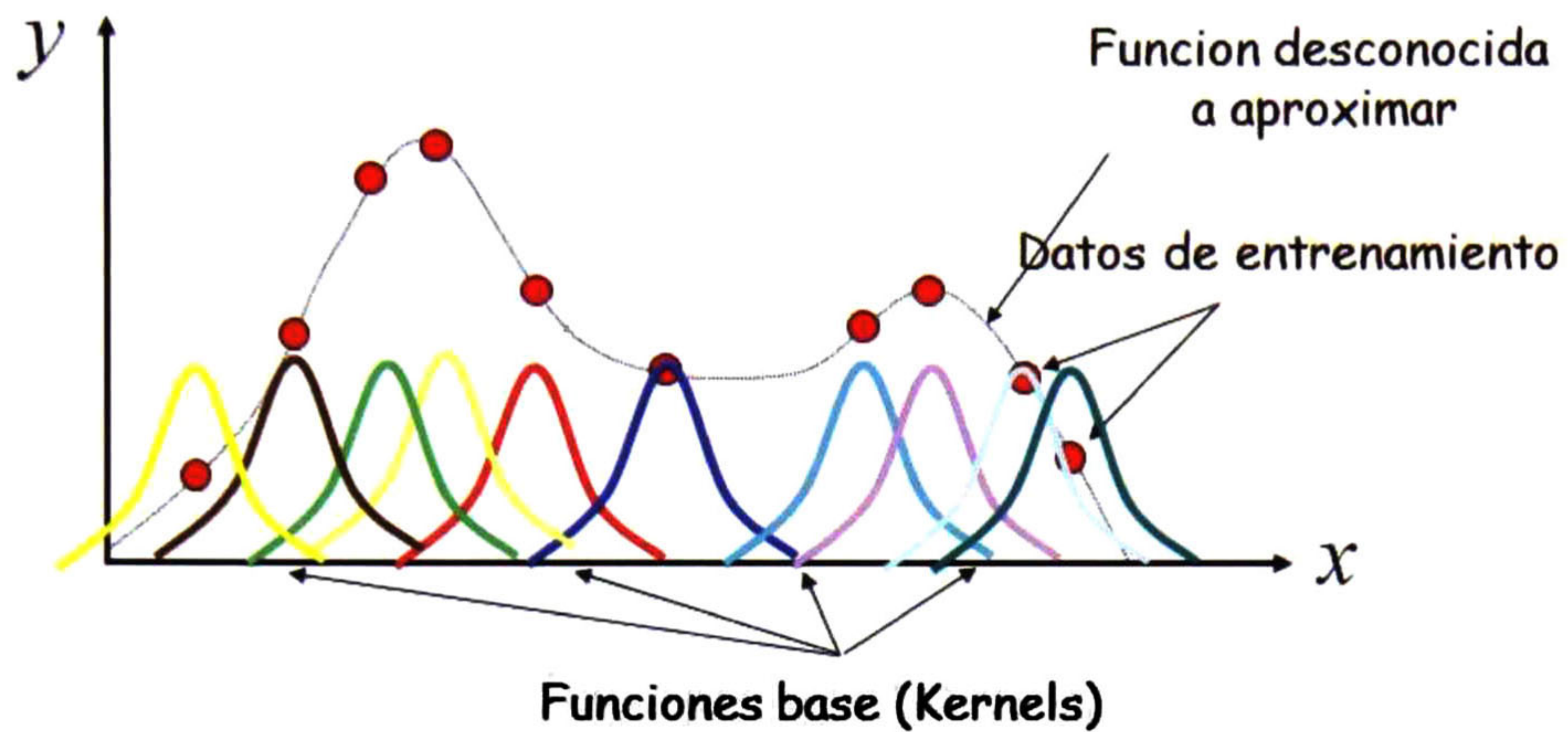
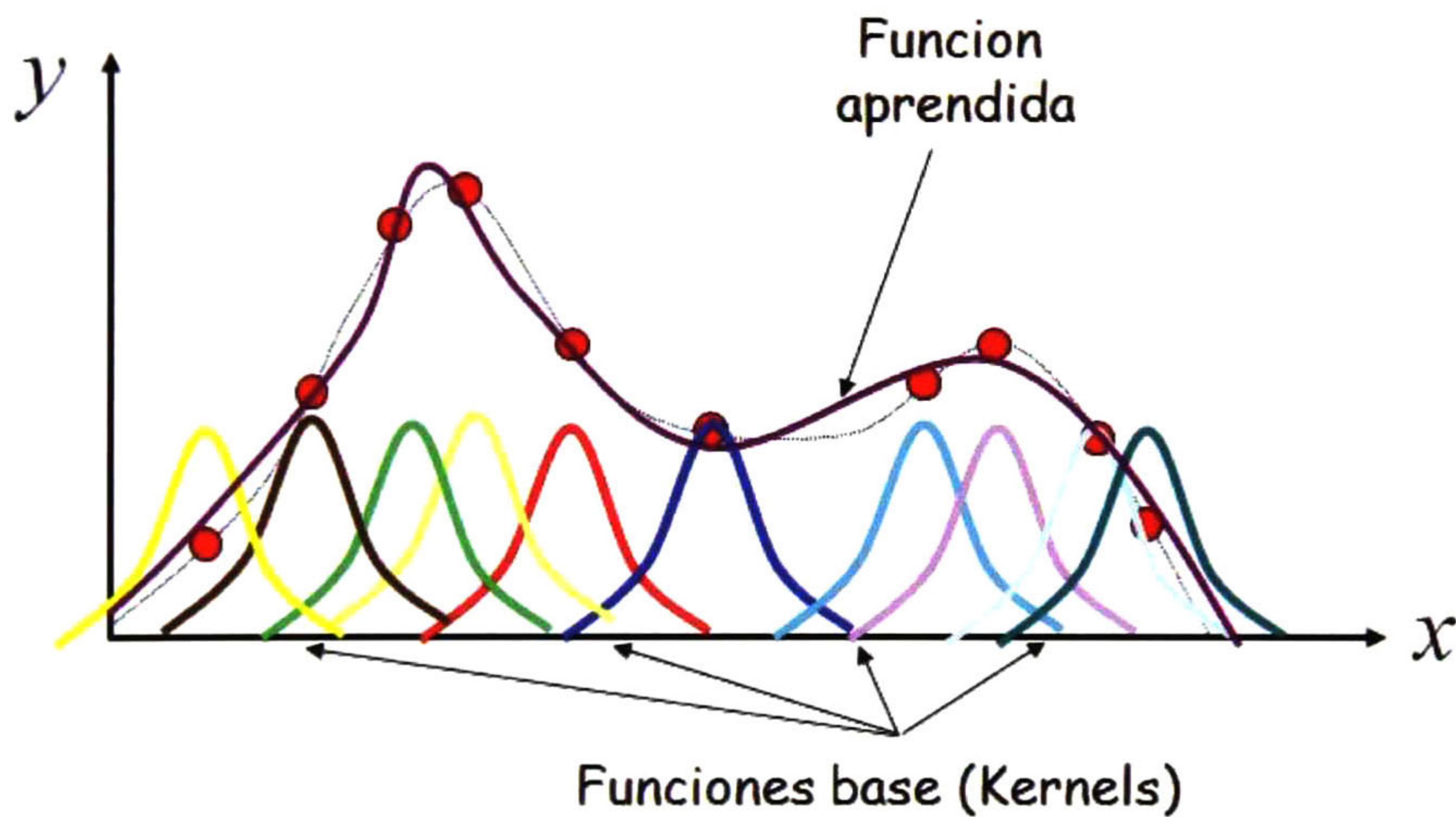


Figura 26. Kernels propuestos para el entrenamiento de la red

Después del entrenamiento, la red genera una nueva función que aproxima a la función original  
 Figura 27.



### 3.3 Redes Generales Esféricas para Regresión.

Igual que antes, proponemos un nuevo tipo de red geométrica, planteada en álgebra conformal, usando hiperesferas como funciones de activación, aprovechando

igual que antes las bondades que tiene trabajar con estos elementos geométricos en dicho marco matemático.

Esta red suele presentar las ventajas de las redes de base radial, con la ventaja adicional que es mucho más rápida de entrenar. El entrenamiento de esta red consiste en encontrar el mejor vector de pesos para minimizar la diferencia entre la salida de la red y la función a aproximar, con una gran ventaja respecto a la RBFN, en este caso los centros quedan fijos, por lo que solo hay que realizar un ajuste lineal de mínimos cuadrados. El parámetro  $b$  es un parámetro que representa el radio seleccionado de los círculos (en una dimensión), y que en la práctica se asocia a la suavidad de la curva obtenida en la aproximación. El usuario debe seleccionar este valor para tener el mejor ajuste en un caso particular. Existen algunas propuestas heurísticas en este sentido principalmente para las GRNN que usan Gaussianas como kernels y que podrían extrapolarse a esta nueva red (en este documento se propone una heurística que puede ayudar a su selección). Existe un punto adicional que debe discutirse para esta red y que tiene que ver con la selección de este parámetro, sucede que en el caso de la función Gaussiana sabemos que su dominio va de  $(-\infty, \infty)$ , pero los círculos tienen un dominio finito, esto hace que  $b$  debe tener un valor mínimo para garantizar que la aproximación de la curva se en todo su rango y no presente huecos. Para entender esto vemos la Figura 27, esta figura representa el dominio de una función en los reales, el intervalo  $(a,b)$ , en el cual se seleccionan una serie de puntos (rojo) para realizar la regresión de una función. A la distancia más grande entre 2 puntos consecutivos la llamamos  $d_{max}$ , en el caso de que  $d \leq \frac{d_{max}}{2}$ , existiría una discontinuidad en la función aproximada justo en el centro donde se localiza  $d_{max}$ , por lo cual debemos garantizar traslape entre los círculos, en otras palabras se debe cumplir que:

$$d > \frac{d_{max}}{2} \quad (3.1.1)$$

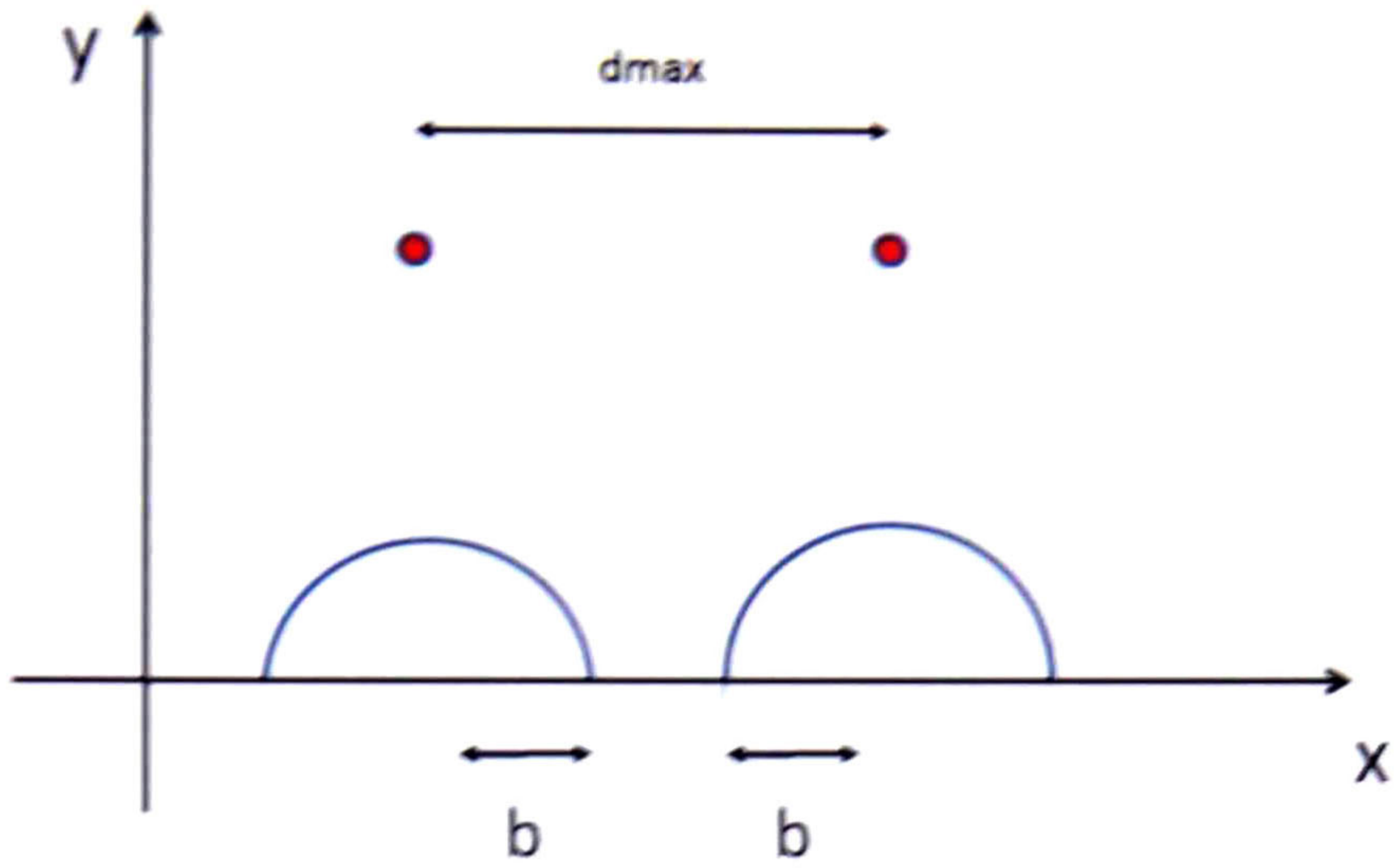


Figura 27. Se debe garantizar que  $2b > d_{max}$  para garantizar el traslape entre círculos

### 3.3 Modelo Neuronal Tacto-Visual

Vamos a suponer que tenemos un objeto tridimensional, como el que se muestra en la Figura 24.

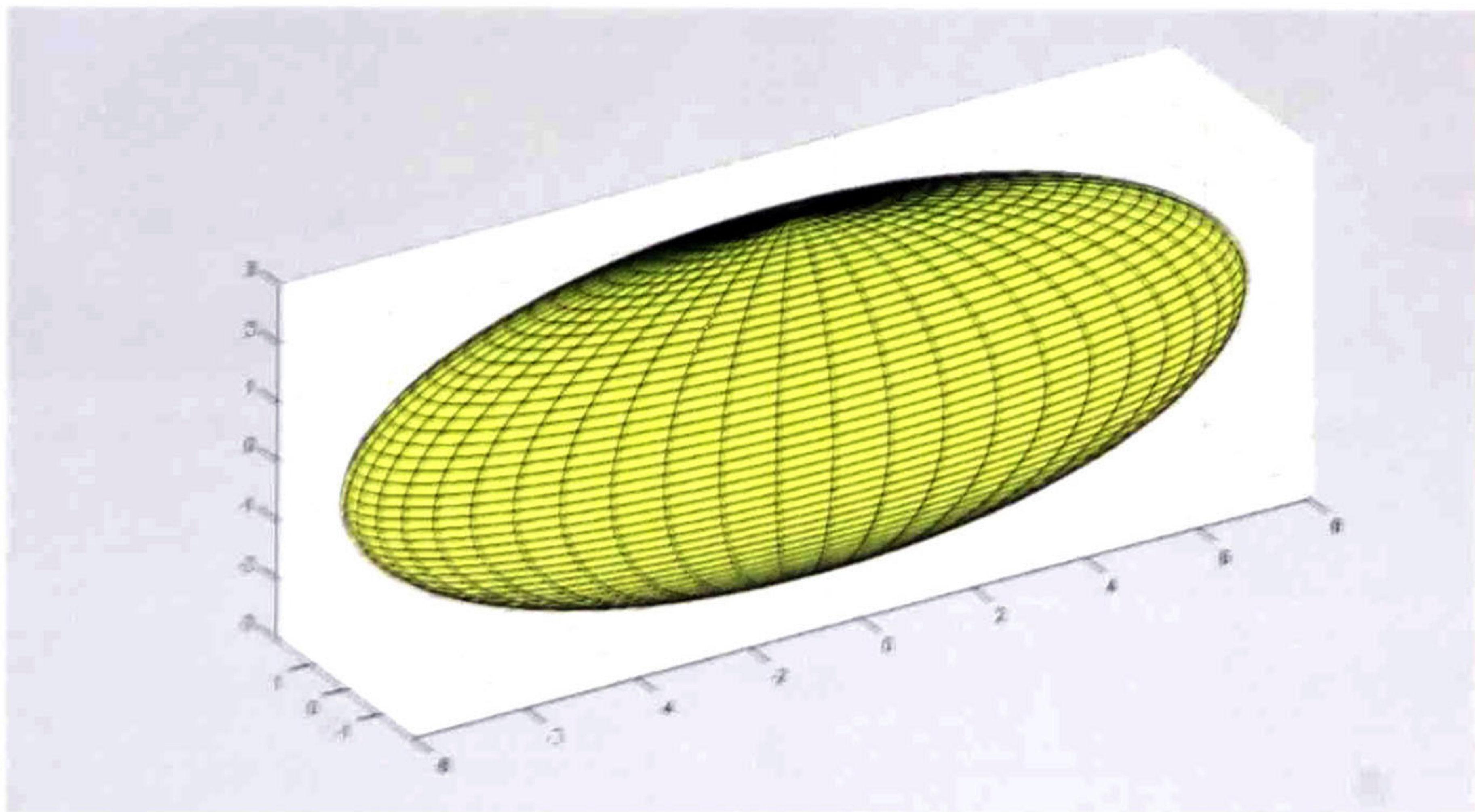


Figura 28. Objeto tridimensional en forma de elipsoide.

Estamos interesados en modelar las propiedades físicas superficiales del objeto como su rigidez o su temperatura, las cuales pueden percibirse con el sentido del tacto de una persona o de una criatura biológica; para esto el objeto puede dividirse en un mallado tridimensional muy fino a lo largo de su superficie, donde cada uno de los elementos resultantes mide de ancho y largo  $\Delta x$  y  $\Delta y$  respectivamente (sin pérdida de generalidad, se van a considerar elementos planares). Estamos interesados en modelar la elasticidad del objeto sobre su superficie, un modelo simple de esto sería considerar que cada uno de los elementos de la malla tiene un resorte en la parte de abajo, y que la constante elástica del resorte ( $k$ ) es una medida de la rigidez del objeto en ese elemento de superficie. Con esto en mente podemos decir que para cada elemento de la malla superficial, existe una rigidez asociada  $k$ , la cual se define como la resistencia que opone un cuerpo elástico a la deformación, entonces el  $i$ -ésimo elemento tiene una rigidez  $k_i$  que se puede calcular como:

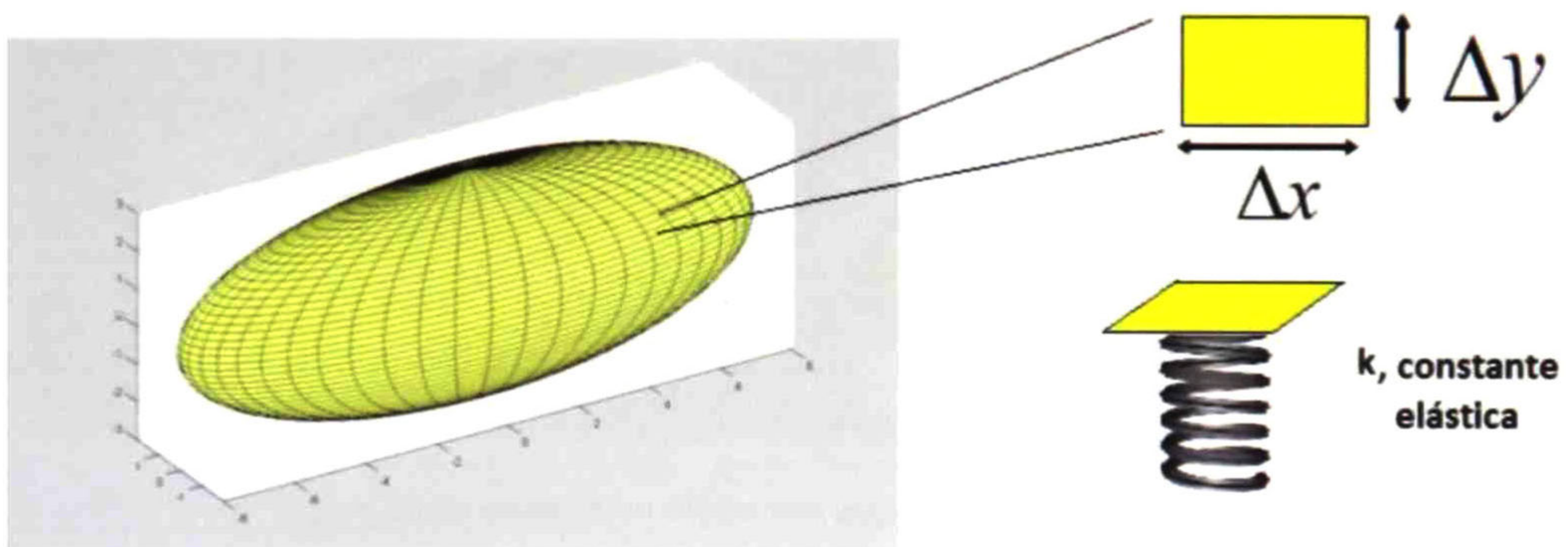


Figura 29. Modelo discreto de la rigidez de la superficie de un cuerpo tridimensional.

$$k_i = \bar{F} / \bar{d} \quad (3.2.1)$$

donde  $\bar{F}$  es la fuerza aplicada al elemento y  $\bar{d}$  es el desplazamiento logrado en la dirección de dicha fuerza. Tomando el límite cuando  $\Delta x, \Delta y \rightarrow 0$ , entonces

podemos afirmar que a lo largo de la superficie del objeto tridimensional existe una función escalar  $K(x, y, z)$  que representa la rigidez superficial del mismo, a lo largo de este trabajo nos estaremos refiriendo a esta función como la función de rigidez superficial.

La función de rigidez superficial podría tener un valor constante, lo cual significaría que la rigidez del objeto referido tiene las mismas propiedades mecánicas sobre su superficie, pero supongamos que tenemos un objeto donde esto no es así, si no que en su superficie se forma una protuberancia (la cual puede ser o no visible) en la cual la rigidez del objeto se ve alterada. En la Figura 30 se puede apreciar un elipsoide en el cual se muestra en amarillo claro la mayor parte de su superficie, pero se muestra una protuberancia la cual visualmente se representa con el color tendiente al rojo. En medicina tenemos un ejemplo muy claro de cuerpos que presentan estas propiedades, como lo son órganos que tienen tumoraciones o calcificaciones superficiales (las cuales puede ser o no visibles).

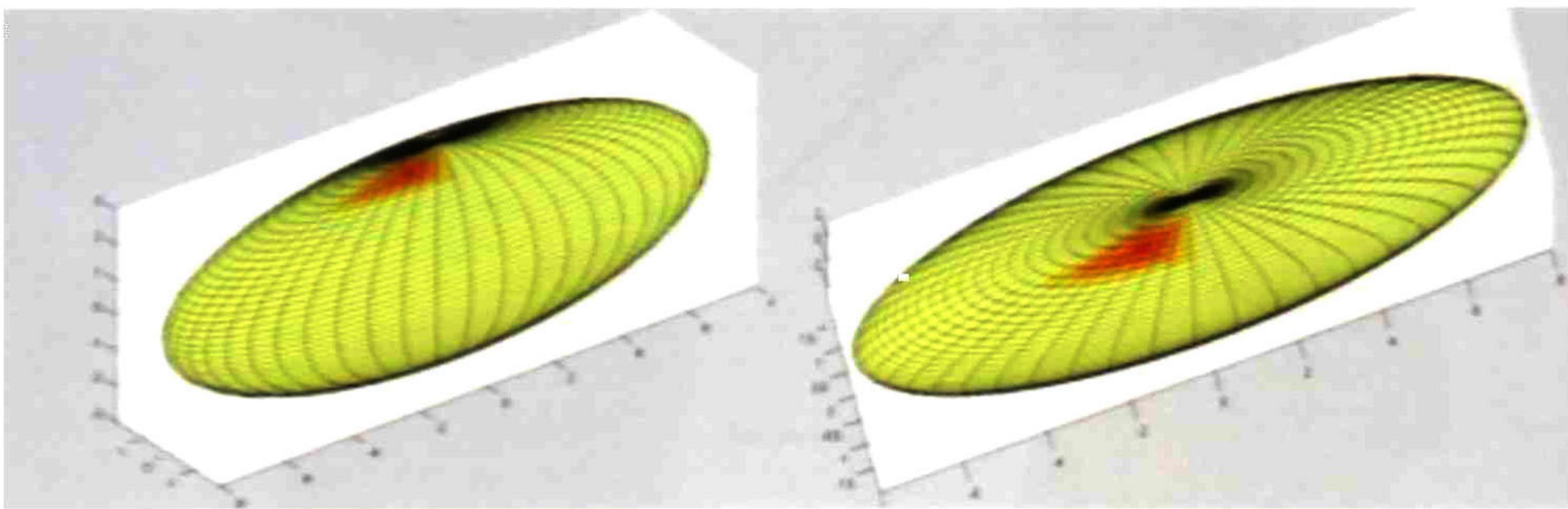


Figura 30. Se muestra un objeto con una protuberancia superficial

Como se ha dicho desde la introducción, la principal propuesta de este trabajo consiste en fusionar 2 sentidos de los seres vivos: el sentido táctil y el sentido visual. Por ejemplo, desde una perspectiva visual este objeto puede ser uniforme a lo largo de su superficie (en este caso la figura no lo es), pero desde el punto de vista táctil el objeto tiene una protuberancia. El modelo aquí propuesto hace una representación mezclada de ambos sentidos.



Para dar algunas ideas sobre este modelo, partimos de un objeto tridimensional de forma irregular, la idea es tomar dicho objeto y realizar un llenado con esferas que no se traslapen entre sí. preferentemente de diferentes radios para optimizar el llenado. Esta idea fue presentada originalmente en [21] donde se presenta un algoritmo para representar objetos tridimensionales no rígidos usando esferas de diferentes radios, este algoritmo fue llamado *marching spheres* por su similitud al algoritmo *marching cubes* [22]. También en [23] se presenta un algoritmo para realizar el llenado de esferas de objetos tridimensionales llamado *sphere packing algorithm*. En la Figura 31 se puede observar un ejemplo de su uso.

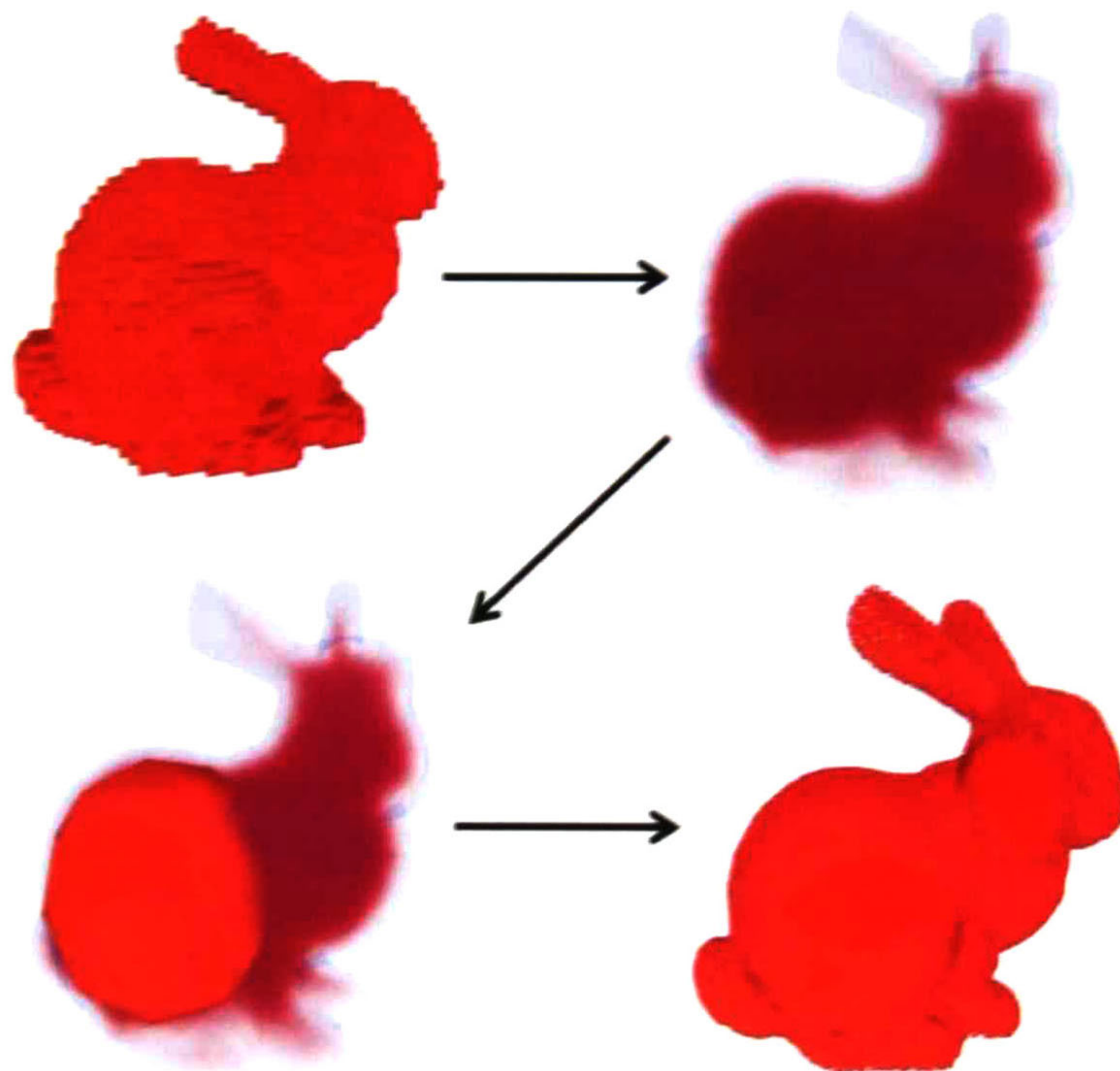


Figura 31. Ejemplo del uso del *sphere packing algorithm* (tomado de [23]).

Con estas ideas en mente podemos ahora presentar nuestro modelo, el cual se implementa en 4 pasos:

1. Se toma una nube de puntos que representa un objeto tridimensional y/o su superficie y se realiza un "llenado" o "reconstrucción" del mismo usando esferas. En este caso se optó por usar el *sphere packing algorithm* antes mencionado. Cabe hacer mención que en nuestro modelo solo estamos interesados en la superficie, por lo que solo estamos interesados en las esferas exteriores.
2. Se toma un conjunto de puntos muestrales sobre la superficie, a los cuales se les calcula su rigidez superficial  $k_i(x,y,z)$ .
3. A través de una Red Neuronal General para Regresión (GRNN) se interpola la función de rigidez superficial  $K(x,y,z)$ . En este punto cabe hacer mención que para la interpolación solo se usan las coordenadas planares  $(x,y)$ , es decir solo se considera la rigidez en cada punto como  $k_i(x,y)$ , esto es porque la rigidez cambia en el interior del cuerpo y en este trabajo no estamos interesados en modelar la rigidez dentro del cuerpo, solo de su superficie.
4. Después de entrenar la red, y como último paso se calcula la rigidez superficial en la coordenada  $(x,y)$  del centro de cada esfera, con lo cual tenemos un conjunto de esferas exteriores que representan visualmente al objeto cada una de las cuales tiene un valor de rigidez.

Hemos llamado a esta metodología *Modelo Neuronal Tacto-Visual*. Los pasos que se describieron antes se resumen en la Figura 32.

## ESQUEMA DEL MODELO NEURONAL TACTO-VISUAL

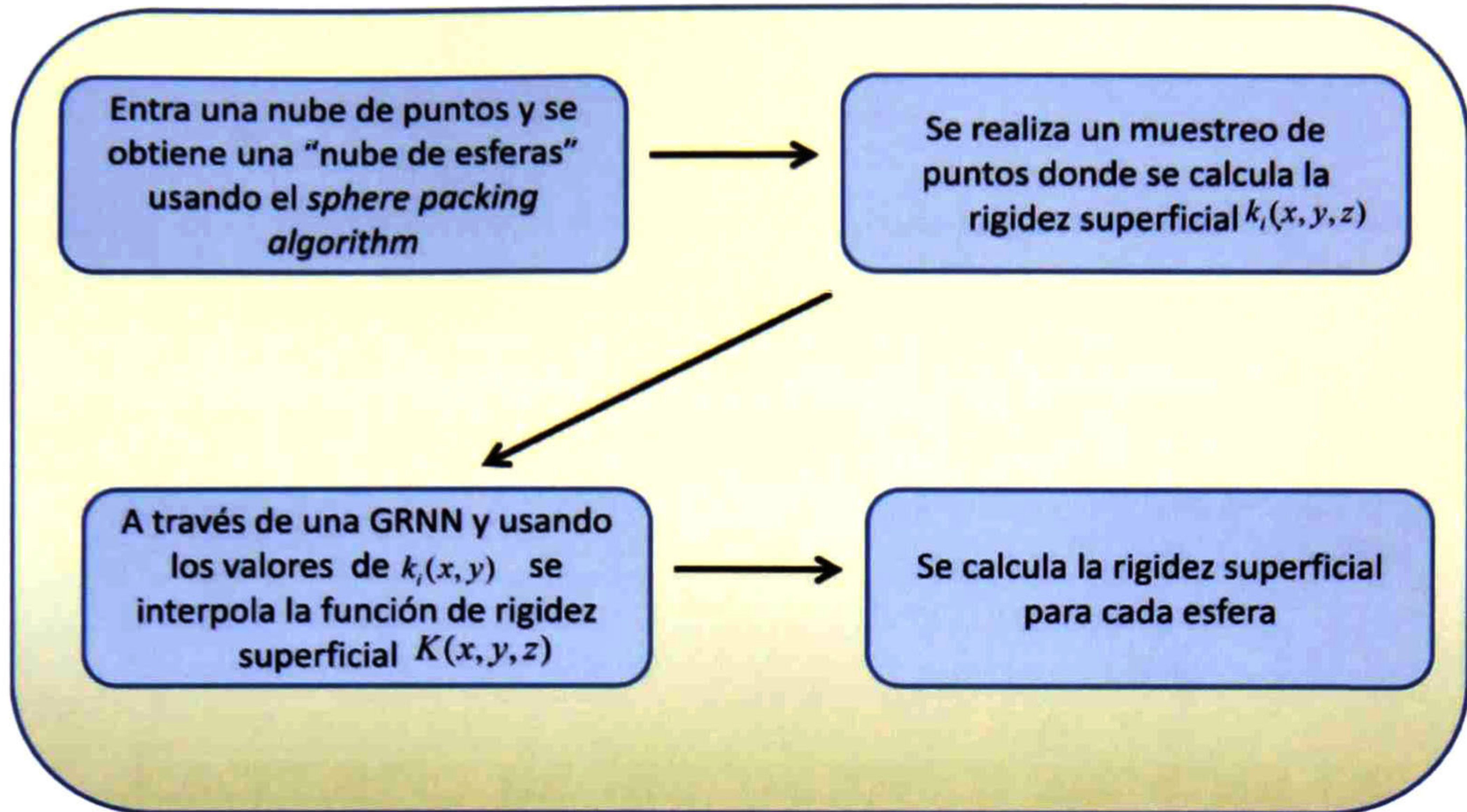


Figura 32. Modelo Neuronal Tacto Visual

Después de aplicar este modelo a un cuerpo, obtenemos una "nube de esferas" de las cuales la GRNN nos da un valor de su rigidez para las exteriores, esto se puede entender mejor observando la Figura 33.

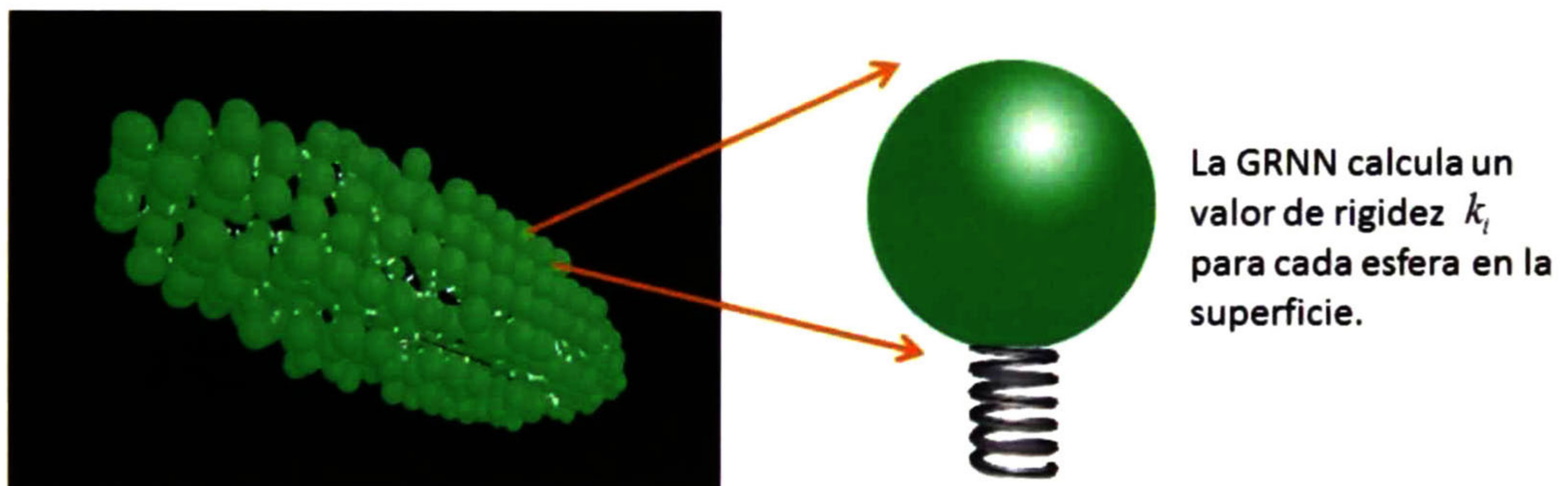


Figura 33. Se muestra el resultado final que se obtiene al aplicar el Modelo Neuronal Tacto-Visual

## 4. Escenario de laboratorio: sistema de cirugía de mínima invasión.

En esta sección se describe brevemente un escenario de laboratorio montado para probar en un caso real el Modelo Neuronal Tacto-Visual, que además sirve como base para la construcción del sistema de cirugía de mínima invasión de bajo costo.

### 4.1 Brazos robot

Uno de las componentes principales del sistema de cirugía de mínima invasión y también necesario para probar el modelo en un escenario real son los brazos robot. De inicio, en los sistemas comerciales se tienen pequeños brazos fabricados a la medida, se evaluó la factibilidad de construir aquí mismo en CINVESTAV dichos brazos, pero debido al tiempo que esto llevaría, incluyendo el desarrollo de los controladores adecuados, podría irse a varios años su diseño y construcción. Otra posibilidad que se evaluó fue el de construirlos en algún centro de investigación que tenga el suficiente expertiz en el área, pero realizar tal cosa excedía por mucho los límites de presupuesto del proyecto, por lo que se optó por reutilizar un pequeño brazo robot con el que ya se contaba en el laboratorio de

control, el cual estaba inhabilitado por el tiempo de desuso, pero el cual estamos en el proceso de puesta en marcha a través del cambio de la memoria RAM y de su recalibración. Este robot es el ADEPT Six 600, que es un brazo robot articulado semi-industrial de 6 grados de libertad fabricado por la empresa ADEPT Technology Inc (Figura 34). Se construyó un efector final para este brazo donde se puede montar un endoscopio estéreo rígido, el cual actuará como los ojos del médico dentro del cuerpo del paciente. Este robot puede ser manipulado a través de un joystick simple, ya que no requiere realimentación táctil. Aunque no tiene una interfaz para poder programarse en algún lenguaje de uso general como C/C++, este robot puede ser programado en su lenguaje nativo y se programó una interfaz que puede comunicar con otro lenguaje a través del puerto serial.

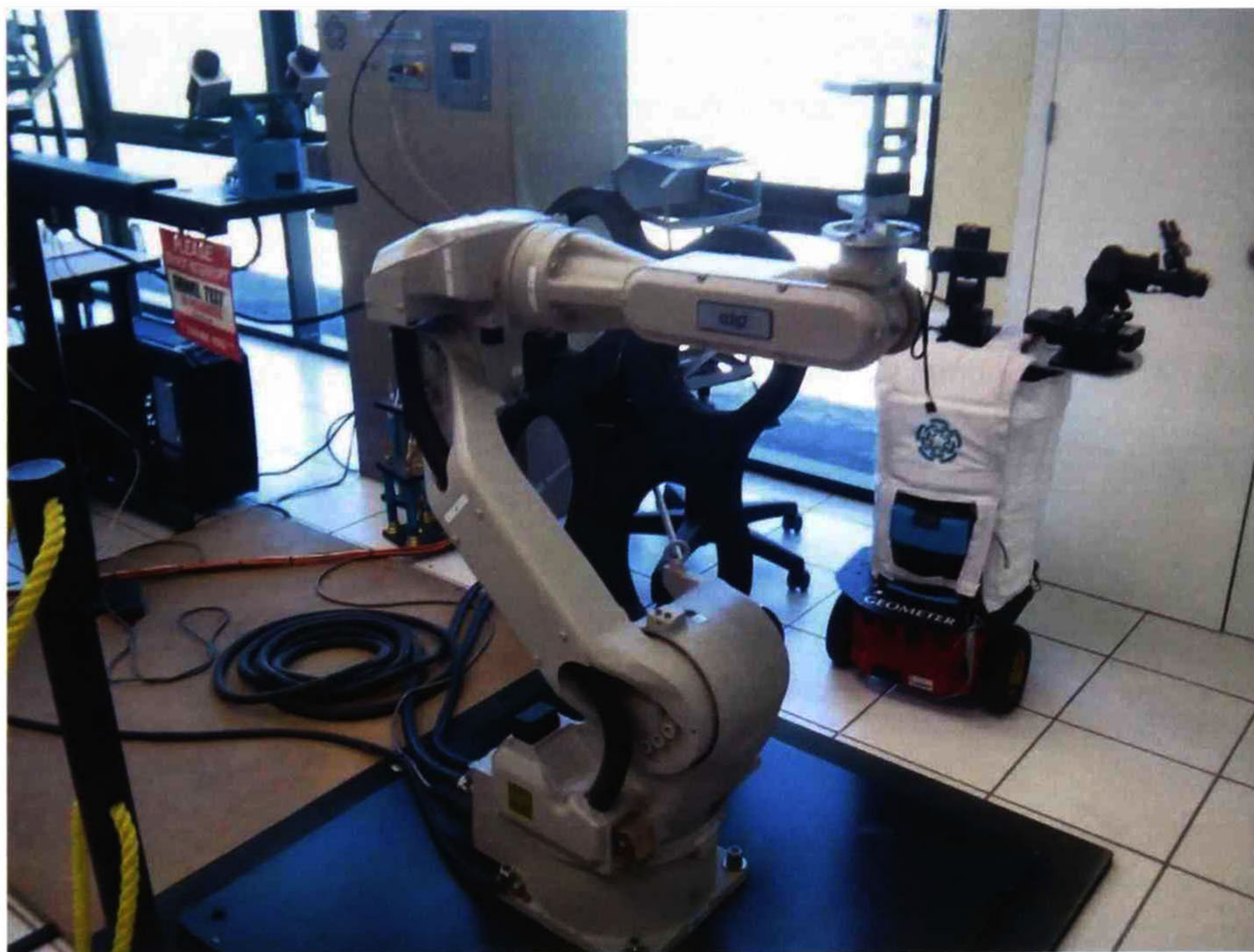


Figura 34. Robot ADEPT 6 600, brazo robot de 6 grados de libertad

Para tal sistema de cirugía se requieren mínimo de 3 brazos de este tipo, ya que uno de ellos debe sostener el endoscopio, mientras que los 2 restantes servirían para actuar como las manos del médico, estos 2 últimos deberán llevar el instrumental quirúrgico necesario para la operación, y deberán ser controlados por

una interfaz háptica que cuenta con un sensor de fuerza en el efector final de tal manera que la persona que lo manipule a través de la interfaz pueda sentir lo que percibe el robot.

Se realizó un comparativo entre robots industriales de diferentes marcas como MITSUBISHI, KUKA, ABB, FANUC, MOTOMAN entre otros, poniendo como principales requisitos:

- capacidad de carga superior a 500 gr.
- alcance horizontal frontal mayor a 70 cm
- lo más ligero posible
- programación abierta en lenguaje de uso general (C/C++)
- seguridad al choque
- bajo costo
- repetitividad y precisión superior a 0.1 mm.
- Montaje en suelo, pared o techo.

El robot que cumplió de forma satisfactoria estos requerimientos fue el robot MH5LF de MOTOMAN (Anexo 1.2, página 96) el cual se puede ver en la Figura 35.



Figura 35. Brazo robot MH5LF de MOTOMAN

El sistema completo de 3 robots se ordenó de acuerdo al arreglo mostrado en la Figura 36

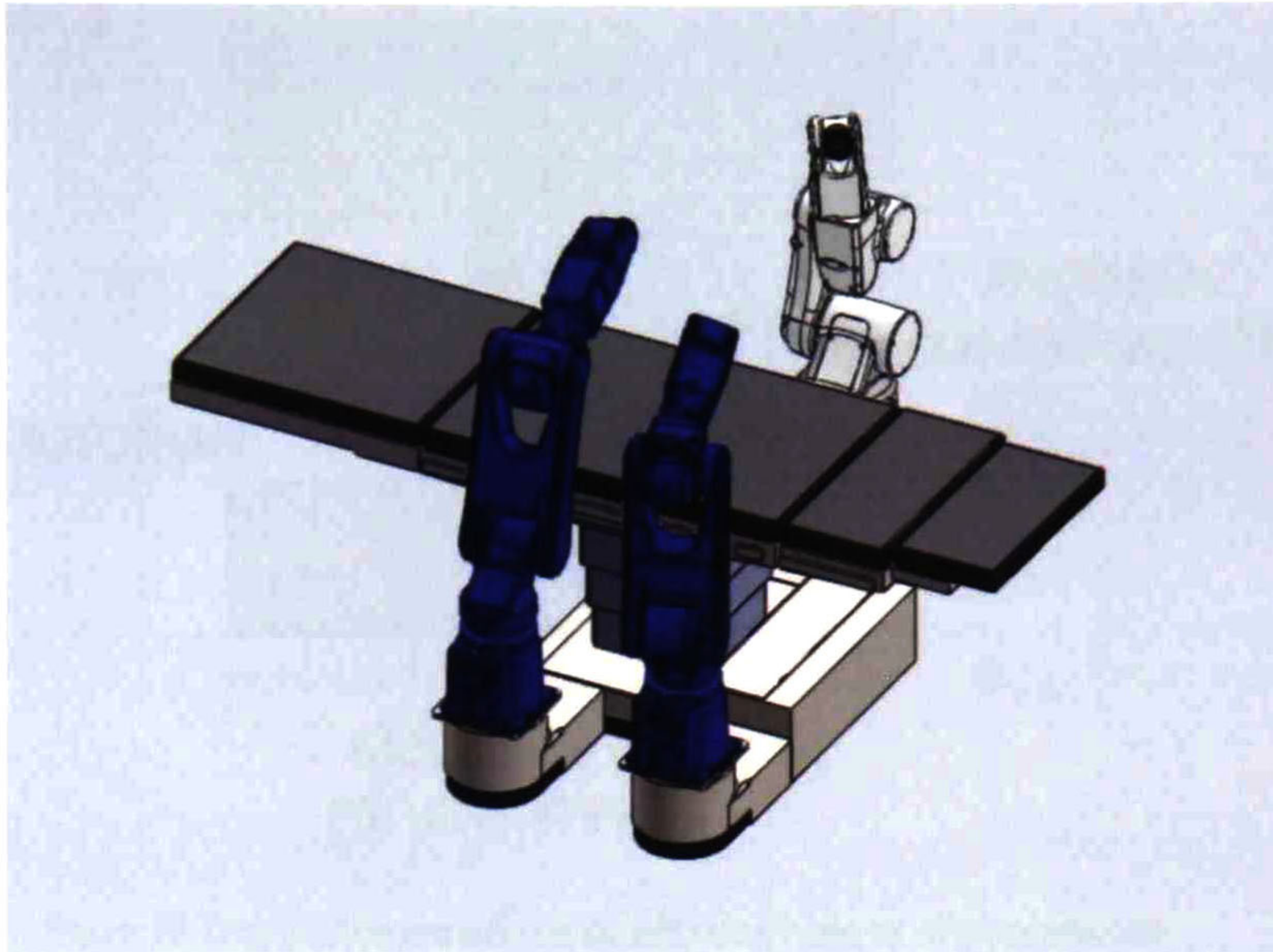


Figura 36. Arreglo de brazos robot propuesto para sistema de cirugía de mínima invasión

Como puede observarse se construyó una base que soporte la mesa quirúrgica, con bases individuales las cuales podrán elevar la altura de los brazos robot. La distribución espacial dentro del laboratorio se puede ver en la Figura 37.

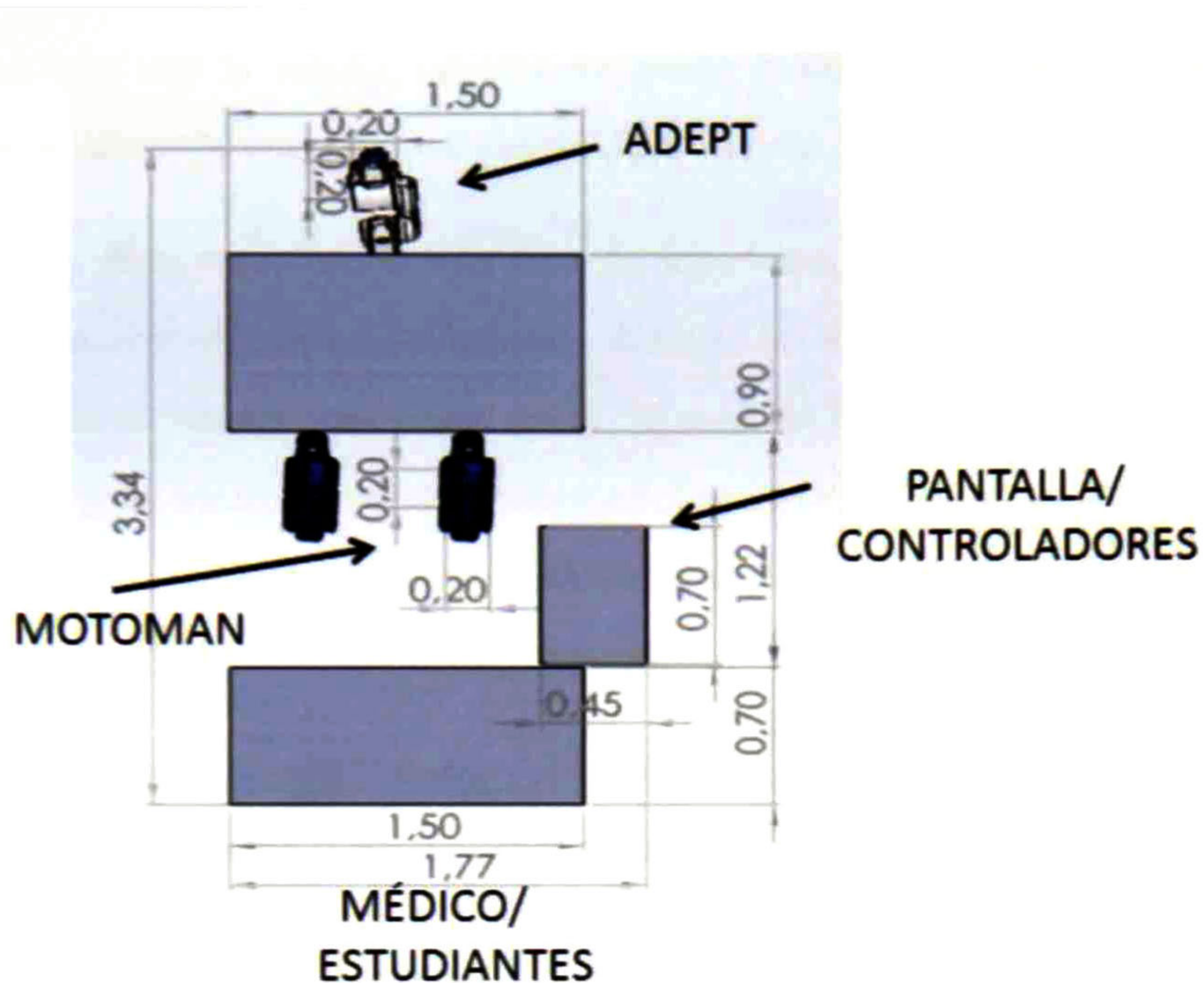


Figura 37. Distribución espacial de sistema de cirugía de mínima invasión

Como se puede observar, los elementos principales que componen el sistema son:

1. El sistema de 3 brazos robot junto a la mesa de cirugía.
2. Una consola donde el médico puede operar el sistema
3. Una pantalla donde el médico pueda observar el curso de sus acciones

## 4.2 Interfaces hápticas

Como ya se mencionó antes, las interfaces hápticas son capaces de transmitir sensaciones táctiles a las personas que los manipulan, pudiendo venir tales sensaciones de un ambiente virtual o de un ambiente real a través de una señal procesada proveniente de un sensor de fuerza.

También como se discutió en el capítulo 1, existe una gran variedad de este tipo de interfaces, para el caso de nuestro sistema nos interesan aquellas de tipo



joystick, ya que con la misma interfaz se puede manipular el brazo robot y al mismo tiempo sentir lo que el robot percibe, por lo cual hace una doble función.

Existe una gran variedad de este tipo de interfaces, desde las más económicas de 3 grados de libertad muy económicas, después las de 6 grados de libertad, los cuales asemejan a un pequeño brazo robot, lo cual las hace muy intuitivas para el cirujano, y después las de 7 grados de libertad, la mayoría de ellas ya especializadas en herramientas quirúrgicas (tijeras, pinzas, etc.), en el Anexo 7.1 mostramos varias de estas junto con un comparativo de las mismas.

Debido a requerimientos de costo y a los alcances del proyecto, se optó por una interfaz de 6 grados de libertad, la Omni haptic device de SENSABLE Inc. (Figura 38 ), se adquirieron 3 de estas y a la fecha se han hecho pruebas diversas de su funcionalidad. Durante el experimento de prueba del Modelo Neuronal Tacto-Visual se usó para interactuar con el objeto virtual reconstruido.



Figura 38. Omi haptic device, interfaz háptica de 6 grados de libertad de SENSABLE Inc.

Además de lo ya mencionado, se cuenta con dos endoscopio estéreo de 4 y 10 mm, de la empresa Richard Wolf Medical Corporation. La ventaja de tener un endoscopio estero sobre uno monocular es que el médico puede percibir ambientes

tridimensionales de la operación. También se cuenta con un casco el cual recibe ciertas órdenes fonéticas en inglés para manipulación de imágenes.

### **4.3 Escenario para prueba del modelo**

Para poder probar el modelo necesitamos 4 elementos:

1. Un phantom que representa un cuerpo y que tenga diferente rigidez a lo largo de su superficie.
2. Un brazo robot que tenga en la punta un sensor de fuerza y que servirá para medir la rigidez en algunos puntos superficiales del phantom.
3. Una cámara RGB-D que abstraiga una nube de puntos de la superficie del objeto.
4. Una interfaz háptica que nos permita interactuar de manera virtual con el objeto reconstruido.

Para este escenario se tomaron los elementos descritos en las secciones 4.1 y 4.2 de este capítulo. Como se muestra en la Figura 39, para el experimento se usaron 2 robots MOTOMAN (Anexo 7.2, página 96). Uno de los robots es el encargado de sostener la cámara RGB-D (Anexo 1.3, página 97) mientras que el otro tiene un sensor de fuerza (Anexo 1.4, página 97) en el efector final; para este sensor se construyó un extensor que a la vez protege el sensor y facilita la toma de muestras de rigidez superficial.

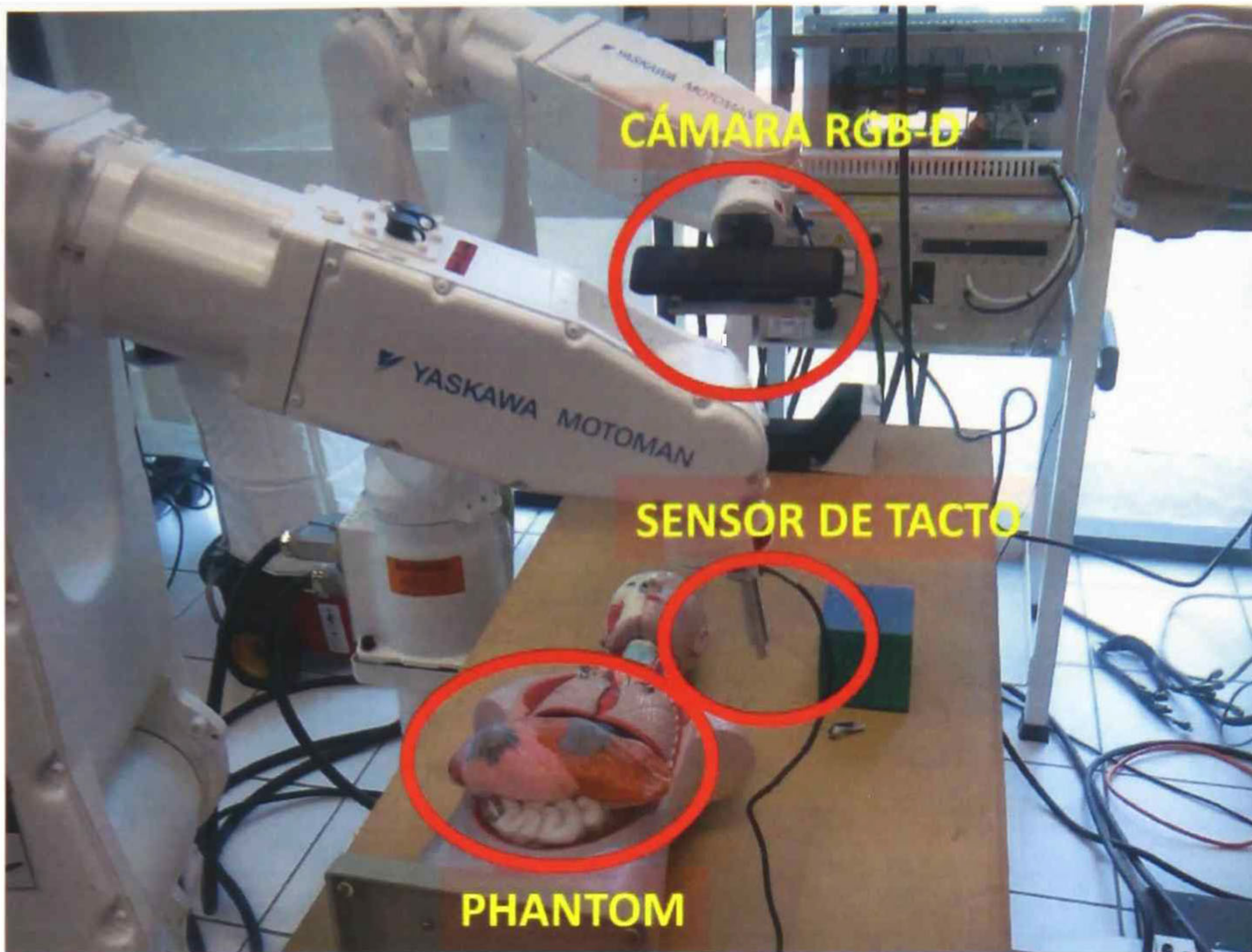


Figura 39. Esquema usado para el experimento

Se construyó un phantom usando plastilina (de diferentes texturas y colores) emulando la forma de un hígado, al cual se le hicieron dos protuberancias emulando dos "tumoraciones". Al centro de estas se le colocó una piedra y alrededor de ellas pegamento endurecido, emulando diferentes grados de rigidez en el área de la protuberancia. La mayor parte de la plastilina usada fue plastilina muy blanda, excepto en una zona donde se usó plastilina endurecida como se muestra en la Figura 40.

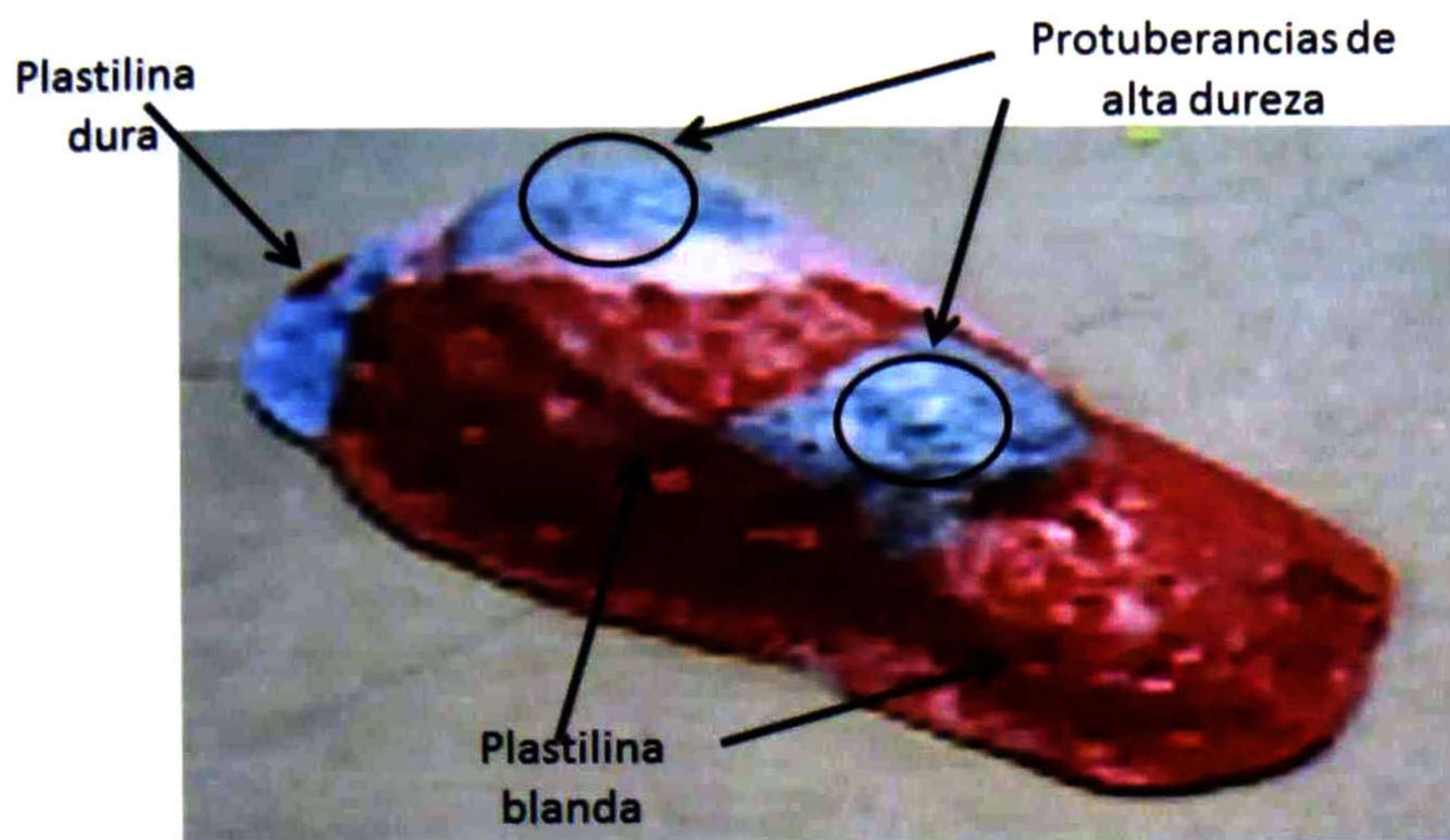


Figura 40. Phantom construido para experimento

## 5. Resultados Experimentales y Simulaciones

En este capítulo presentamos simulaciones y aplicaciones hechas con las redes propuestas en el capítulo anterior. La red esférica de base radial fue implementada en Matlab mientras que el modelo neuronal tacto-visual fue implementado de manera híbrida usando MATLAB y CLUCalc [24]. Se detalla un experimento de laboratorio hecho con una cámara RGB-D, un phantom, un sensor de tacto y el sistema descrito en el capítulo anterior para mostrar el potencial neuronal tacto-visual.

### 5.1 Redes Generales Esféricas para Regresión.

Ahora vamos a enfocarnos en presentar algunos resultados y aplicaciones de las SGRNN planteadas en la sección 3.4. Primeramente presentamos una interpolación de una serie de puntos, esta interpolación se hace usando la SGRNN y también la GRNN convencional, el resultado se puede ver en la Figura 41. Como podemos ver las interpolaciones hechas por una u otra función podrían ser muy diferentes.

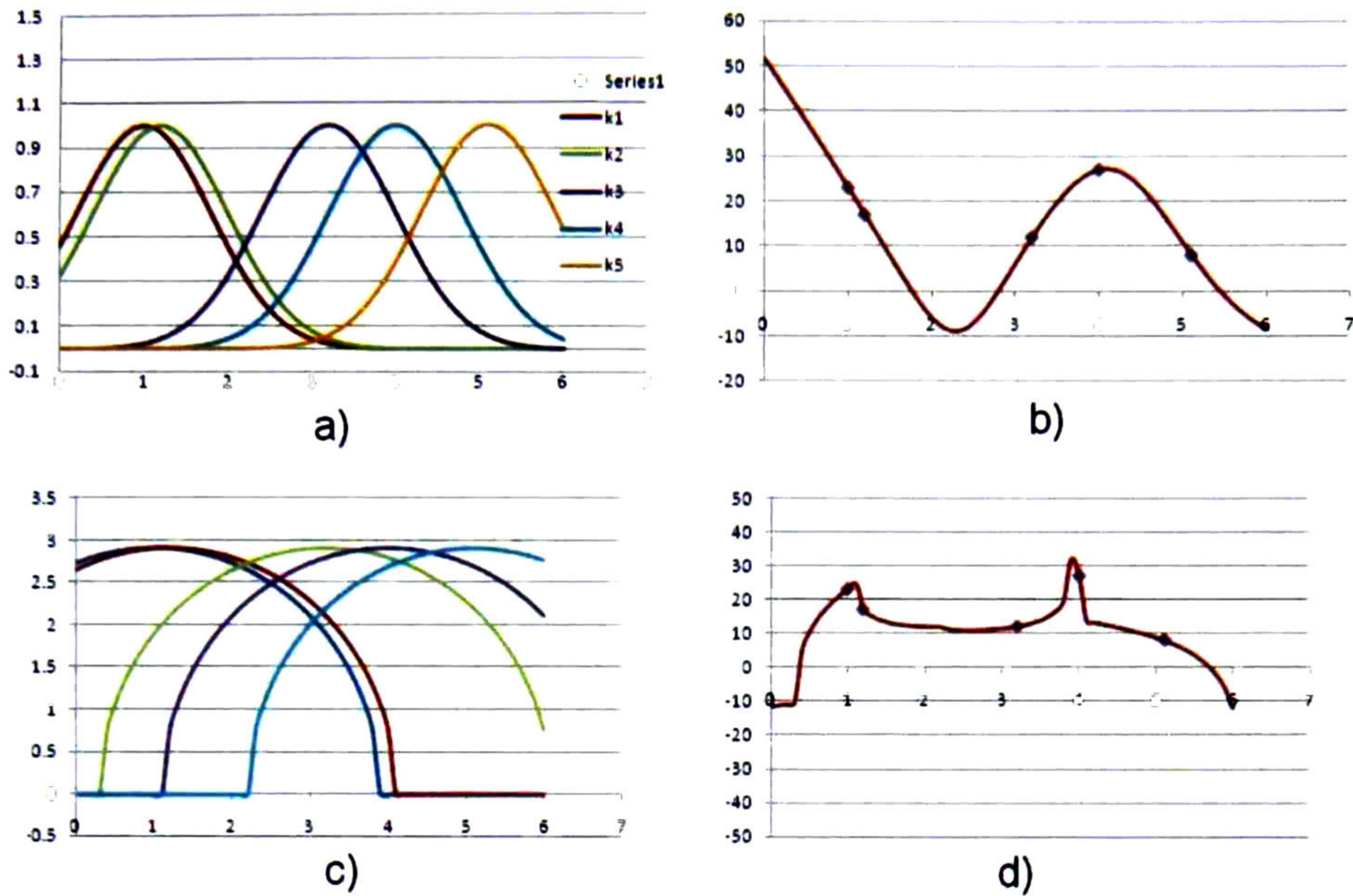


Figura 41. Conjunto de puntos interpolados con la GRNN (arriba) y con la SGRNN (abajo)

Para darnos una idea del desempeño de ambas redes (la GRNN y la SGRNN), vamos a tomar un polinomio en el dominio  $[0,4.9]$ :

$$\begin{aligned}
 P(x) = & 0.004135767x^{10} - 0.11187453x^9 + 1.2962494x^8 \\
 & - 8.3709208x^7 + 32.794892x^6 - 79.344987x^5 + 115.03922x^4 \\
 & - 91.038203x^3 + 30.425164x^2 - 0.068383666x + 0.28950418
 \end{aligned} \tag{5.1.1}$$

este polinomio se muestra en la Figura 42.

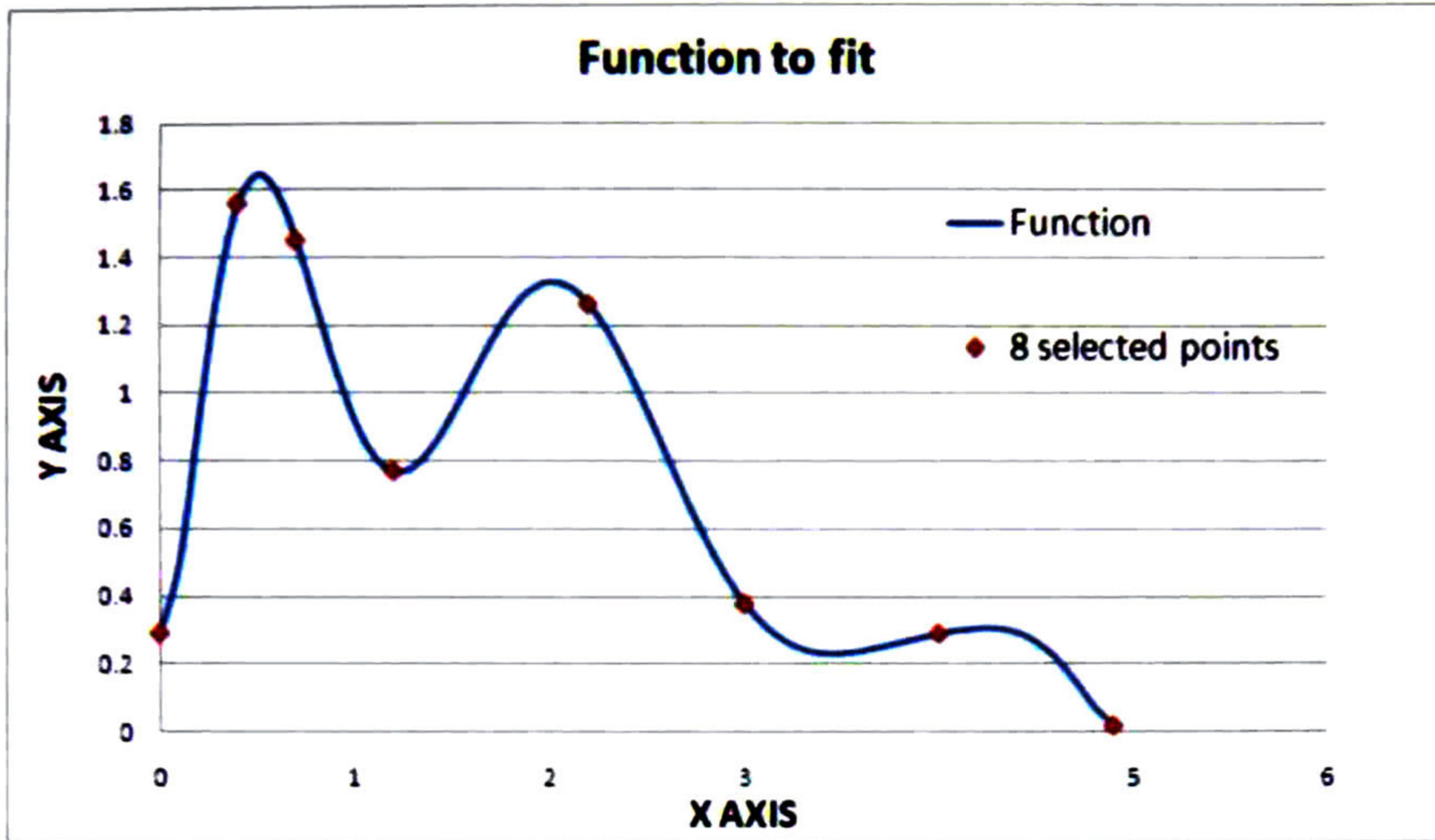


Figura 42. Polinomio propuesto para medir desempeño de la GRNN vs. la SGRN

Usando tal polinomio proponemos el siguiente experimento, realizar la interpolación de la función usando un muestreo uniforme de 8 puntos y después de 12 puntos, esta interpolación se realiza con ambas redes, los resultados se muestran en la Figura 43.

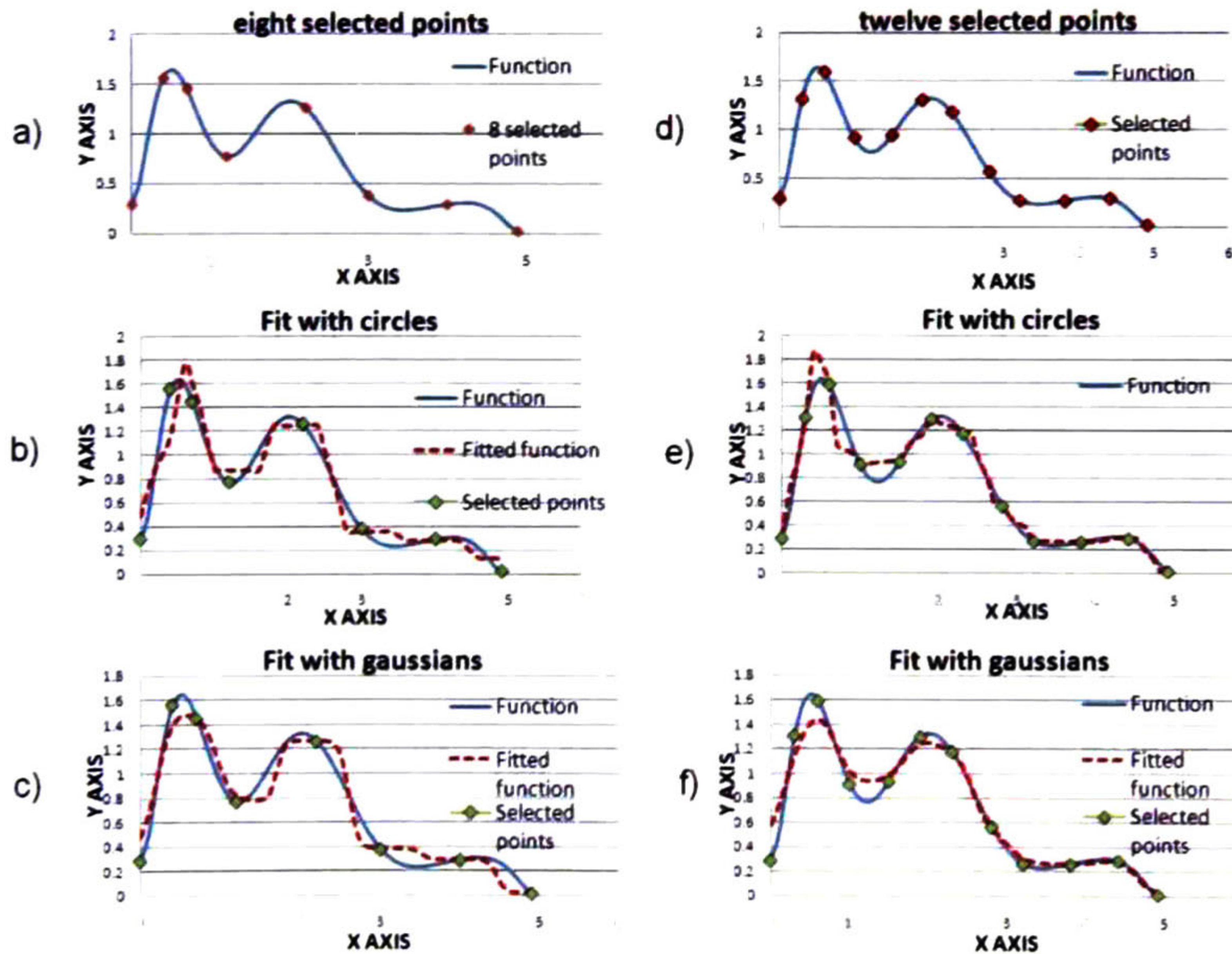
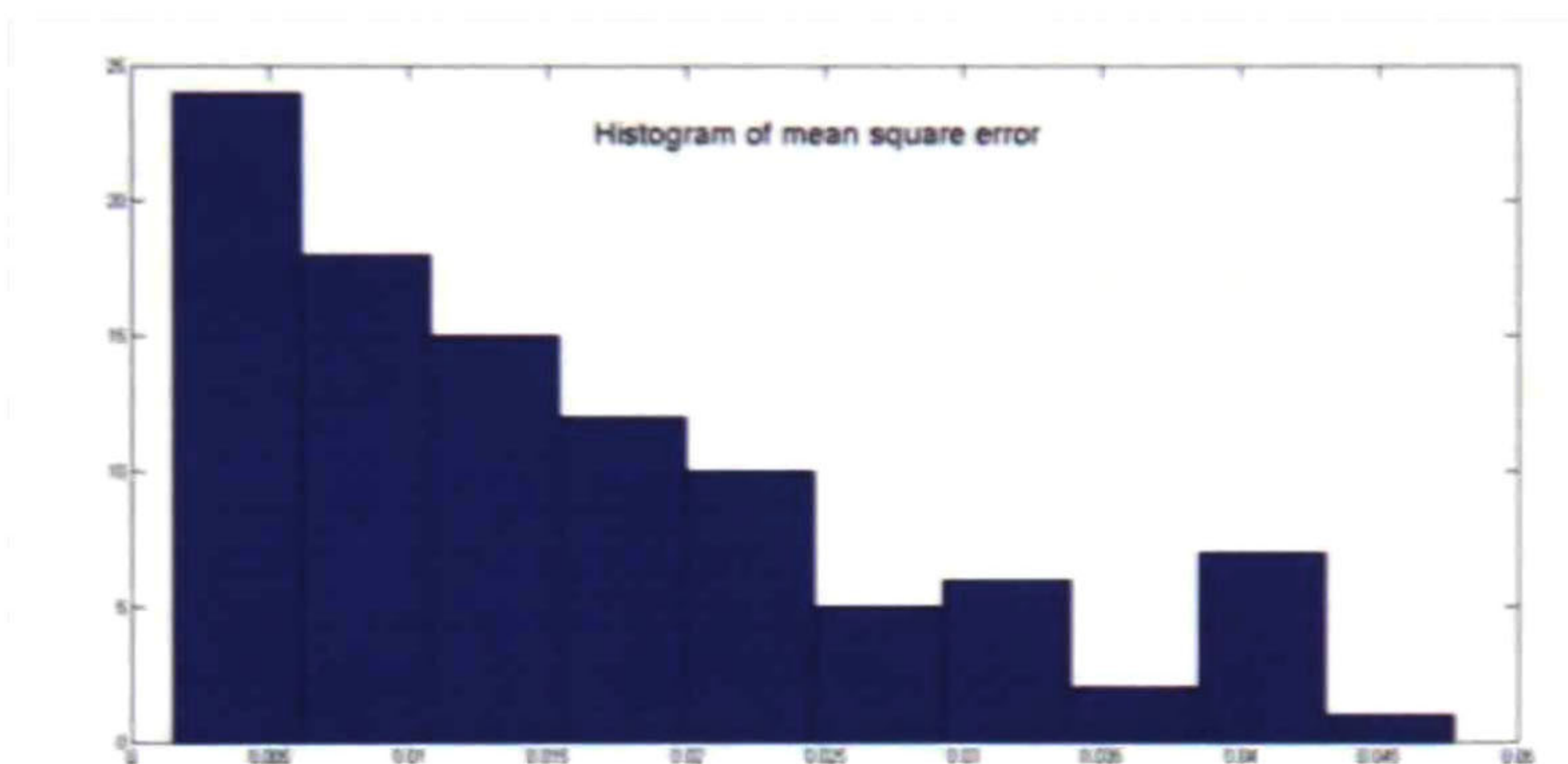
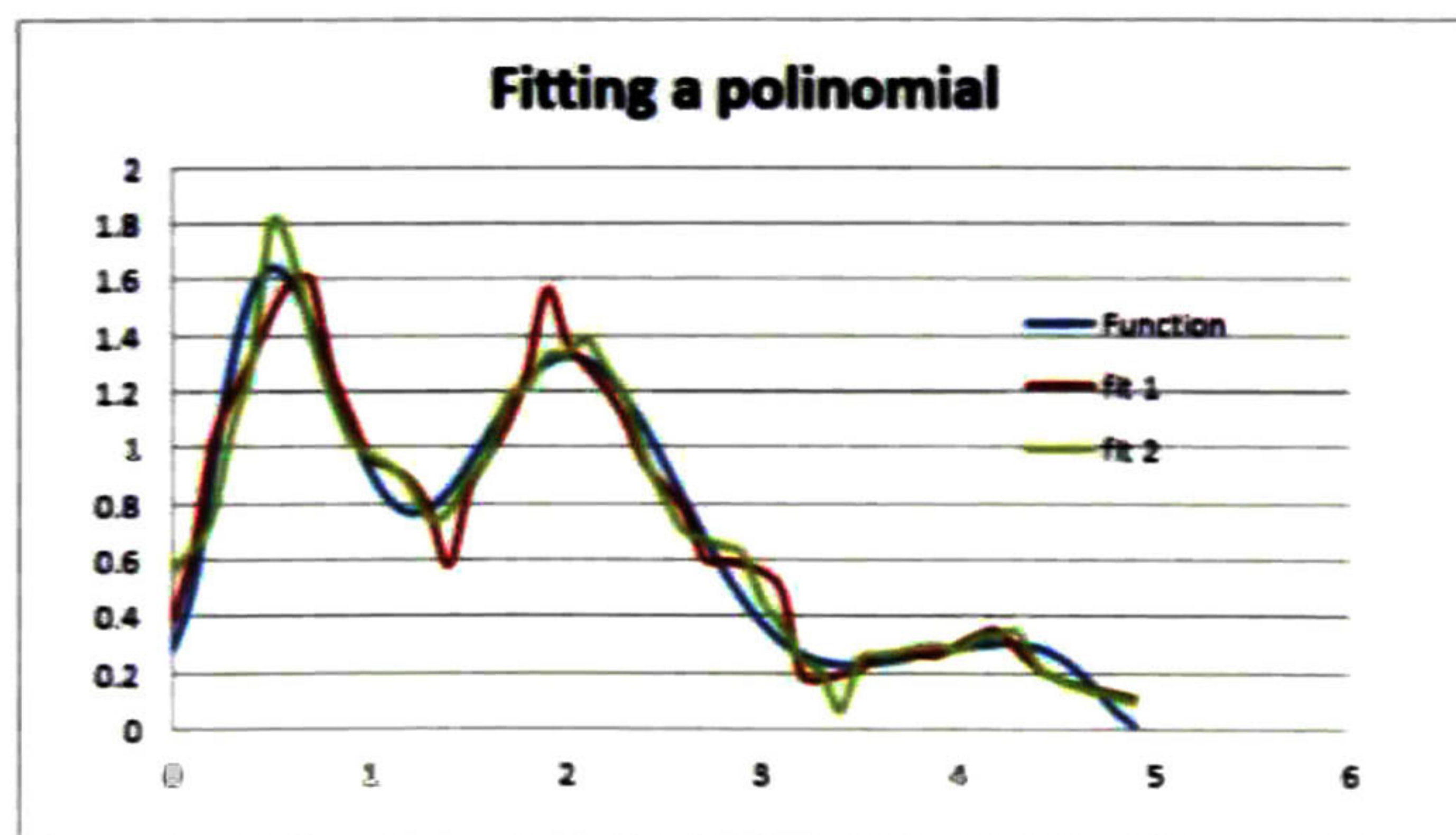


Figura 43. Se muestra a la izquierda la interpolación de un polinomio usando un muestreo uniforme de 8 puntos, con la GRNN y con la SGRN, a la derecha lo mismo pero usando un muestreo uniforme de 12 puntos

Como se puede ver, el desempeño de ambas redes es similar en ambos casos. Otro experimento que mostramos en este trabajo es el de realizar la interpolación usando la SGRN pero con puntos tomados aleatoriamente (en contraste con el experimento anterior donde el muestreo es a intervalos constantes); esto nos da una idea de la repetitividad de la interpolación. La cantidad de puntos aleatorios tomados en cada experimento es 20, y el experimento se repitió 100 veces, los resultados se muestran en la Figura 44.



(a)



(b)

Figura 44. Se muestra en (a) el histograma de MSE al interpolar con 20 puntos seleccionados de forma aleatoria 100 veces, en (b) 2 ejemplos de interpolación tomados del conjunto de 100.



En la Figura 44 (a) se muestra un histograma del comportamiento del MSE para los 100 experimentos realizados, con ello podemos ver que al incrementarse el número de puntos muestra, la red realmente tiende a converger a la curva que se está ajustando; en (b) 2 ejemplos de interpolación tomados de los 100 experimentos.

Otro cuestionamiento que queremos discutir es acerca del valor de parámetro  $b$  en la ecuación (3.0.3), que como ya se mencionó en la sección 3.4 representa el radio de los círculos para la interpolación (en el caso de la GRNN representa la desviación estándar de la Gaussiana), y que está asociado con la suavidad de la interpolación. De acuerdo a la ecuación(3.1.1) este tiene un valor mínimo aceptable, entonces podemos realizar la interpolación nuevamente del polinomio de la ecuación (5.1.1) para distintos valores de  $b$ . En la Figura 45 presentamos el comportamiento del MSE al variar el parámetro  $b$  en el intervalo  $[0.5d_{max}, 0.8d_{max}]$ ; en esta se puede ver que el variar el radio de los círculos no impacta de manera tan significativa el desempeño de la red al menos en este rango, este y otros experimentos similares nos permiten afirmar que para interpolar polinomios, una buena heurística es seleccionar

$$b = 0.7d_{max} . \quad (5.1.2)$$

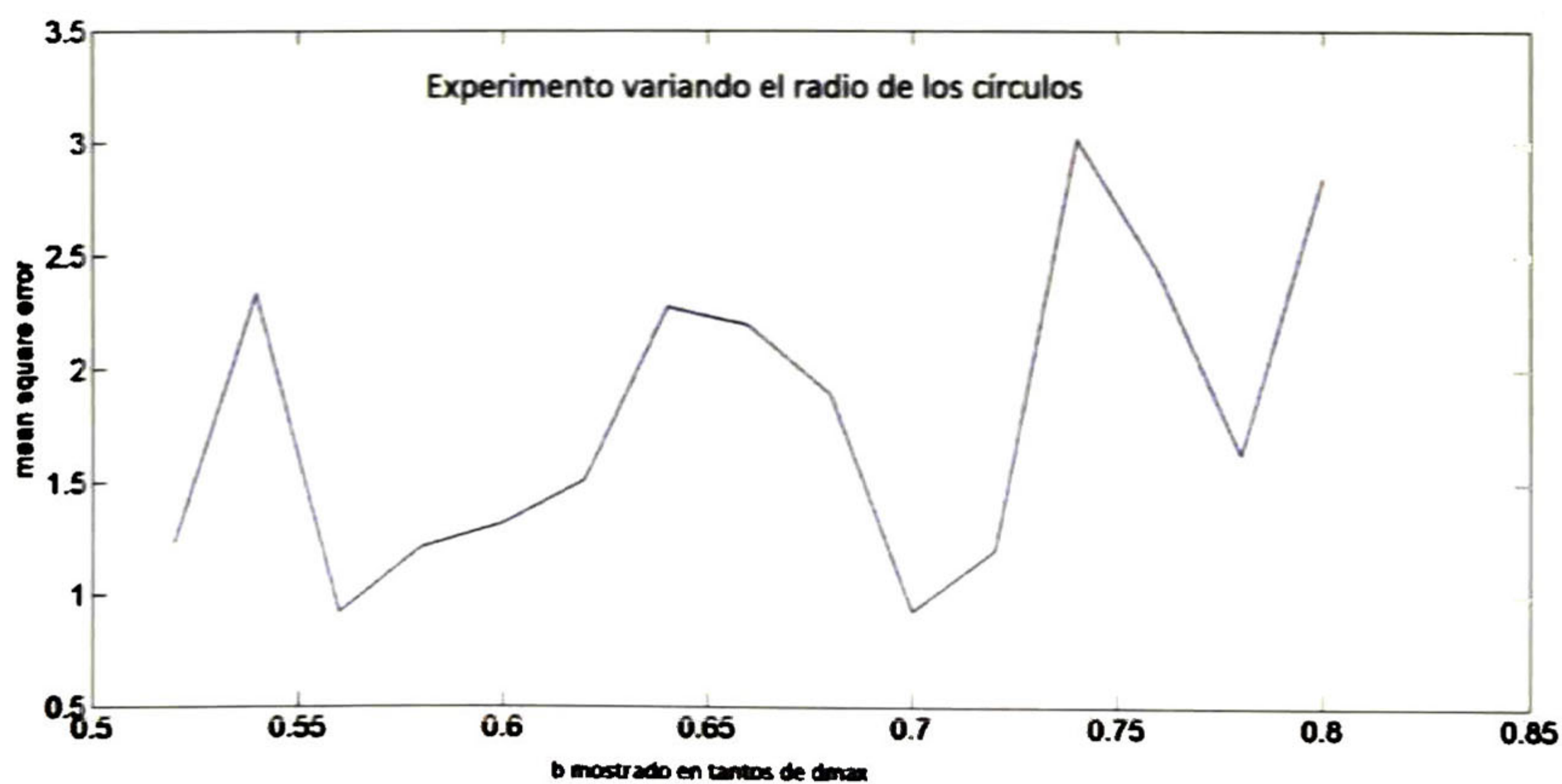


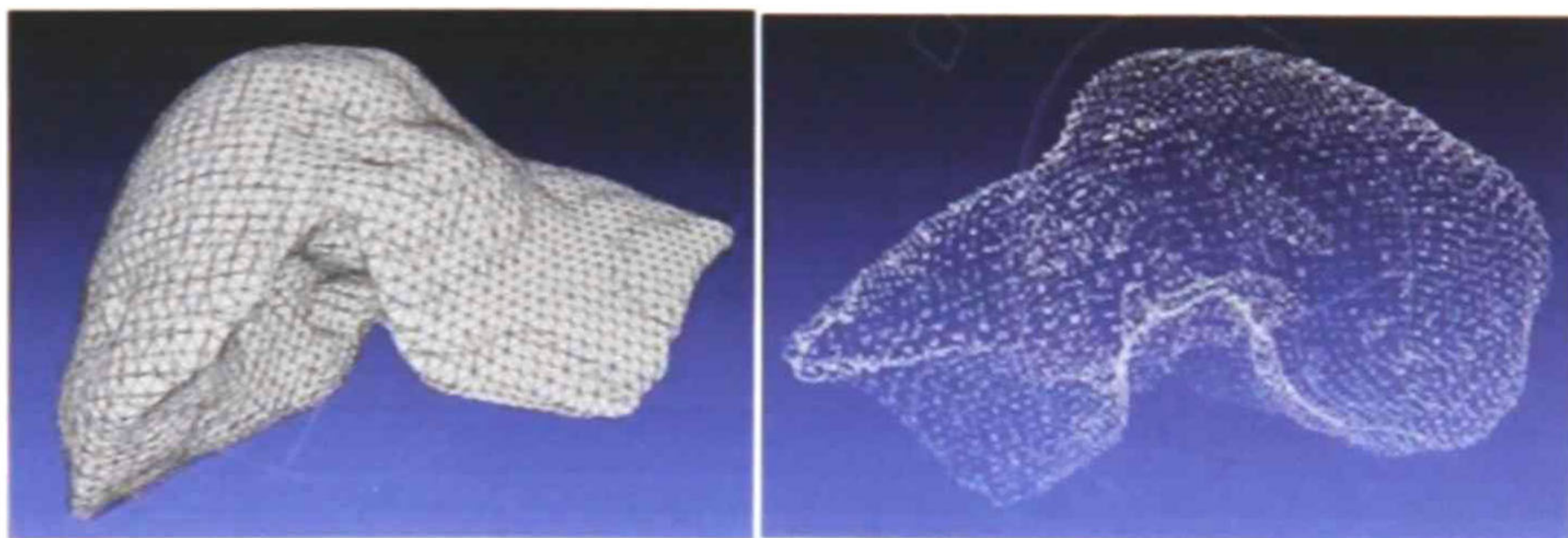
Figura 45. Error cuadrático Medio (MSE) al variar el radio de los círculos en la interpolación

Ahora vamos a presentar una aplicación de la SGRN, se van a reconstruir cortes visuales de un hígado, usando el modelo 3D tomado del proyecto Visible Human de la National Library of Medicine [24]. En esta base de datos se cuenta con un cuerpo humano completo, escaneado usando tomografía computarizada, resonancia magnética y cortes físicos seccionales. El objetivo del proyecto es la creación de una representación tridimensional, completa y anatómicamente detallada, de los cuerpos de un hombre y una mujer normales. En la Figura 46 se muestra una nube de puntos de un hígado tomado de la librería digital mencionada.

Para poder visualizar mejores los resultados, se corta el hígado en 2 partes, la parte alta y la parte baja, esto se puede hacer definiendo un plano que pase aproximadamente por el centro, de manera visual seleccionamos los 3 puntos  $p_1(43, 51, 88)$ ,  $p_2(220, 56, 117)$  y  $p_3(103, 200, 158)$ ; y usamos al ecuación del plano  $\pi_x$ , que en OPNS se calcula como

$$\pi_x = p_1 \wedge p_2 \wedge p_3 \wedge e_\infty, \quad (5.1.3)$$

Se corta el hígado usando dicho plano, las 2 secciones resultantes del corte se muestran en la Figura 47



(a)

(b)

Figura 46. (a) Malla de un hígado humano, (b) nube de puntos del hígado.

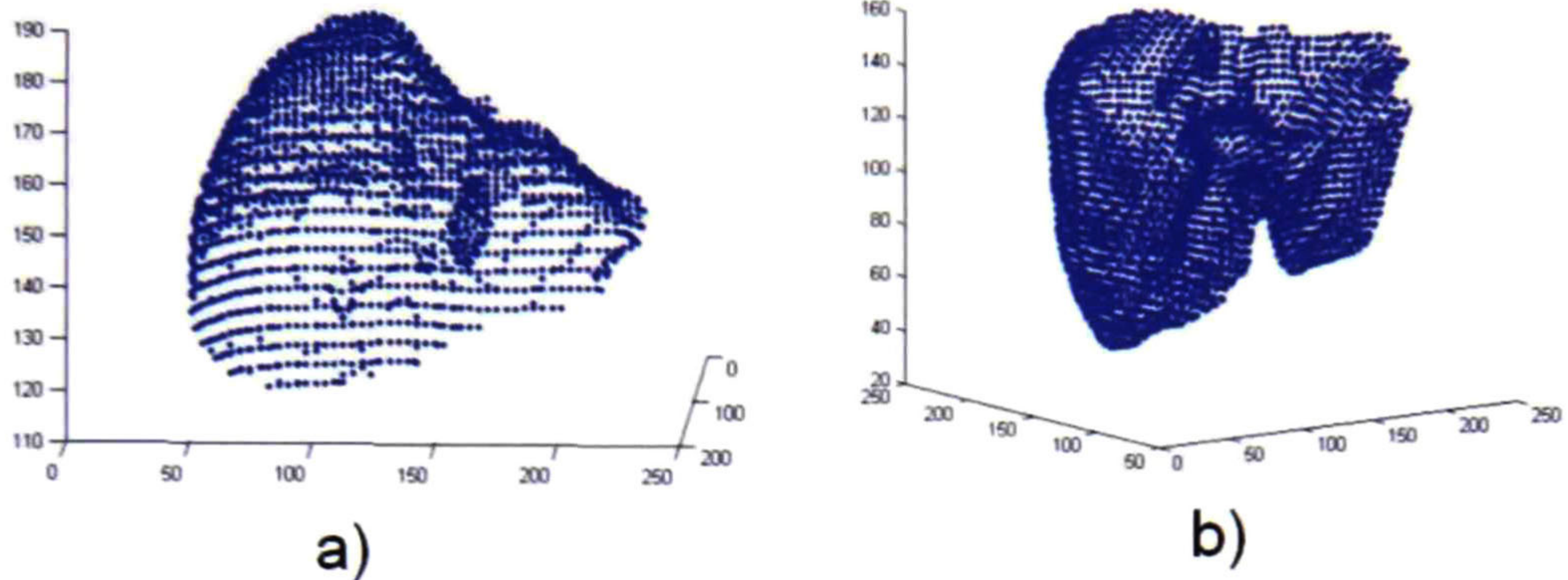


Figura 47. Nubes de puntos del hígado.

Ahora se toma una sección de la parte alta del hígado y haremos una reconstrucción de este corte usando la SGRN. Se llamará corte sagital a un corte paralelo al eje x y corte axial a uno paralelo al eje y, como se muestra en la Figura 48. Cortes sagitales y axiales del hígado.

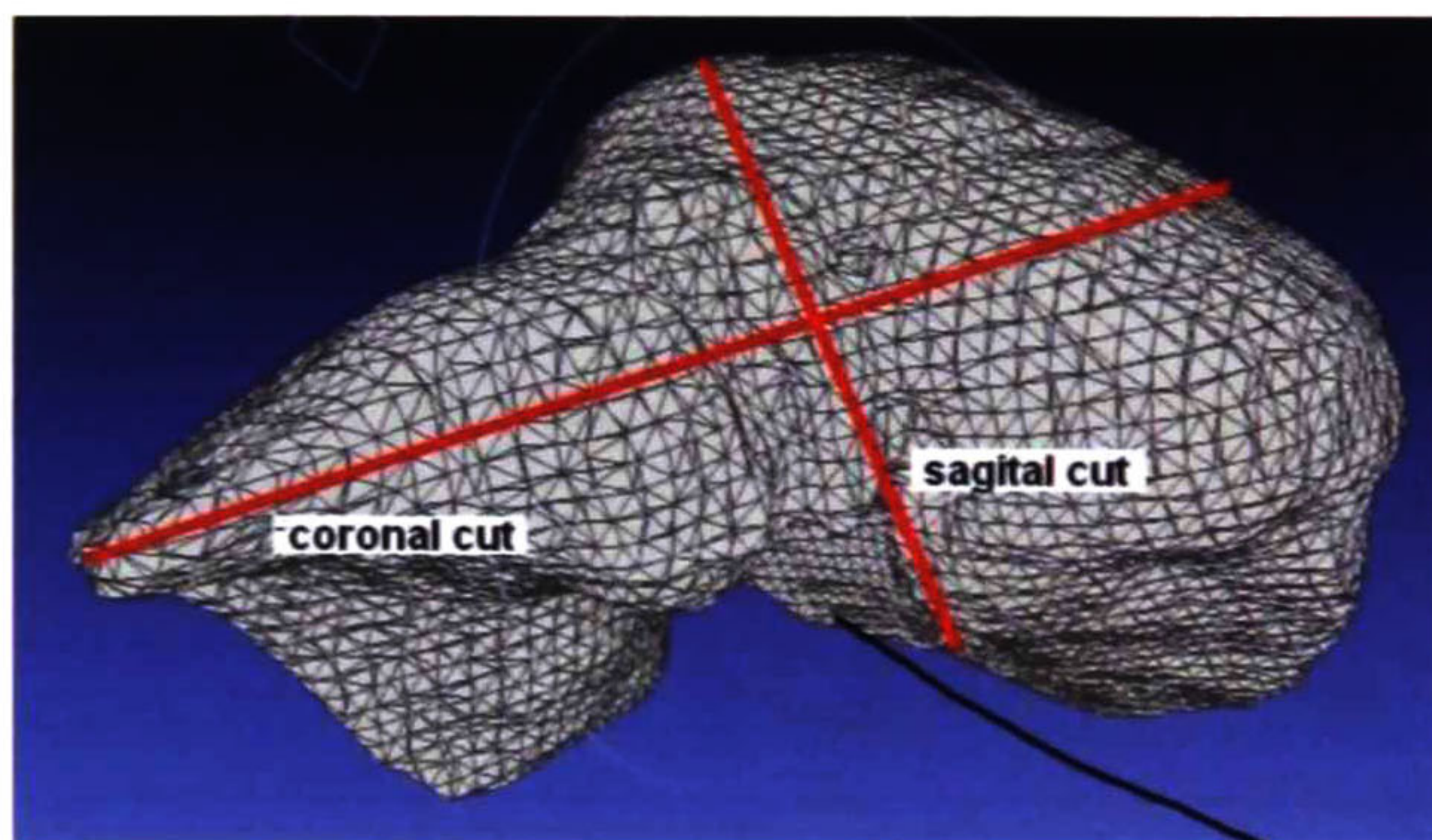


Figura 48. Cortes sagitales y axiales del hígado

Se toma un corte sagital y uno axial, y tomamos 10 puntos a lo largo del cada uno de ellos, y se usa la SGRN para reconstruir el resto del corte, los resultados se muestran en la Figura 49.

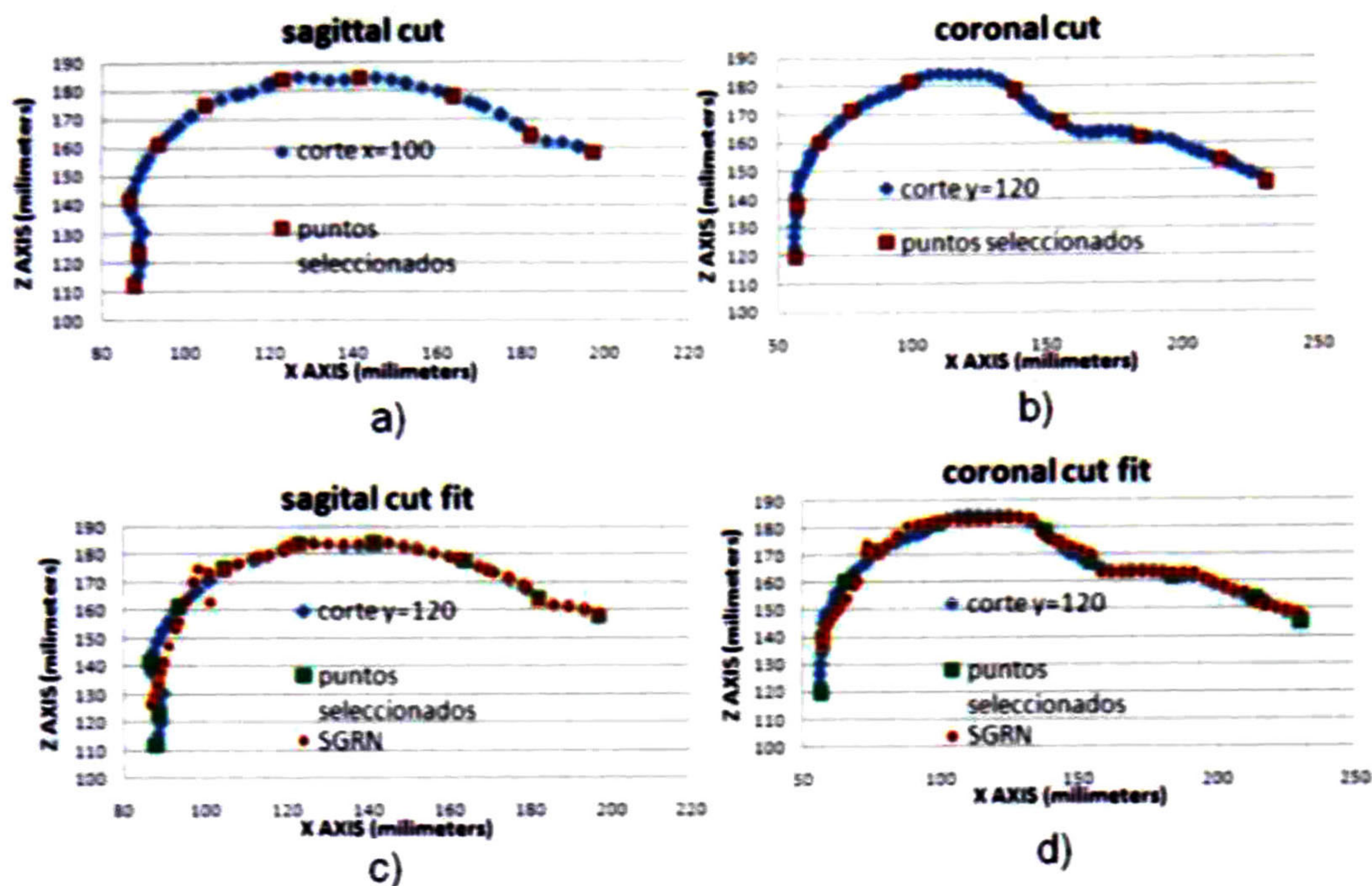


Figura 49. Se muestra un corte sagital (a) y un corte axial (b); también se muestra en (c) y (d) la reconstrucción usando la SGRN de los cortes sagital y coronal respectivamente (los puntos rojos representan los obtenidos con la SGRN).

## 5.2 Implementación del Modelo Neuronal Tacto-Visual

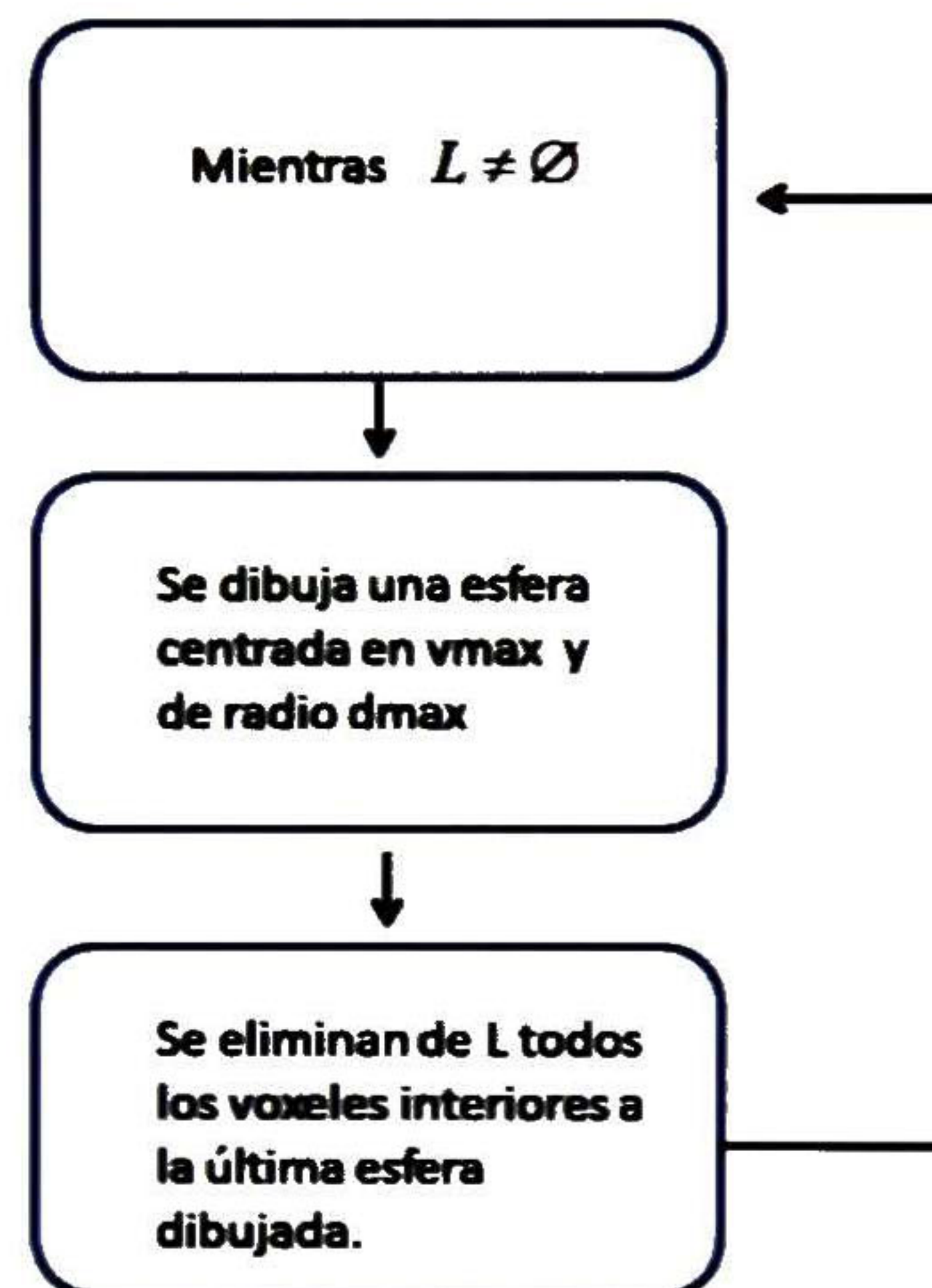
El Modelo Neuronal Tacto-Visual presentado en la sección 3.3 fue implementado usando 2 lenguajes de programación CLUCalc [24] y MATLAB 2011.

La primera parte del modelo correspondiente a la recepción de la nube de puntos y la implementación del *sphere packing algorithm* fue implementado usando el Álgebra Geométrica Conformal en el lenguaje CLUCalc. El algoritmo implementado se describe a continuación:

1. Se realiza un voxelizado tridimensional, seleccionando un valor para el lado de cada voxel como delta. De la selección de este valor dependerá la

precisión del algoritmo, para valores de delta pequeños el llenado de esferas será más preciso en la superficie pero requiriendo más tiempo de cómputo.

2. Se hace una lista  $L$  con las coordenadas de todos los voxeles que tienen puntos de la nube en su interior o superficie.
3. Para cada voxel en  $L$ , se calcula su distancia más corta a la superficie, de tal modo que para cada voxel en  $L$  se le asocia un valor  $d$ . Al valor más grande  $d$  de la lista lo llamamos  $d_{max}$  y  $v_{max}$  a las coordenadas del voxel que lo contiene.
4. Se aplica la siguiente función de manera recursiva:



El código de este algoritmo en el anexo 1.5 de esta tesis. Es mucho más simple observar el poder de este algoritmo en 2D, por lo que se hizo una implementación de este tipo. En la Figura 50 podemos ver la reconstrucción de un objeto plano usando círculos. Observe como los círculos interiores contienen una gran cantidad de puntos de la nube, mientras que los círculos exteriores marcan el contorno de la figura.

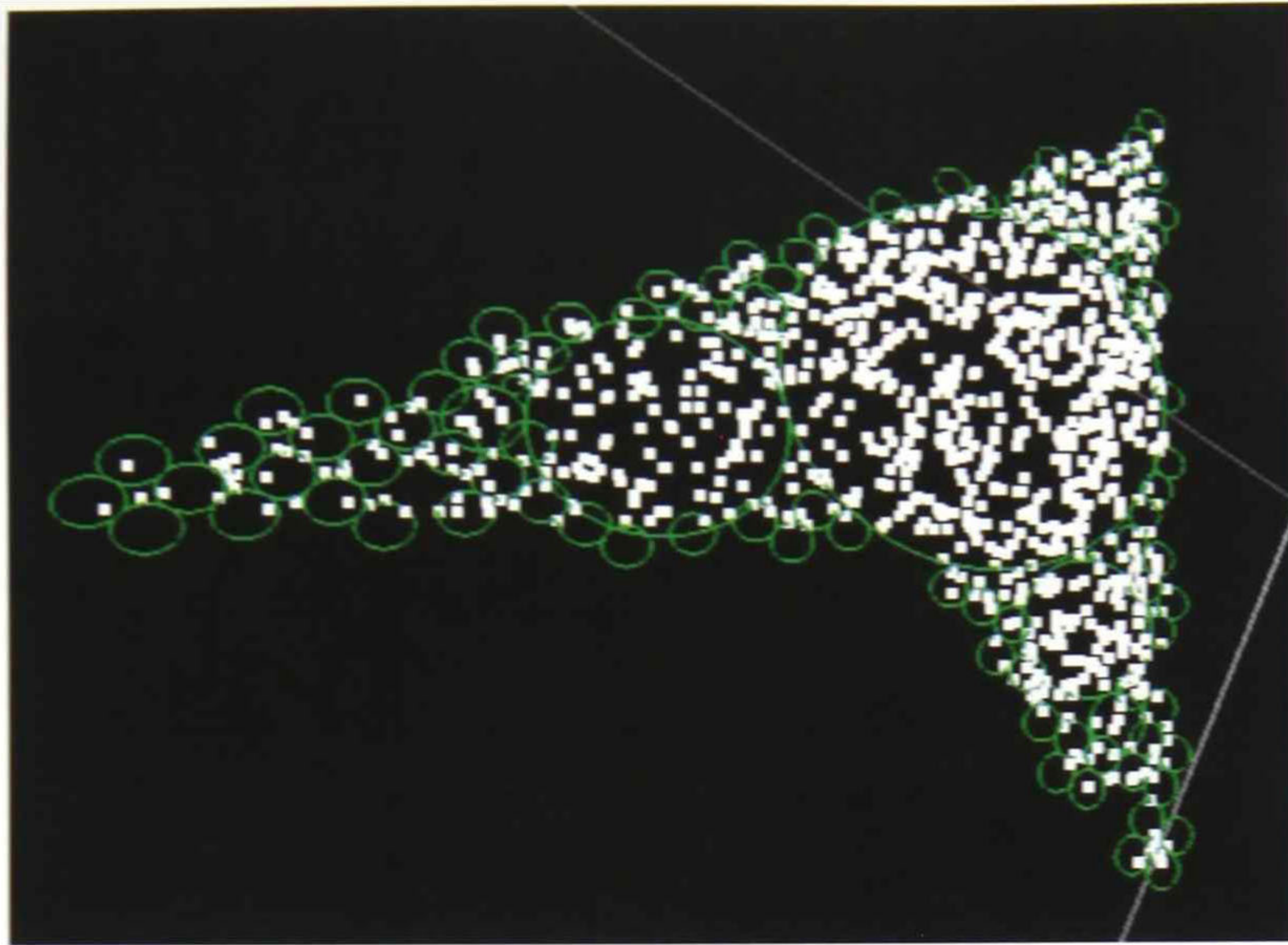


Figura 50. Reconstrucción de un objeto 2D usando círculos.

Ahora vamos a observar algunas reconstrucciones de objetos tridimensionales. Primeramente se muestra en la Figura 51 dos vistas de la reconstrucción de un objeto. Es fácil observar que en el interior quedan contenidas las esferas de radios grandes mientras que las esferas de radios menores se encuentran representando la superficie.

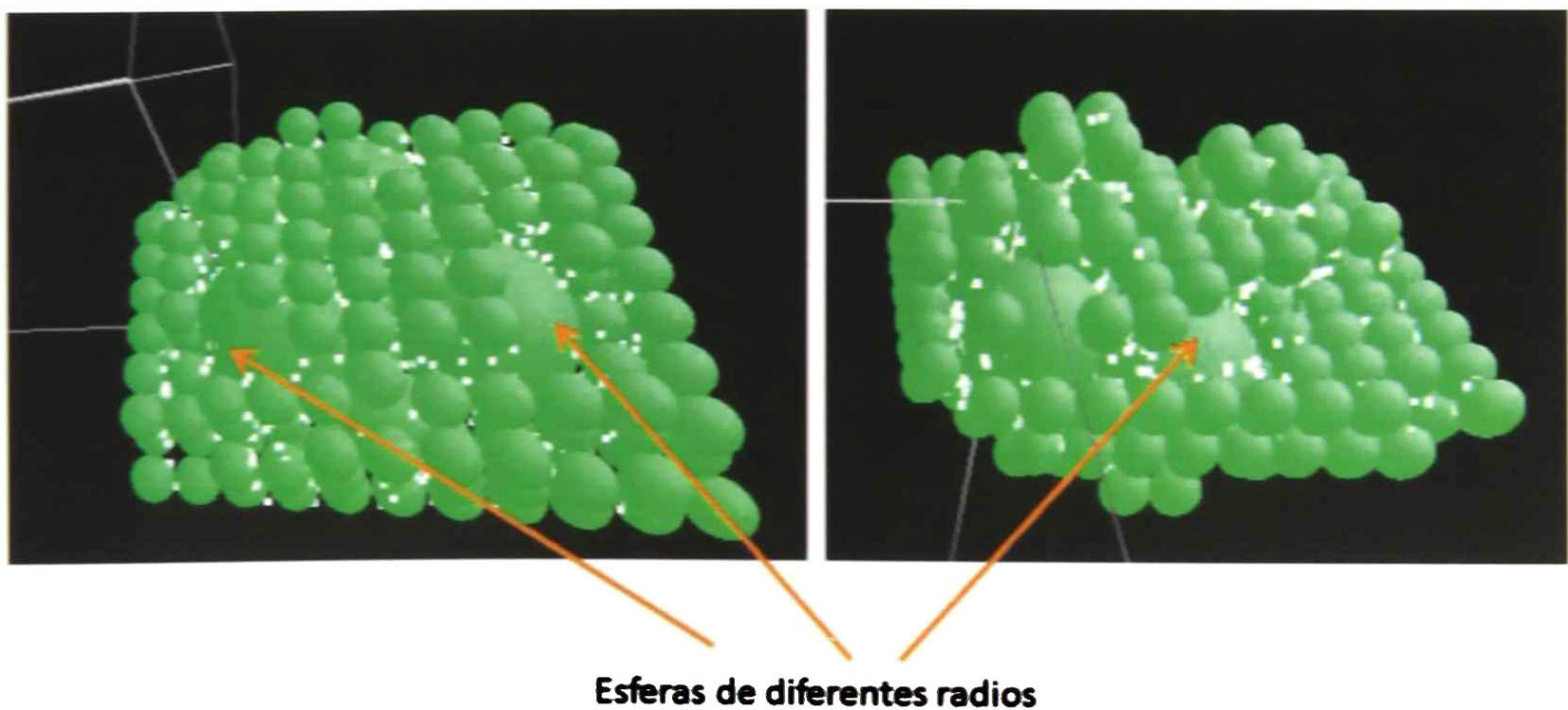


Figura 51. Reconstrucción de un objeto tridimensional. Se pueden observar esferas de diferentes radios al centro del objeto

Otro aspecto que vamos a resaltar de este algoritmo es acerca de cómo afecta el valor delta (primer paso del *sphere packing algorithm*) a las reconstrucciones, recordemos que de este valor depende la exactitud de la reconstrucción así como el tiempo de cómputo de la misma, para valores de delta pequeños tenemos una mayor precisión en la reconstrucción (sobre todo en los bordes) pero con un alto costo en cuanto al tiempo de cómputo. En la Figura 52 se muestran la reconstrucción del mismo objeto para diferentes valores de delta.

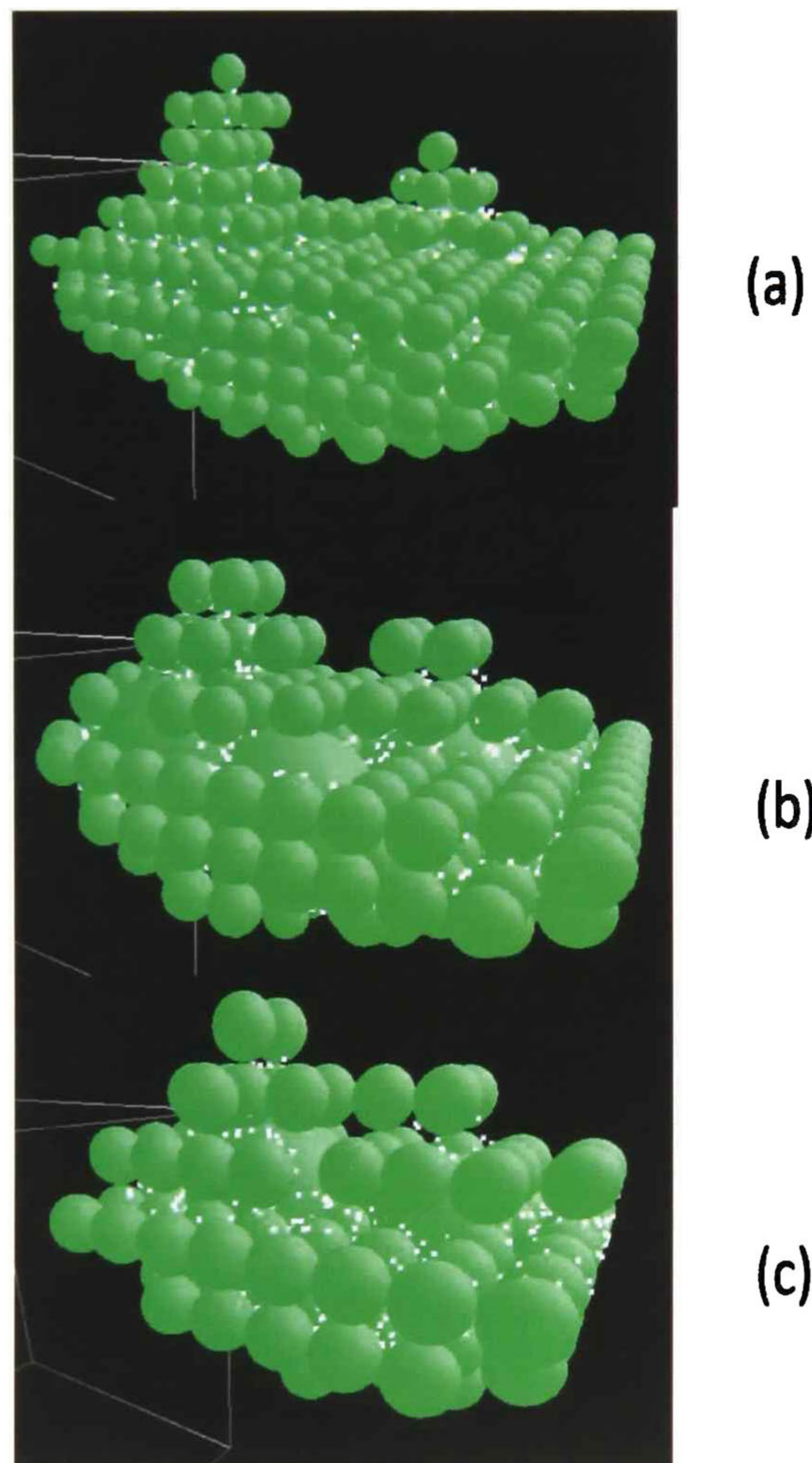


Figura 52. Se muestra la reconstrucción de un mismo objeto pero con diferentes valores de delta, en a) 0.2, en b) 0.3 y en c) 0.4

El segundo paso más que código se debe realizar de manera experimental, en la siguiente sección se detalla un experimento realizado en el laboratorio. El tercer y cuarto paso del método, que corresponde al entrenamiento de la red neuronal y luego al cálculo de la rigidez en cada esfera con la misma red se realizaron usando el toolbox NN (Neural Networks) de MATLAB que tiene la GRNN pre programada.

### 5.3 Prueba en laboratorio del Modelo Neuronal Tacto-Visual.

En esta sección se detalla un experimento realizado para probar el modelo Neuronal Tacto Visual. El objetivo del experimento fue el de obtener una representación virtual de un objeto además de poder representar simultáneamente la rigidez sobre su superficie. Para este experimento se usó el sistema para cirugía de mínima invasión detallado en la sección 4.3 de este documento; la forma como se realizó dicho experimento es la siguiente:

1. Se obtiene una nube de puntos (escaneo 3D) usando la cámara RGB-D fija, cuidando en centrar el phantom aproximadamente en la imagen.
2. Se tocan algunos puntos sobre la mesa donde se encuentra el phantom y se registran sus coordenadas, es de especial interés un punto central sobre el cual se colocará el phantom. En la imagen bidimensional de la cámara es posible detectar por un humano el punto donde el sensor de tacto realiza el contacto. de esta forma se almacena la coordenada (x,y,z) sobre el plano donde se realiza el contacto en el marco de referencia de la cámara RGB-D.
3. Con el sensor de tacto fijado en el efector final de un robot, se toca una serie de puntos sobre el phantom, lo cual nos permite obtener la rigidez del cuerpo en diferentes puntos sobre su superficie. En este punto realizamos la siguiente observación. de acuerdo a nuestro modelo la rigidez superficial de un cuerpo k se calcula de acuerdo a la fórmula:

$$k = \vec{F} / \vec{d}$$



ya discutida en la sección 3.3. Recordando que  $\vec{d}$  es el desplazamiento al aplicar de manera ortogonal a la superficie una fuerza  $\vec{F}$ . entonces si a lo largo del experimento el valor de  $\vec{d} = \text{constante}$ , podemos considerar que  $k \sim |\vec{F}|$  (el valor del escalar  $k$  es directamente proporcional a la magnitud de la fuerza aplicada). Esto se logra poniendo una marca en la punta del extensor del sensor que hace en la práctica una buena aproximación de esto, lo cual tomaremos como un supuesto. Las mediciones de la fuerza a lo largo del experimento se hicieron en libras.

4. Cada vez que se realiza un contacto además de registrar la fuerza obtenida con el sensor se almacena la coordenada  $(x,y,z)$  del punto de contacto. Igual que en el paso 2, el punto en cuestión se encuentra expresado en el marco de referencia de la cámara RGB-D

## Procesamiento de los datos

El procesamiento de los datos obtenidos se realiza de la siguiente forma:

1. Se tiene una gran nube de puntos ya que la cámara RGB-D tiene una resolución de 640x480 pixeles, lo cual nos da un total de 307,200 puntos en el espacio. De estos solo nos interesan los puntos que contienen al phantom, entonces se detecta en la imagen 2D el punto central (obtenido en el segundo paso del experimento) sobre el cual se coloca el phantom y se toma una vecindad que garantice contener al phantom completamente. El resultado de esto se puede observar en la Figura 53.
2. En la es claro notar un hueco en la medición del sensor, esto es debido a la forma del phantom y a que solo se tiene una perspectiva (esto se puede salvar fusionando varias vistas del sensor). Se realiza un filtrado de los pixeles correspondientes a este hueco, los resultados se pueden

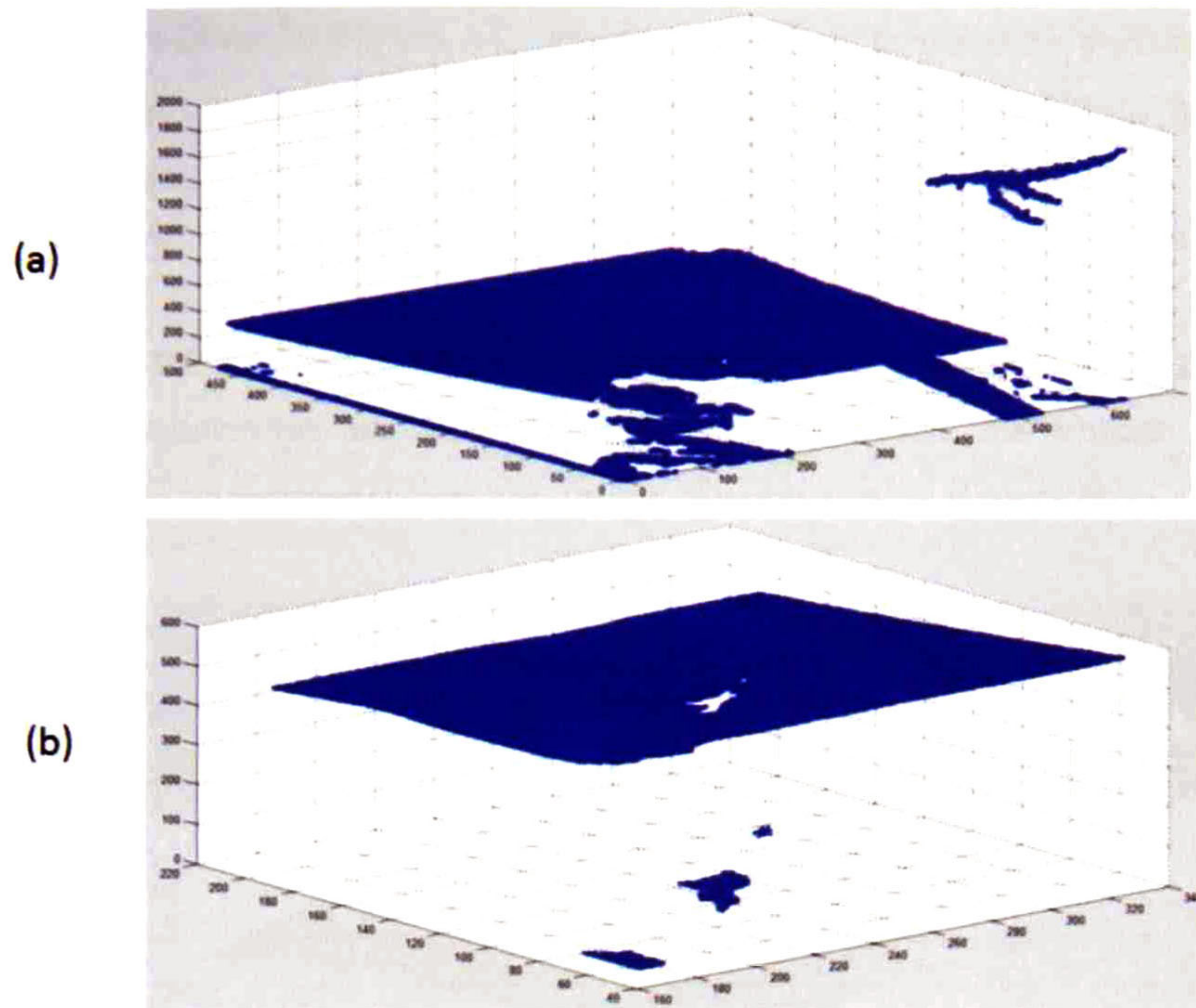


Figura 53. Se muestra en a) la nube de puntos 3D obtenida originalmente con la cámara y en b) solamente el área de interés que contiene al phantom

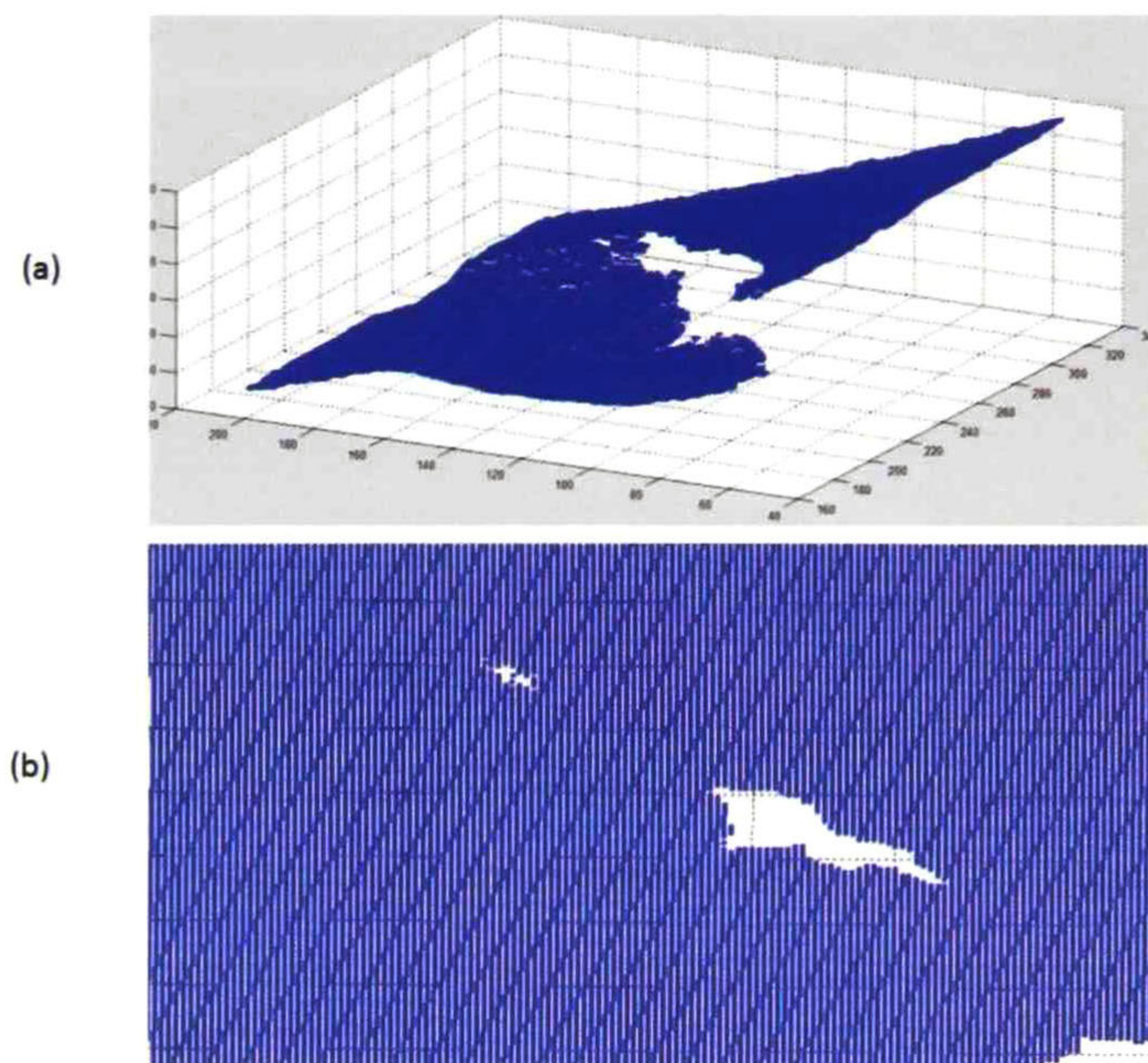


Figura 54. Se muestra la nube de puntos obtenida con la cámara RGB-D del phantom en a) una vista lateral, en b) una vista desde superior.

observar en la Figura 54, en a) se muestra la nube de puntos desde un costado, en b) una vista desde arriba. Hasta este momento la cantidad de puntos es de 24,341.

3. Como tercer paso añadimos en esta misma nube de puntos los puntos que fueron registrados en los contactos con el sensor, los cuales fueron 70 durante el experimento, esto se puede observar en la Figura 55.

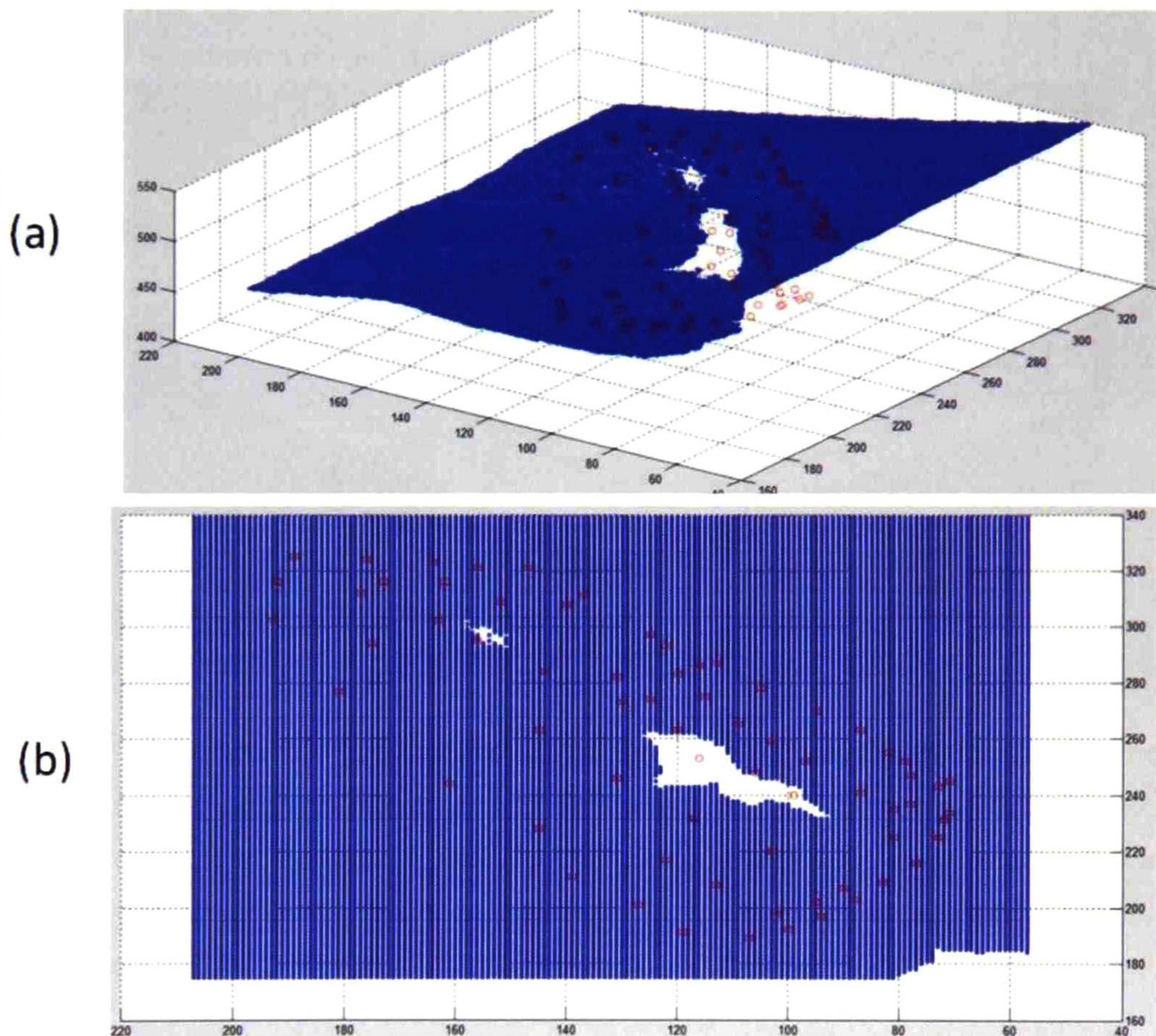


Figura 55. Se muestra la nube de puntos obtenida con la cámara RGB-D del phantom (azul) y los puntos "tocados" con el sensor de tacto (rojo) en a) una vista lateral, en b) una vista desde superior

4. El siguiente paso en el procesamiento es hacer corresponder de manera exacta cada punto de contacto del sensor (rojo) con un punto sobre la

superficie del phantom (azul). Para ello se buscan la coordenadas del pixel más cercano al punto de contacto. El resultado es que en 68 de los 70 puntos de contacto hay una correspondencia perfecta con alguno de los puntos sobre la nube en las coordenadas (x,y) (distancia entre punto de contacto y puntos sobre la nube =0), pero hay pequeñas variaciones en la coordenada z, lo cual es fácilmente comprensible ya que al momento del contacto la superficie del phantom es afectada por la presencia del sensor, por lo que nos quedamos con los 68 puntos de correspondencia perfecta y a cada uno de ellos lo dejamos con la coordenada z de la nube (punto exacto sobre la superficie), esto se muestra en la Figura 56.

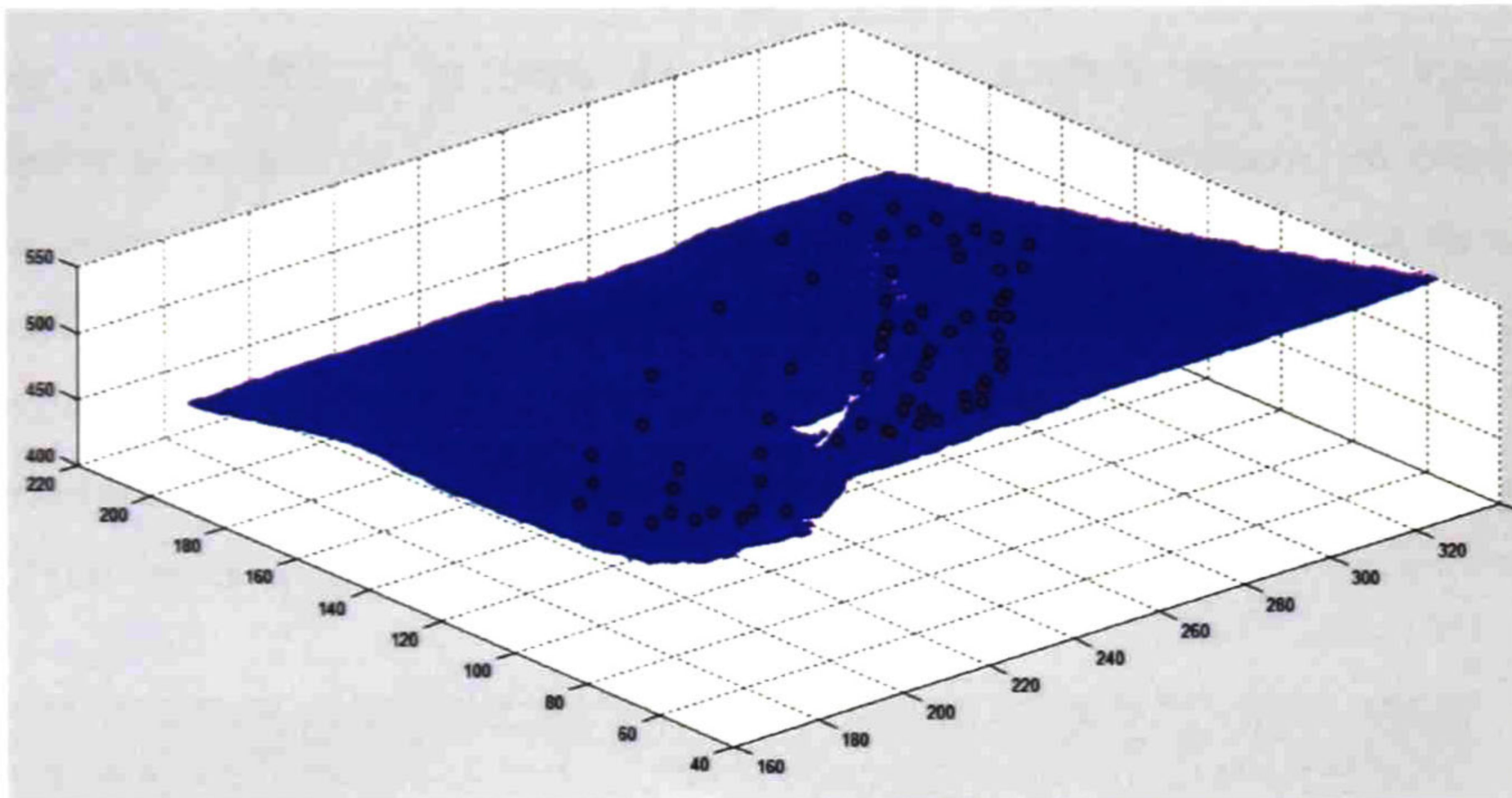


Figura 56. Vista lateral de la nube de puntos de superficie del phantom (azul) con los puntos donde se calculó la rigidez (azul)

5.-Como siguiente paso vamos a quedarnos solo con el 10% de los puntos de la nube ya que no requerimos una densidad tan alta para nuestros cálculos, esto se logra tomando 1 de cada 10 puntos a intervalos constantes, además de conservar los 68 puntos de contacto, el resultado se puede observar en la Figura 57

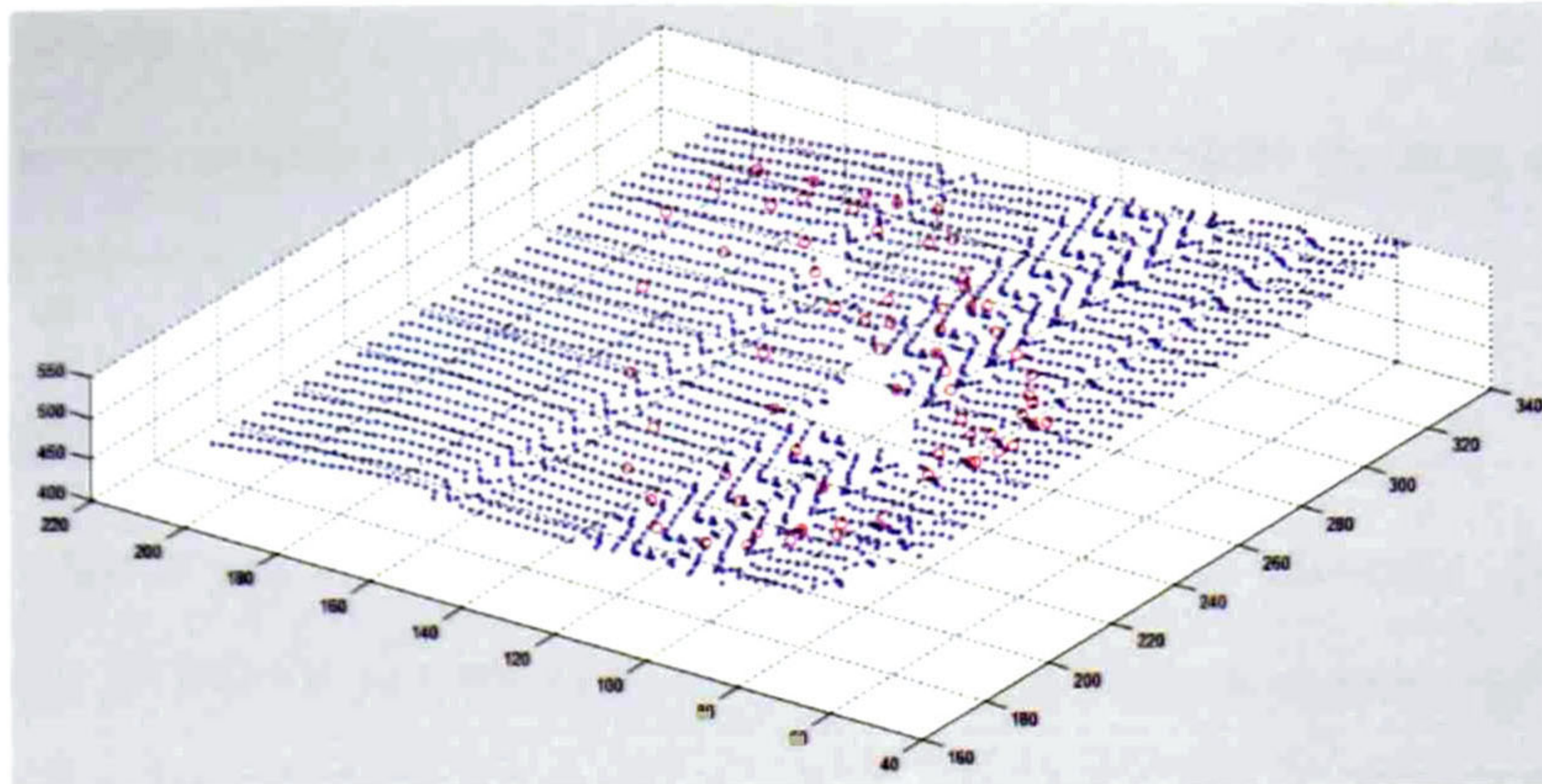


Figura 57.- Nube de puntos reducida al 10% (azul) y puntos muestra para cálculo de la rigidez (rojo)

6.-Por simplicidad a la hora de entrenar el modelo neuronal, vamos a trasladar el origen de los datos (marco referencial de la cámara) al punto  $P_0$  cuyas coordenadas  $(x,y)$  corresponden al centroide  $(x,y)$  de la nube de datos y cuya coordenada  $z$  corresponde a la coordenada  $z$  del punto central obtenido en el paso 2 del experimento. Adicionalmente normalizaremos las 3 coordenadas en la misma proporción para la nube caiga en un rango de  $(-1,1)$  en sus 3 ejes, como se puede ver en la Figura 58.

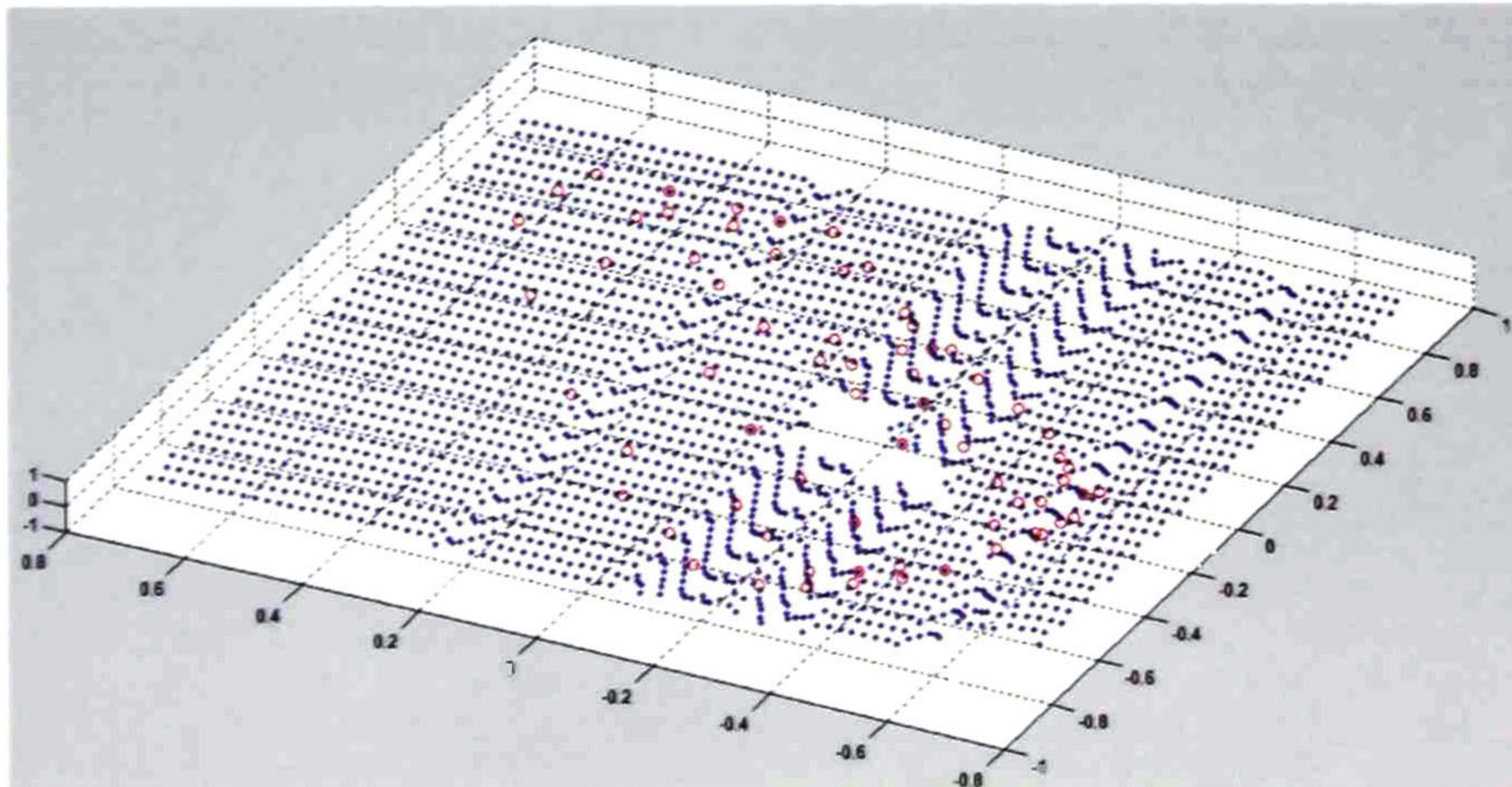
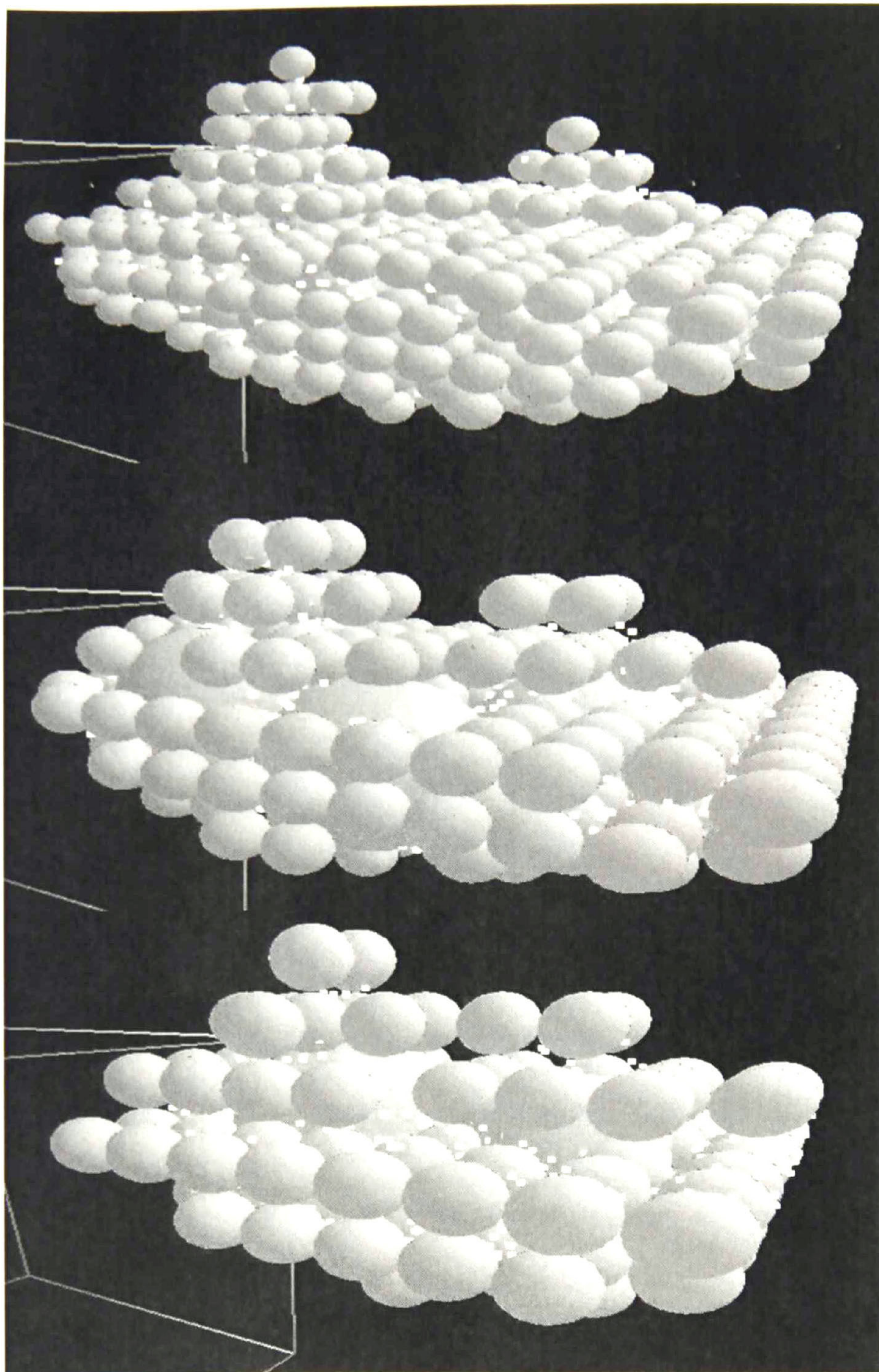


Figura 58. Nube de puntos trasladada al origen y normalizada

Finalmente tenemos lo necesario para probar el modelo, una nube de puntos que representa una superficie espacial y un conjunto muestral de la nube del cual tenemos datos sobre su rigidez superficial.

## Resultados del experimento

Ya con los datos procesados, comenzamos a aplicar el método neuronal. De inicio se toma la nube de puntos procesada y se aplica el algoritmo de reconstrucción con esferas, se realiza para diferentes valores de delta, los resultados se muestran en la Figura 59.



(a)

(b)

(c)

Figura59. Se muestra la reconstrucción de la nube de puntos con esferas tomando diferentes valores de delta, en a)  $\delta=0.15$ , en b)  $\delta=0.1$  y en c)  $\delta=0.05$

Para darle una mejor precisión a los resultados del experimento, tomamos  $\delta=0.05$  (primer paso del *sphere packing algorithm*). Quitamos la nube de puntos que dio origen a nuestra nube de esferas, y nos quedamos solo con un punto exterior de cada esfera según se explicó en el modelo. En la Figura 60 a) se puede observar la nube de esferas con su punto representativo, en b) se puede observar la nube de puntos reducida (de aquí en adelante vamos a trabajar solo con estos puntos, aunque recordemos que cada uno de ellos es representativo de una esfera).

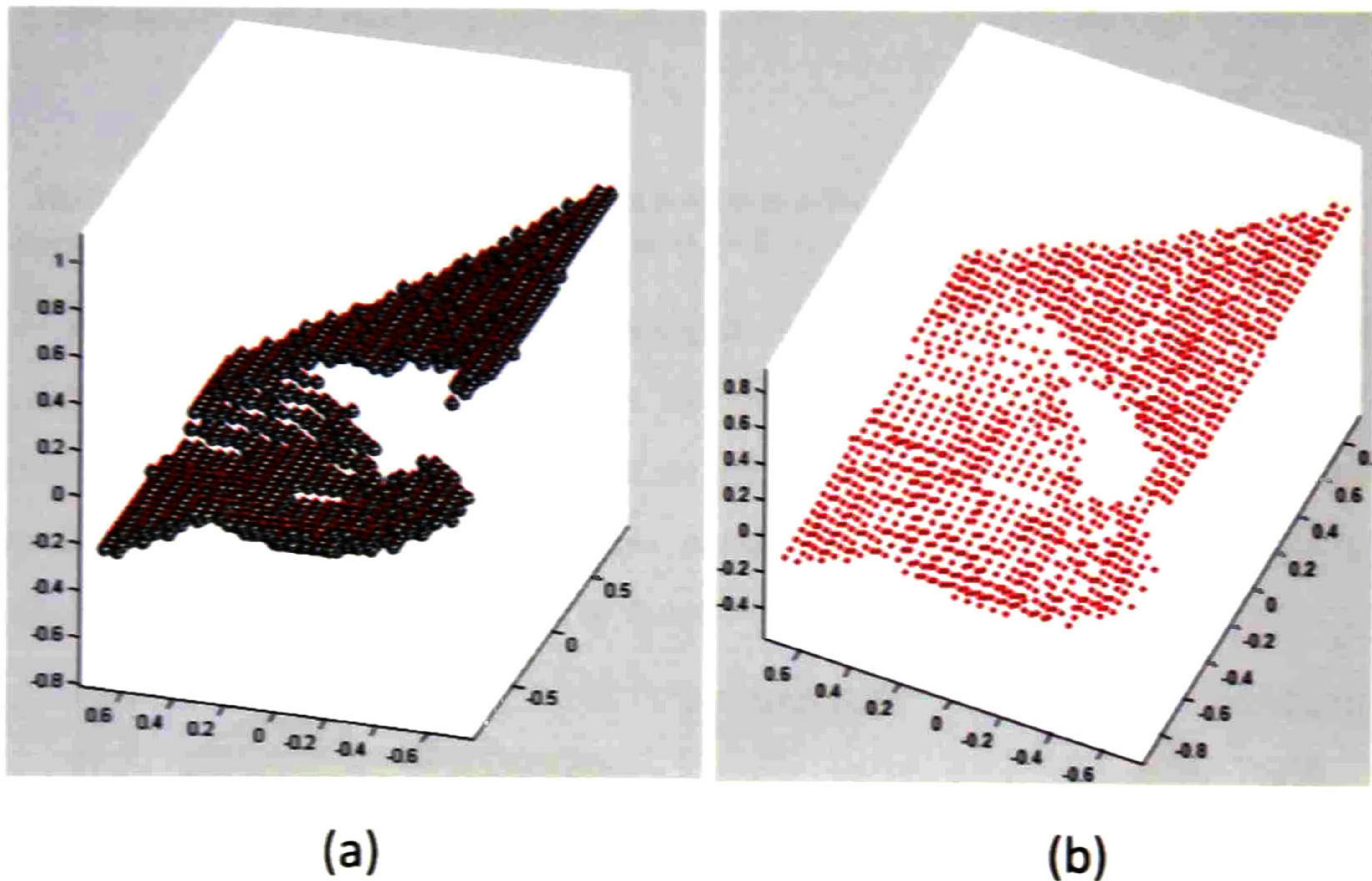


Figura 60. Se muestra en (a) la nube de esferas representativa de las superficies, y en b) la nube de puntos reducida (cada punto representa una esfera)

Siguiendo con el modelo neuronal, en la Figura 61 se muestra esta nube de puntos reducida en rojo, además de la nube muestral de la cual conocemos su rigidez superficial en azul. La vista lateral permite darnos una idea de las magnitudes de las mediciones obtenidas con el sensor de fuerza (libras).



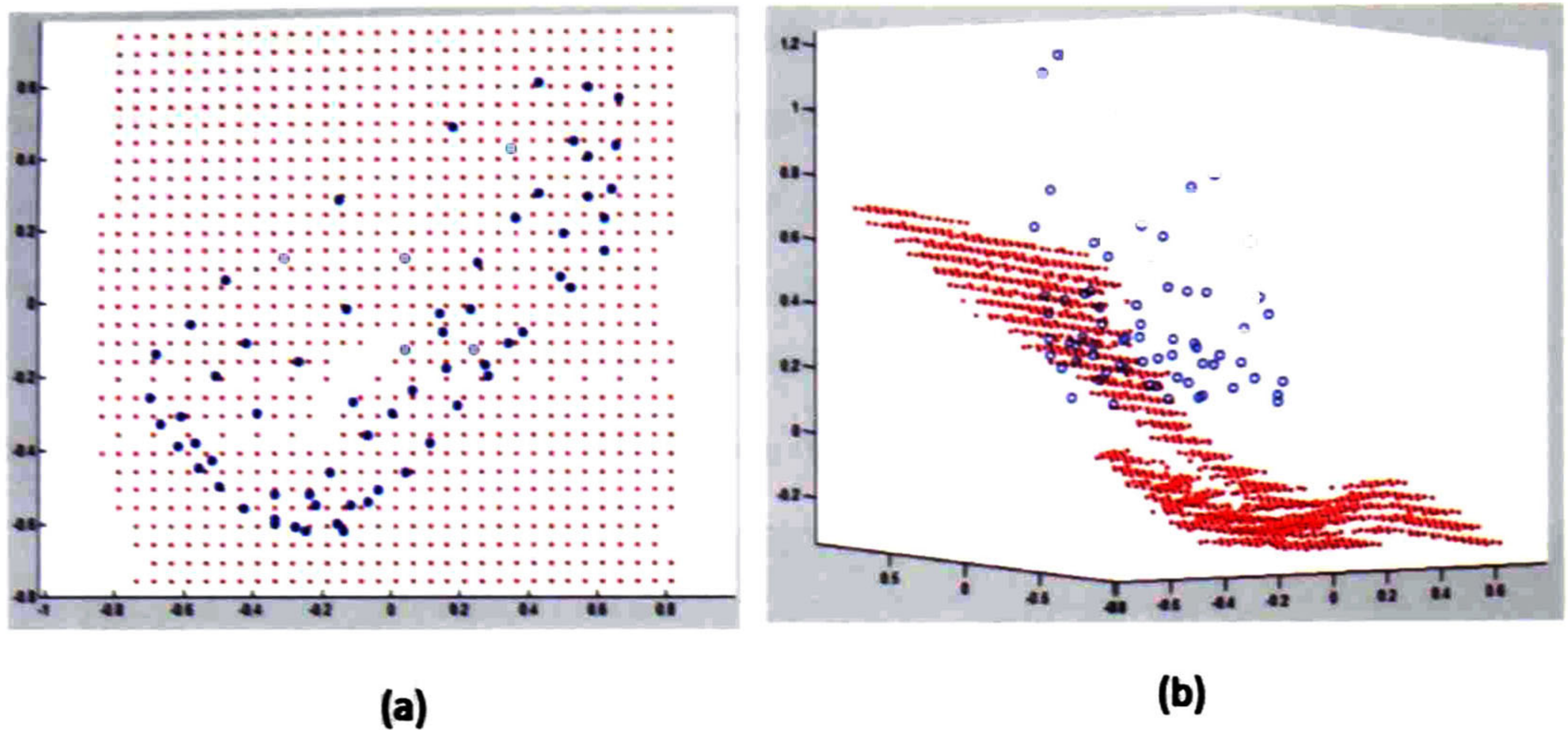


Figura 61. Se muestra desde la parte de arriba(a) la nube de puntos reducida (rojo) junto con los puntos muestrales donde se conoce la rigidez superficial (azul). En (b) la misma figura desde una perspectiva lateral.

Ahora tomamos los datos sobre la rigidez superficial y entrenamos la red con estos. Como se mencionó antes, el tiempo de entrenamiento es muy corto (inmediato desde el punto de vista de un ser humano); se usó un spread de 0.06, en la Figura 62 se muestran varias vistas de la reconstrucción de función de rigidez superficial obtenida por la red la red.

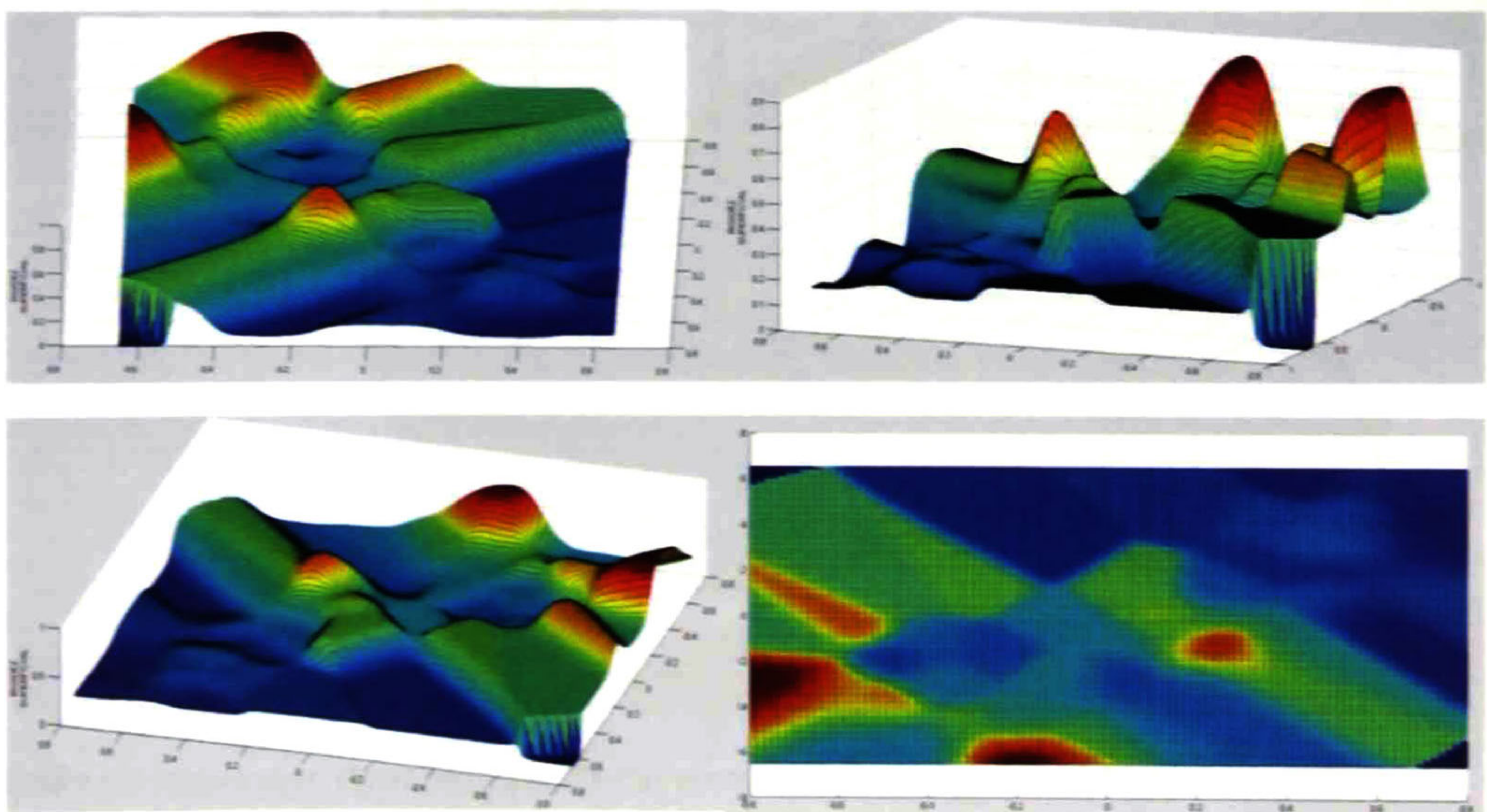


Figura 62. Varias vistas de la función de rigidez superficial obtenida por la red

En la se muestran zonas correspondientes entre el phantom y la función de rigidez superficial  $K(x,y,z)$ . En la reconstrucción se observan algunas mesetas que no tendrían en principio razón de aparecer, lo que sucedió en este caso es que en las orillas la altura del phantom es muy pequeña quedando casi pegado a la mesa, entonces cuando el sensor tocaba el phantom casi tocaba la mesa, por lo que percibía una alta dureza dando origen a tales mesetas.

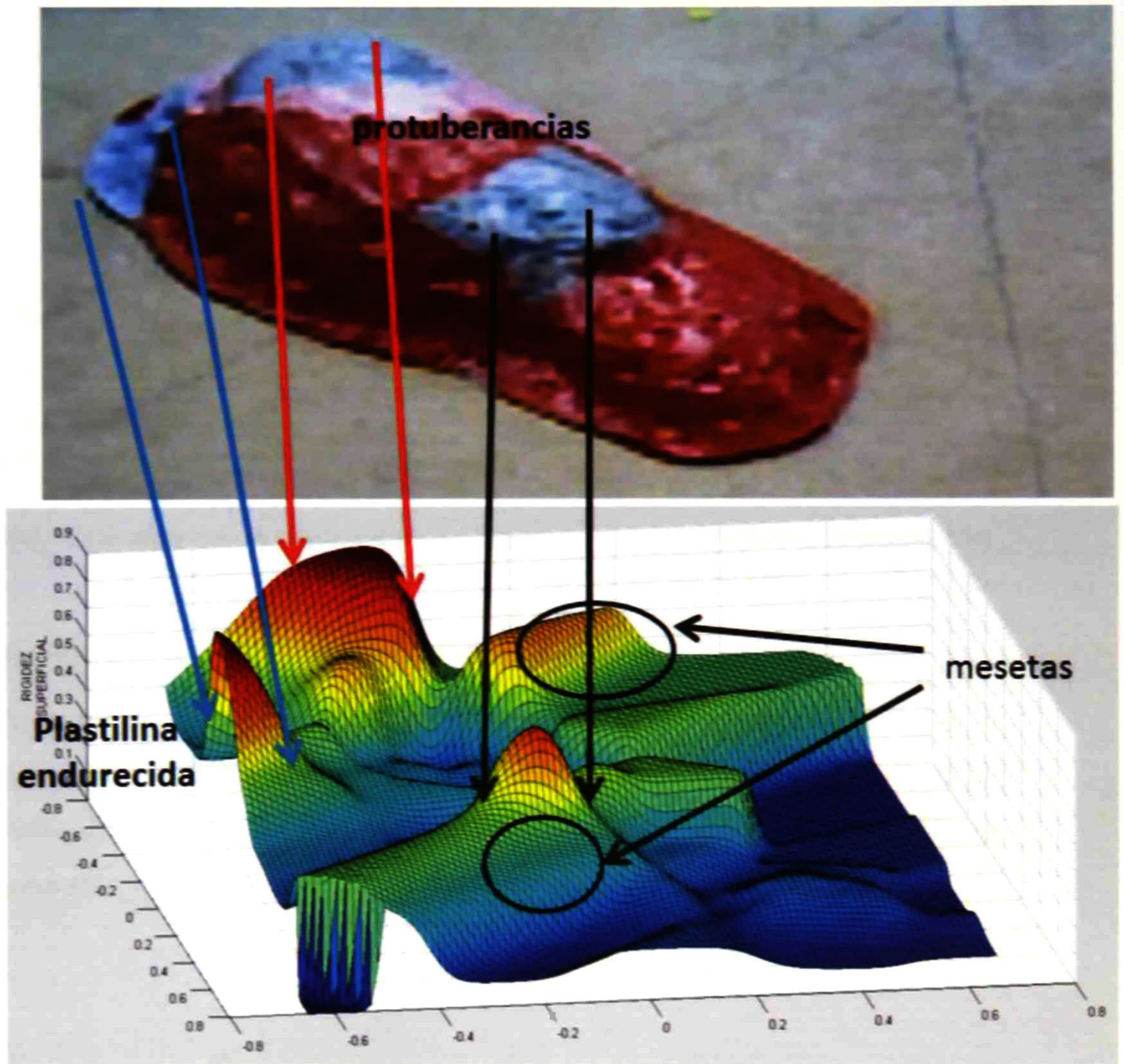


Figura 63. Se muestran zonas correspondientes entre el phantom y la función de rigidez superficial

Ahora usamos la red entrenada para calcular el valor de la rigidez para cada esfera. Se muestran los resultados en la Figura 65, las esferas, las esferas de color rojo más intenso representan zona de mayor rigidez superficial y viceversa.

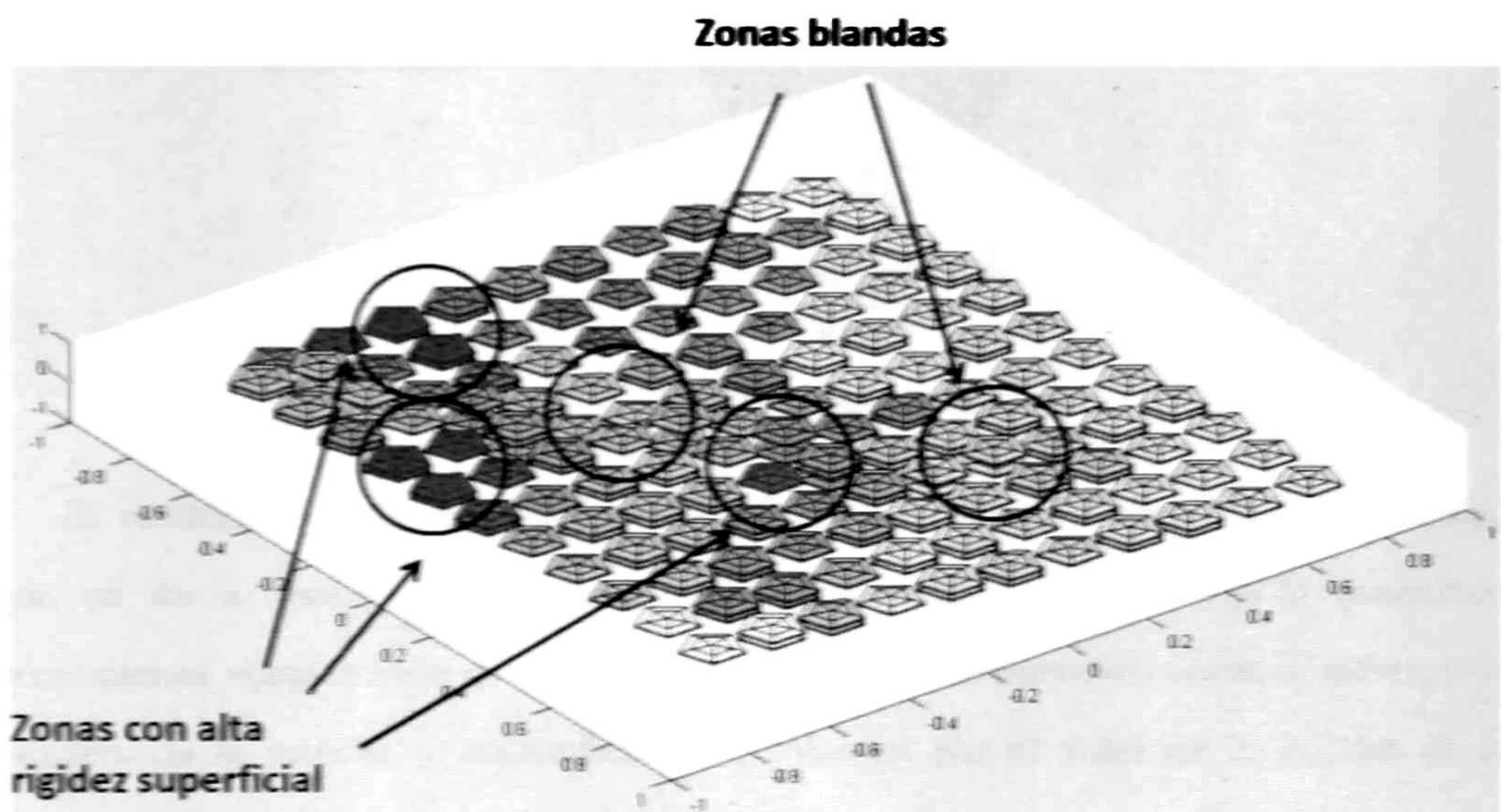


Figura 65. Se muestra la nube de esferas que recrea la superficie del phantom, las esferas de color rojo más intenso representan zona de mayor rigidez superficial y viceversa.

### Interacción con el objeto virtual reconstruido

Como último paso vamos a "tocar" la superficie virtual reconstruida usando la interfaz háptica. Para ello vamos a proponer un renderizado muy simple aprovechando la representación que ya se tiene con esferas. Para entenderlo veamos la Figura 66.

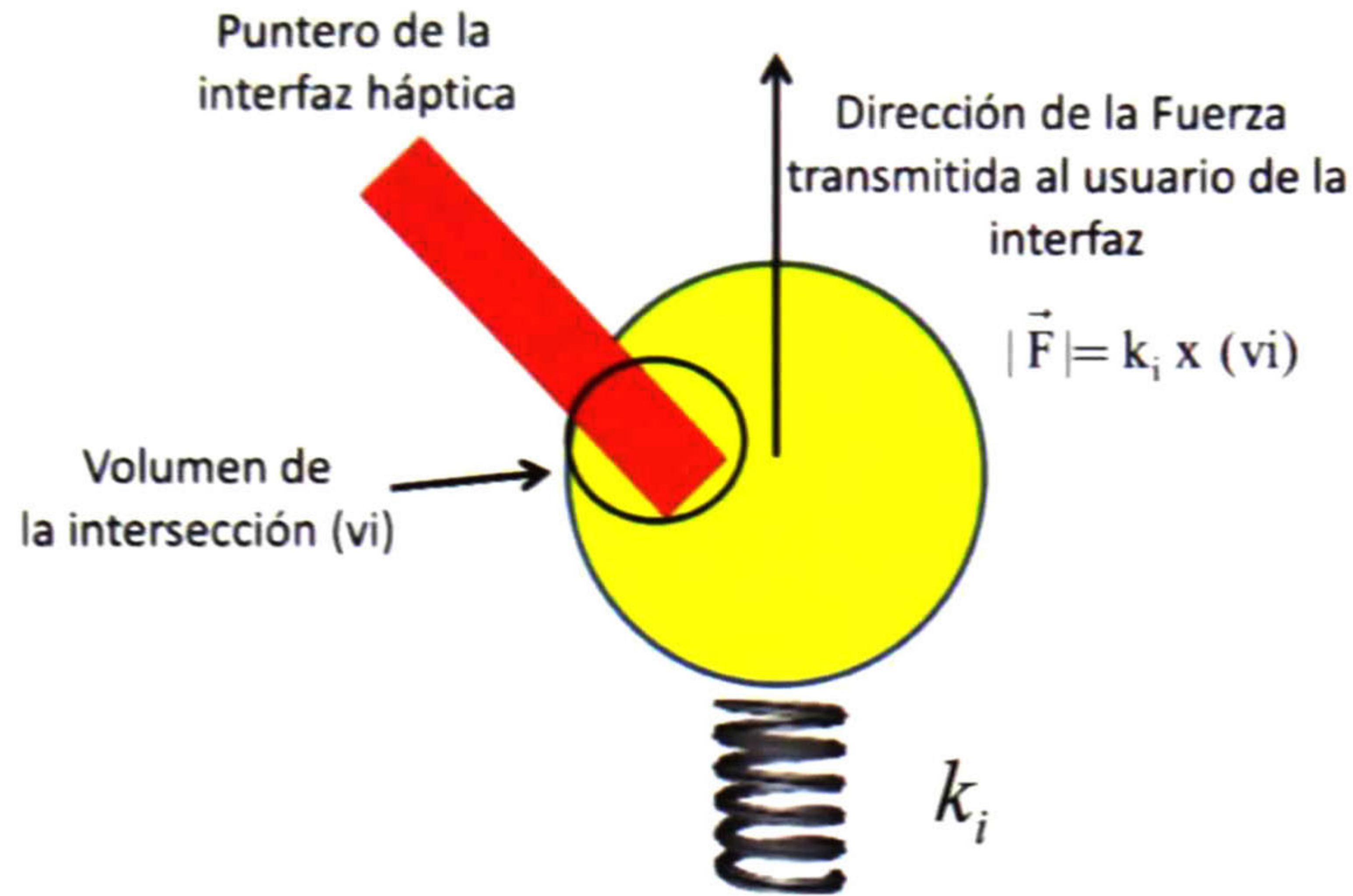


Figura 65. Esquema de renderizado háptico

El renderizado háptico consiste en calcular la fuerza  $\vec{F}$  (magnitud y dirección) que se va a transmitir al usuario de la interfaz. Para calcular la magnitud proponemos simplemente calcular el volumen de la intersección entre la esfera y el puntero de la interfaz y multiplicar este volumen por el valor de la rigidez de la esfera calculada con la red neuronal ( $k$ ). esto es:

$$|\vec{F}| = k_i \times (v_i) \quad (5.3.1)$$

donde  $|\vec{F}|$  es la magnitud de la fuerza transmitida al usuario de la interfaz,  $v_i$  es el volumen de la intersección entre la esfera y el puntero de la interfaz y  $k_i$  es la rigidez de la esfera obtenida con la red neuronal. La dirección de la fuerza será simplemente la dirección ortogonal al plano  $xy$  donde se realizó la reconstrucción (como trabajo a futuro se va a calcular la dirección ortogonal a la superficie reconstruida, esto se puede hacer durante el primer paso del Modelo Neuronal Tacto-Visual). En el caso que el puntero interactúe con varias esferas la fuerza transmitida al usuario es simplemente la suma vectorial de fuerzas calculadas con este mismo método para cada esfera (Figura 66).

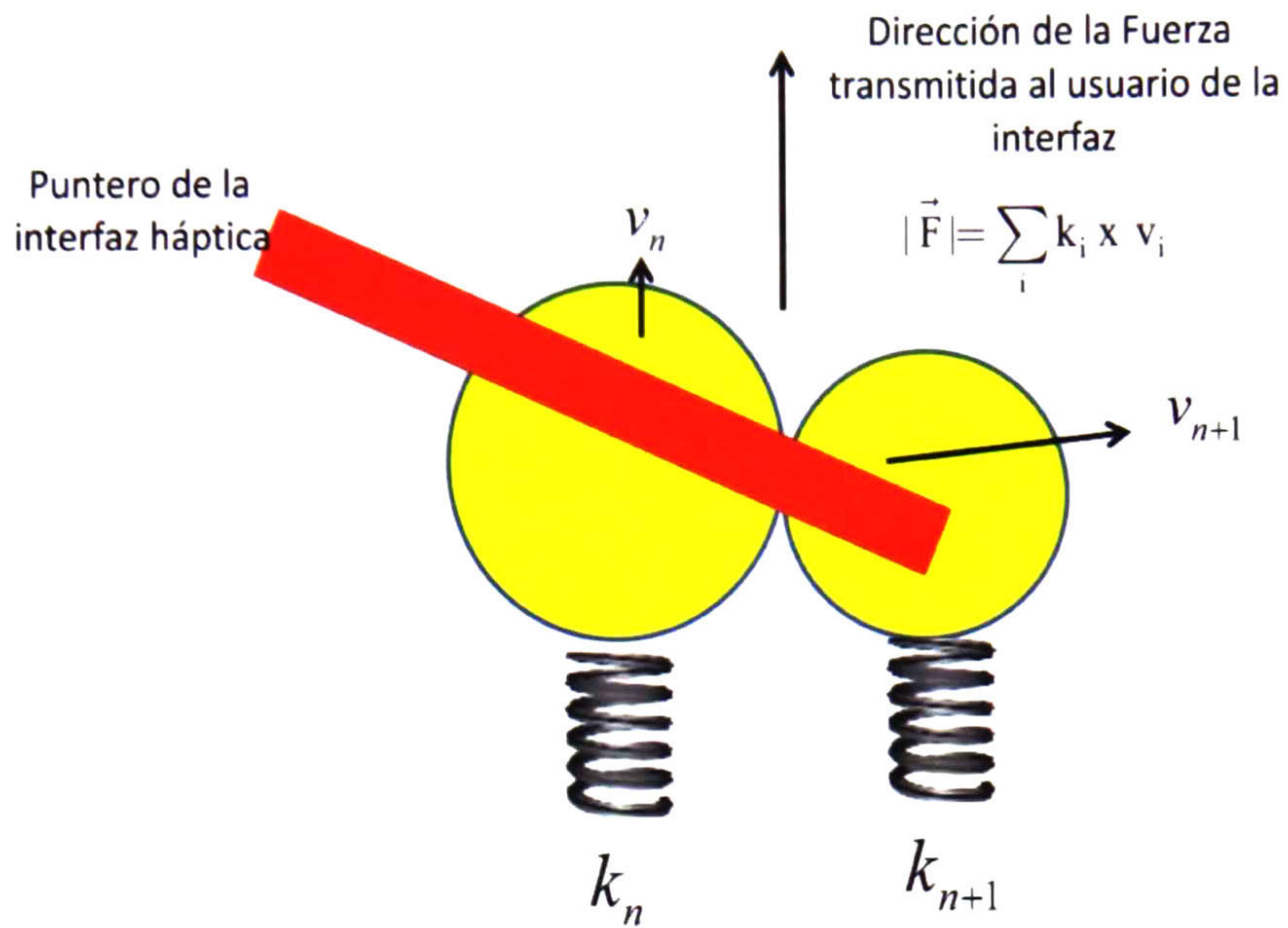


Figura 66. Esquema de renderizado háptico cuando el puntero intersecta más de una esfera.

## 6. Conclusiones y trabajo a futuro

En esta sección se mencionarán las conclusiones del presente trabajo y se describirá también el trabajo restante para concluir la tesis doctoral.

### 6.1 Conclusiones

En este trabajo se ha desarrollado y probado el modelo neuronal tacto-visual que es capaz de representar objetos tridimensionales desde un punto de vista visual-espacial, además que el mismo modelo contiene las características físicas superficiales del objeto en cuestión. También se ha resaltado las posibles aplicaciones que este modelo puede tener para aplicaciones médicas.

Se han presentado los avances en la construcción de un sistema para cirugía de mínima invasión de bajo costo, se ha discutido cuestiones acerca de su diseño y se han descrito algunos de sus principales componentes, haciendo especial énfasis en el papel que juega la interfaz háptica para la manipulación de los brazos robot.

Por otro lado, se desarrolló una nueva red geométrica, la red general de regresión esférica, descrita en el lenguaje del Algebra Conformal, usando esferas o

círculos como funciones de activación, aprovechando la simplicidad que tiene trabajar con estos elementos geométricos en el álgebra ya mencionada. Se mostraron diversas aplicaciones resaltando aplicaciones en áreas médicas, específicamente para la reconstrucción de objetos de formas irregulares como órganos y se propone como una posible alternativa su uso en aplicaciones hápticas.

La red descrita en este trabajo puede ser usada para representar superficies altamente no lineales variando los centros de los círculos o esferas. De manera opuesta a las GRNN convencional, nuestra red codifica superficies no lineales usando entidades geométricas como esferas, lo cual es esencial para relacionar datos perceptuales con restricciones geométricas de mecanismos robotizados (Figura 67).

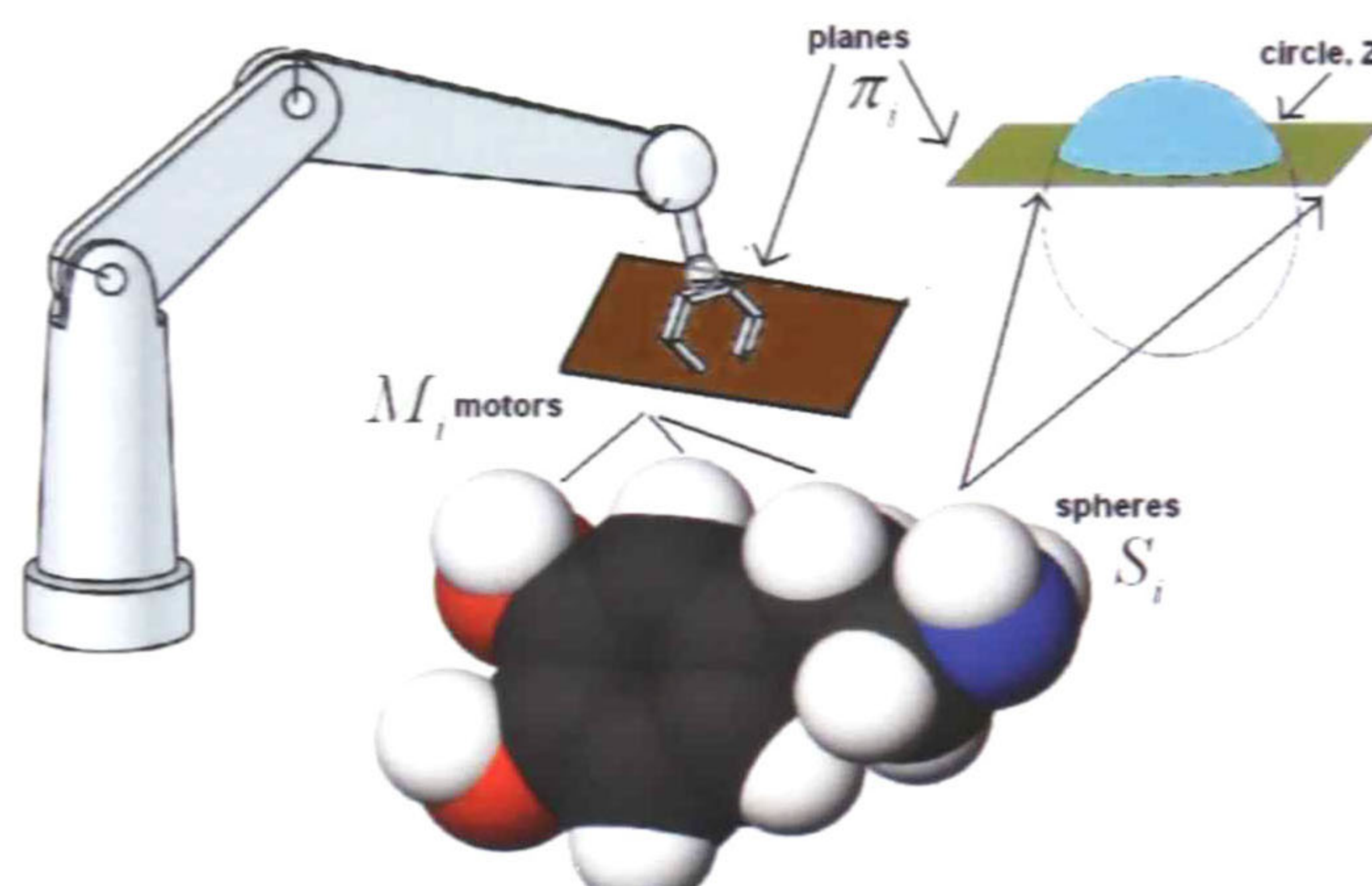


Figura 67. El efector final de un robot puede ser relacionado con esferas individuales representadas en nuestra red usando el mismo lenguaje geométrico.

## 6.2 Trabajo a futuro

Se plantean los siguientes puntos que podrían seguir a este trabajo:

- Mejora del experimento para probar el modelo neuronal tacto-visual fusionando varias vistas, esto permitirá mejorar la representación visual del objeto.
- Experimento de laboratorio para la reconstrucción táctil de objeto aprovechando la interfaz háptica, se deberá corroborar que el uso del sistema háptico mejorará la reconstrucción del objeto usando un muestreo de características superficiales mucho menor.
- Puesta en marcha del sistema de cirugía de mínima invasión, deberán quedar los robots trabajando por medio del uso de la interfaz háptica.
- Solicitud de patente/modelo de utilidad para el Modelo Neuronal Tacto-Visual con renderizado háptico propuesto.



## 7 Publicaciones

### **En Revista:**

*Eduardo Bayro-Corrochano, Eduardo Vazquez-Santacruz, Eduardo Moya-Sanchez and Efrain Castillo-Muñiz, Geometric Bio-Inspired Networks for Recognition of 2D and 3D Low-Level Structures, Neural Networks (submitted).*

### **En Conferencias internacionales:**

*Castillo-Muniz, E.; Bayro-Corrochano, E., "The Spherical Radial Function Network for reconstruction in medical robotics", in proceedings of the Biomedical Robotics and Biomechatronics, BioRob 2012, pp.974,979*

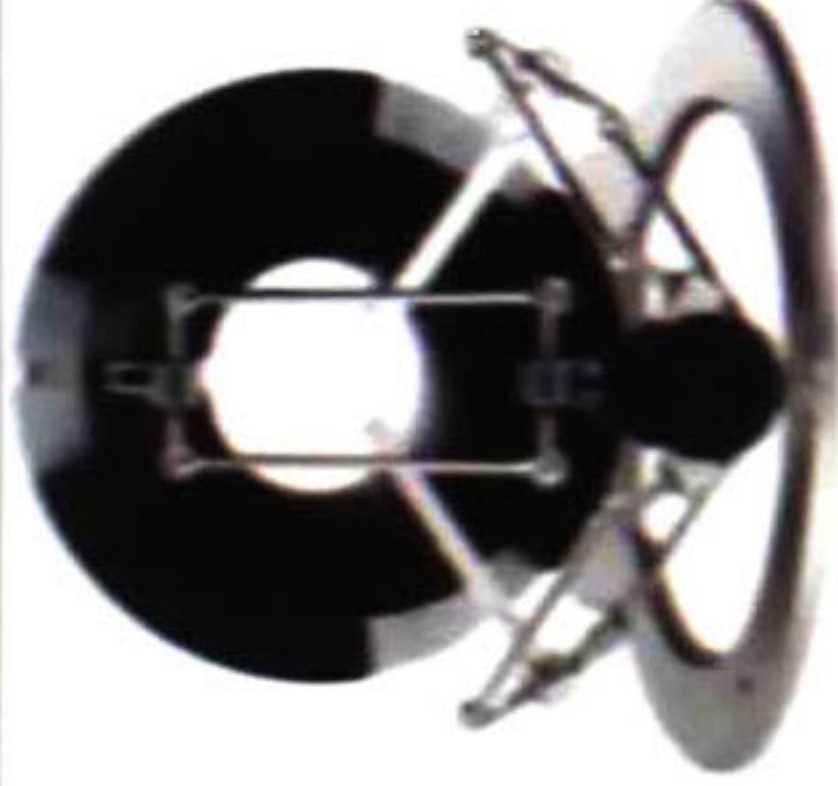
*Castillo-Muniz, E.; Bayro-Corrochano, E., "Geometric Spherical Networks for Visual Data Processing", in proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), pp.1,8.*





*Castillo-Muniz, E.; Bayro-Corrochano, E., "The Spherical General Regression Network for reconstruction in medical robotics," World Automation Congress (WAC), 2012 , pp.1,6, 24-28*



*Efrain Castillo-Muñiz, Jorge Rivera-Rovelo, Eduardo Bayro-Corrochano: "Using Haptic and Neural Networks for Surface and Mechanical Properties 3D Reconstruction", in Proceedings of the 6th international conference on Ubiquitous Computing and Ambient Intelligence (UCAmI'12), pp. 422,429.*

# 7 ANEXOS

## 7.1 Comparativa de diferentes interfaces hápticas comerciales

Distribuidor	interfaz	Foto	Aplicaciones Médicas	DOF	Sotware	Costo
ForceDimension (Suiza) <a href="http://www.forcedimension.com/">http://www.forcedimension.com/</a>	delta		Cuentan con varios tipos de interfaces hechas específicamente para aplicaciones médicas y otras para otros tipos de aplicaciones	3,6	Force Dimension SDK, escrito en C++ y la librería CHAI 3D, en modalidad OPEN SOURCE	\$25000 dlls

Haption (Francia) <a href="http://www.haption.com/site/eng/index.html">http://www.haption.com/site/eng/index.html</a>	serial		Sus interfaces se usan tanto en industria como en aplicaciones médicas	6	Diferentes API's para trabajar y complementar con software CAD	
Novint Technologies <a href="http://home.novint.com/products/novint_falcon.hp">http://home.novint.com/products/novint_falcon.hp</a>	delta		Esta interfaz es la más barata del mercado y se oferta para videojuegos, pero puede ser usado para diferentes propósitos.	3	Librerías para desarrollo en C++	\$300 dls.
MBP (Canadá) <a href="http://www.mpb-technologies.ca/">http://www.mpb-technologies.ca/</a>	Cubico (paralelo)		Aplicaciones médicas e industriales. El de 7 grados de libertad tiene unas tijeras aparte de los 6 grados de libertad	6,7		
Quanser <a href="http://www.quanser.com/choice.asp">http://www.quanser.com/choice.asp</a>	Tipo pantógrafo		Aplicaciones de investigación y didácticas. También se ofrece en paquete con un robot.	5	Quarck	\$6000 dlls.

<p>Sensable (USA)</p> <p><a href="http://www.sensable.com">www.sensable.com</a></p>	<p>Serial</p>		<p>Este es la interfaz más vendida desde su creación. Tienen varios modelos similares con diferentes prestaciones y ha sido usado en aplicaciones médicas y de otros muchos tipos</p>	<p>6</p>	<p>Software propio o regala librerías para desarrollo en C++</p>	<p>\$2200 dls.</p>
<p>Xitact (Suiza)</p> <p><a href="http://www.xitact.com/">http://www.xitact.com/</a></p>	<p>-----</p>		<p>Esta empresa desarrolla interfaces hápticas para simular los instrumentos en un procedimiento médico,</p>	<p>5</p>		<p>\$30000 dls</p>

## 7.2 Robots Motoman



ROBOT SPECIFICATIONS			
Robots	MHSF	MHSLF	
Structure	Vertical jointed-arm type	Vertical jointed-arm type	
Controlled Axes	6	6	
Payload	5 kg (11 lbs)	5 kg (11 lbs)	
Vertical Reach	1,193 mm (47")	1,560 mm (61.4")	
Horizontal Reach	706 mm (27.8")	895 mm (35.2")	
Repeatability	±0.02 mm (±0.0008")	±0.03 mm (±0.001")	
Maximum Motion Range	S-Axis (Turning/Sweep)	±170°	±170°
	L-Axis (Lower Arm)	+150°/-65°	+150°/-65°
	U-Axis (Upper Arm)	+255°/-136°	+255°/-138°
	R-Axis (Wrist Roll)	±190°	±190°
	B-Axis (Bend/Pitch/Yaw)	±135°	±135°
T-Axis (Wrist Twist)	±360°	±360°	
Maximum Speed	S-Axis	376°/s	270°/s
	L-Axis	350°/s	280°/s
	U-Axis	400°/s	300°/s
	R-Axis	450°/s	450°/s
	B-Axis	450°/s	450°/s
T-Axis	720°/s	720°/s	
Approximate Mass	27 kg (59.5 lbs)	29 kg (63.9 lbs)	
Brakes	All axes	All axes	
Power Consumption	1 kVA	1 kVA	
Allowable Moment	R-Axis	12 N · m	12 N · m
	B-Axis	12 N · m	12 N · m
	T-Axis	7 N · m	7 N · m
Allowable Moment of Inertia	R-Axis	0.3 kg · m <sup>2</sup>	0.3 kg · m <sup>2</sup>
	B-Axis	0.3 kg · m <sup>2</sup>	0.3 kg · m <sup>2</sup>
	T-Axis	0.1 kg · m <sup>2</sup>	0.1 kg · m <sup>2</sup>
Internal Electric Cable	3 BC 10 Conductors 4 BC 6 Conductors	3 BC 10 Conductors 4 BC 6 Conductors	
Internal Air Hose	2 - 1/4" PT Connections	2 - 1/4" PT Connections	

FS100 CONTROLLER SPECIFICATIONS*	
Dimensions (mm)	470 (w) x 200 (h) x 420 (d) (18.5" x 7.9" x 16.5")
Approximate Mass	20 kg (44.1 lbs)
Cooling System	Direct cooling
Ambient Temperature	During operation: 0° to 40° C (32° to 104° F) During transit and storage: -10° to 60° C (14° to 140° F)
Relative Humidity	90% max. non-condensing
Primary Power Requirements	Single-phase or 3-phase power, 200/230 VAC at 50/60 Hz (MPP3, MPK2, MH6F, HP20F require 3-phase)
External Transformer (optional)	For 480/575 VAC installations
Digital I/O	NPN-Standard PNP-Optional
Position Feedback	Absolute encoder
Program Memory	JOB: 10,000 steps, 1,000 instructions CIO Ladder: 1,500 steps
Pendant Dim. (mm)	169 (w) x 314.5 (h) x 50 (d) (6.7" x 12.4" x 2")
Pendant Weight	998 kg (2.2 lbs)
Interface	One Compact Flash slot; One USB port (1.1)
Pendant Playback Buttons	Teach/Play/Remote Keypress selector Servo On, Start, Hold, and Emergency Stop Buttons
Programming Language	INFORM III, menu-driven programming, MotoPlus SDK (C language) – optional
Maintenance Functions	Displays troubleshooting for alarms
Number of Robots/Axes	Up to 2 robots, 16 axes (requires 2 controllers)
Multi Tasking	Up to 6 concurrent jobs, 1 system job
Fieldbus	All common networks supported
Ethernet	10 Base T/100 Base TX
Safety	Dual-channel Emergency Stop Pushbuttons, 3-position Enable Switch, Manual Brake Release

Note: Use DX100 controller for arc welding applications.  
\* See FS100 Controller data sheet (DG-S09) for complete specifications

## 7.3 Cámara RGB -D

Este sensor RGB-D, el PrimeSense Sensor RD 1.09 Figura 7, fabricado y patentado por la empresa Israelí PrimeSense es capaz de escanear objetos en un rango de 0.35-1.4m, tiene una resolución de imagen de 640 x 480 (VGA) y no se vende comercialmente, por lo que la información técnica disponible acerca de este dispositivo es muy limitada.

## 7.4 Sensor de Fuerza



Celda de carga miniatura universal, fabricada por Loadstar Sensors tiene las siguientes características operativas:

1. 25mm de diámetro por 30 cm de largo
2. Uso en Presión o Tensión +/- 5%
3. Conexión usb o wireless
4. Salida Analoga 0-5V
5. No linealidad 0.5%

## 7.5 Programa en CLUCalc V4.3.2

Este código en CLUCalc recibe una nube de puntos y ajusta esferas de diferentes radios a la misma.

```

FrameBoxSize = 0;
DefVarsN3();

:IPNS;
//Defino un plano con 3 puntos
:White;
:IPNS;
p4=VecN3(-0.2,0.3,0);
p5=VecN3(0.4,0.2,0);
p6=VecN3(0.1,-0.5,0);
pla=(p4^p5^p6^e); //plano de proyección

//Aquí leo un archivo con la nube de
L=ReadData("processed_cloud.txt");
N=Size(L);
Puntos=Matrix([L(1)]);
minx=99999;
maxx=-99999;
miny=99999;
maxy=-99999;
minz=99999;
maxz=-99999;
delta=0.15;

i=2;
cont=0;

loop //leo todos los puntos y los pongo en la
matriz Puntos
{
    if (i > N-1)
        break;
    fila=Matrix([L(i)]);
    Puntos=AddRows(Puntos,fila);
}

if (Puntos(i,1)>maxx)
    maxx=Puntos(i,1);

if (Puntos(i,1)<minx)
    minx=Puntos(i,1);

if (Puntos(i,2)>maxy)
    maxy=Puntos(i,2);

if (Puntos(i,2)<miny)
    miny=Puntos(i,2);

if (Puntos(i,3)>maxz)
    maxz=Puntos(i,3);

if (Puntos(i,3)<minz)
    minz=Puntos(i,3);

i=i+1;
cont=cont+1;
}

i=1;

:OPNS;
loop
{
    if (i > cont)
        break;
    :White;
    :a=VecN3(Puntos(i,1),Puntos(i,2),Puntos
(i,3));
    i=i+1;
}

```



```

//hacemos un voxelizado de tamaño mm x nn,
lleno de 0's donde no hay figura y 1's donde si
mm=ceil((maxy-miny+delta/2)/delta)+2;
nn=ceil((maxx-minx+delta/2)/delta)+2;
pp=ceil((maxz-minz+delta/2)/delta)+2;

```

```
Vox_tot=Matrix(mm,nn*pp);
```

```
i=1;
```

```

//Aqui tengo un voxelizado lleno de ceros,
pongo unos donde hay puntos

```

```

loop
{
    if (i > N-1)
        break;
    x=Puntos(i,1);
    y=Puntos(i,2);
    z=Puntos(i,3);
    ptx=ceil((x-minx+delta/2)/delta)+1;
    pty=ceil((y-miny+delta/2)/delta)+1;
    ptz=ceil((z-minz+delta/2)/delta)+1;
    Vox_tot(mm-pty+1,ptx+nn*(ptz-1))=1;

    i=i+1;
}

```

```
//Cuento cuantas celdas (cont1) tienen 1's
```

```

i=2;
cont1=0;

```

```

loop
{
    if(i>mm) break;
    j=1;
    loop
    {
        if (j>nn*pp) break;

        if(Vox_tot(i,j)==1) cont1=cont1+1;
        j=j+1;
    }
    i=i+1;
}

```

```

//Calculo Vect1 donde guardo todas las celdas y
su distancia a la superficie

```

```
Vect1=Matrix(cont1,3);
```

```

cont1=0;
i=2;
loop
{
    if( i > mm) break;
    j=2;
    loop
    {
        if(j > nn*pp) break;

        if(Vox_tot(i,j)==1)
        {

```

```

            cont1=cont1+1;
            dmin=9999999999;
            dist=0;
            cont=1;
            loop //mido distancia hacia arriba
            {
                if(Vox_tot(i-cont,j)==0)
                    break;
                else
                {
                    dist=dist+1;
                    cont=cont+1;
                }
            }
            if(dmin>dist)
                dmin=dist;
        }
    }
    dist=0;
    cont=1;

```

```

        loop //mido distancia hacia abajo
        {
            if(Vox_tot(i+cont,j)==0)
                break;
            else
            {
                dist=dist+1;
                cont=cont+1;
            }
        }
    }
    if(dmin>dist)
        dmin=dist;
    dist=0;
    cont=1;

```

```
        loop //mido distancia hacia la derecha
```

```

{
  if(Vox_tot(i,j+cont)==0)
    break;
  else
  {
    dist=dist+1;
    cont=cont+1;
  }
}
if(dmin>dist)
  dmin=dist;

dist=0;
cont=1;

loop //mido distancia hacia la
izquierda
{
  if(Vox_tot(i,j-cont)==0)
    break;
  else
  {
    dist=dist+1;
    cont=cont+1;
  }
}
if(dmin>dist)
  dmin=dist;

dist=0;
cont=1;

loop //mido distancia hacia el frente
{
  if(Vox_tot(i,j+(cont*nn))==0)
    break;
  else
  {
    dist=dist+1;
    cont=cont+1;
  }
}
if(dmin>dist)
  dmin=dist;

dist=0;
cont=1;

loop //mido distancia hacia atras

```

```

{
  if(Vox_tot(i,j-(cont*nn))==0)
    break;
  else
  {
    dist=dist+1;
    cont=cont+1;
  }
}
if(dmin>dist)
  dmin=dist;

dist=0;
cont=1;

loop //mido distancia hacia 45 grados
{
  if(Vox_tot(i+cont,j+cont)==0)
    break;
  else
  {
    dist=dist+1;
    cont=cont+1;
  }
}
if(dmin>dist)
  dmin=dist*sqrt(2);

dist=0;
cont=1;

loop //mido distancia hacia 135 grados
{
  if(Vox_tot(i-cont,j+cont)==0)
    break;
  else
  {
    dist=dist+1;
    cont=cont+1;
  }
}
if(dmin>dist)
  dmin=dist*sqrt(2);

dist=0;
cont=1;

loop //mido distancia hacia 225 grados
{
  if(Vox_tot(i-cont,j-cont)==0)

```

```

        break;
    else
    {
        dist=dist+1;
        cont=cont+1;
    }
}
if(dmin>dist)
    dmin=dist*sqrt(2);

    dist=0;

cont=1;

loop //mido distancia hacia 315 grados
{
    if(Vox_tot(i+cont,j-cont)==0)
        break;
    else
    {
        dist=dist+1;
        cont=cont+1;
    }
}
if(dmin>dist)
    dmin=dist*sqrt(2);

dist=0;
cont=1;

loop //mido distancia hacia 45 grados
fte
{

if(Vox_tot(i+cont,(j+cont)+(cont*nn))==0)
    break;
    else
    {
        dist=dist+1;
        cont=cont+1;
    }
}
if(dmin>dist)
    dmin=dist*sqrt(3);

    dist=0;

cont=1;

loop //mido distancia hacia 135 grados
fte
{

```

```

        if(Vox_tot(i-
cont,j+cont+(cont*nn))==0)
            break;
        else
        {
            dist=dist+1;
            cont=cont+1;
        }
    }
    if(dmin>dist)
        dmin=dist*sqrt(3);

        dist=0;

cont=1;

loop //mido distancia hacia 225 grados
fte
{
    if(Vox_tot(i-cont,j-
cont+(cont*nn))==0)
        break;
    else
    {
        dist=dist+1;
        cont=cont+1;
    }
}
if(dmin>dist)
    dmin=dist*sqrt(3);

    dist=0;

cont=1;

loop //mido distancia hacia 315 grados
fte
{
    if(Vox_tot(i+cont,j-
cont+(cont*nn))==0)
        break;
    else
    {
        dist=dist+1;
        cont=cont+1;
    }
}
if(dmin>dist)
    dmin=dist*sqrt(3);

    dist=0;

cont=1;

```

```

loop //mido distancia hacia 45 grados
atrás
{
  if(Vox_tot(i+cont,(j+cont)-
(cont*nn))==0)
    break;
  else
  {
    dist=dist+1;
    cont=cont+1;
  }
}
if(dmin>dist)
  dmin=dist*sqrt(3);

  dist=0;
  cont=1;

loop //mido distancia hacia 135 grados
atrás
{
  if(Vox_tot(i-cont,j+cont-
(cont*nn))==0)
    break;
  else
  {
    dist=dist+1;
    cont=cont+1;
  }
}
if(dmin>dist)
  dmin=dist*sqrt(3);

  dist=0;
  cont=1;

loop //mido distancia hacia 225 grados
atrás
{
  if(Vox_tot(i-cont,j-cont-
(cont*nn))==0)
    break;
  else
  {
    dist=dist+1;
    cont=cont+1;
  }
}
if(dmin>dist)

```

```

  dmin=dist*sqrt(3);
  dist=0;
  cont=1;

loop //mido distancia hacia 315 grados
atrás
{
  if(Vox_tot(i+cont,j-cont-
(cont*nn))==0)
    break;
  else
  {
    dist=dist+1;
    cont=cont+1;
  }
}
if(dmin>dist)
  dmin=dist*sqrt(3);

  Vect1(cont1,1)=i;
  Vect1(cont1,2)=j;
  Vect1(cont1,3)=dmin;
}

  j=j+1;
}
i=i+1;
}

:Green;

cont2=cont1;
cont3=10;
//Ahora hago un loop para poner las esferas
j=1;
List_esf=[ ];
loop
{
  if(cont2==0) break;

  //Busco en Vect1 la d_max
  i=1;
  dmax=-5;
  loop
  {
    if (i > cont2) break;

    if(dmax<Vect1(i,3))

```

```

{
  x_max=Vect1(i,1);
  y_max=Vect1(i,2);
  dmax=Vect1(i,3);
}

i=i+1;
}

z_max=floor(y_max/nn);
y_maxr=y_max-(z_max*nn);

//// dibujo un círculo centrado en ese punto y
con radio d_max
if(dmax>0)
{
  :Sp = SphereN3(delta*(y_maxr-
2)+minx,miny+delta*(mm-1-
x_max),delta*(z_max-1)+minz, (dmax*delta));
  List_esf<< [delta*(y_maxr-
2)+minx,miny+delta*(mm-1-
x_max),delta*(z_max-1)+minz, (dmax*delta)] ;
}
else
{
  :Sp = SphereN3(delta*(y_maxr-
2)+minx,miny+delta*(mm-1-
x_max),delta*(z_max-1)+minz, (delta/2));
  List_esf<< [delta*(y_maxr-
2)+minx,miny+delta*(mm-1-
x_max),delta*(z_max-1)+minz, (delta/2)] ;
}
}

```

```

/// Quito de Vect1 todos los puntos internos a
la esfera
cont3=0;
i=1;
Vect1_aux=Matrix(cont1,3);

loop
{
  if (i > cont2) break;
  z_max2=floor(Vect1(i,2)/nn);
  y_maxr2=Vect1(i,2)-(z_max2*nn);
  d=sqrt(pow((x_max-
Vect1(i,1)),2)+pow((y_maxr-
y_maxr2),2)+pow((z_max-z_max2),2));

  if(dmax<d)
  {
    cont3=cont3+1;
    Vect1_aux(cont3,1)=Vect1(i,1);
    Vect1_aux(cont3,2)=Vect1(i,2);
    if((d-dmax)<Vect1(i,3))
Vect1_aux(cont3,3)=d-dmax;
    else Vect1_aux(cont3,3)=Vect1(i,3);
  }

  i=i+1;
}
cont2=cont3;
j=j+1;
Vect1=Vect1_aux;

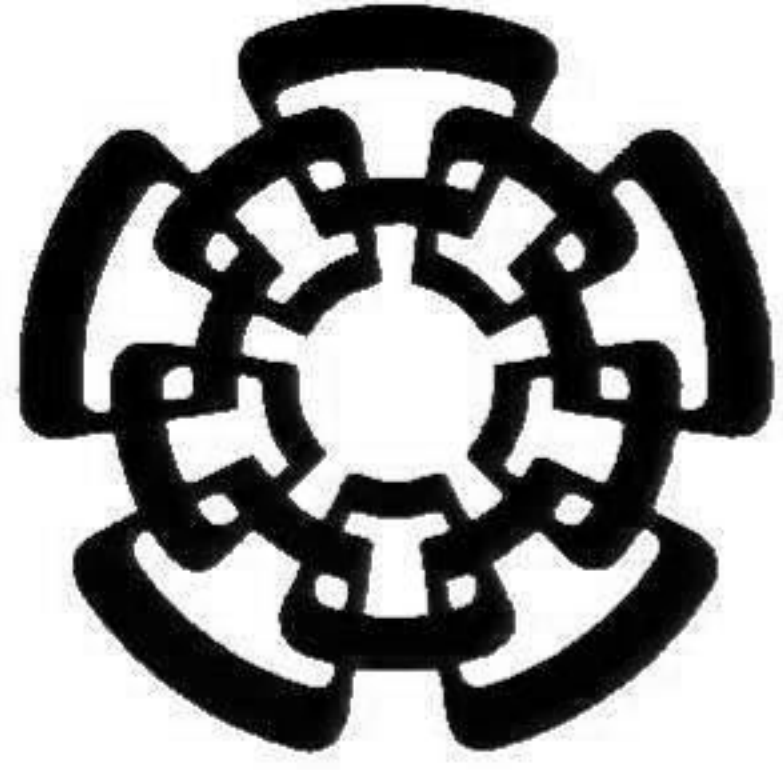
}
WriteData( "Esferas2.dat", List_esf );

```

## Bibliografía

1. *On intelligence*. Hawkins, Jeff and Blakeslee, Sandra. New York : Holt, 2004.
2. Olshausen, Michael Cohnitz. *Pop-up, precision lock-cylinder that reveals at once, with visual and tactile cues, who else with a key has sought or gained entry*. US6516643 B1 USA, 2 11, 2003.
3. *Reference frames for representing visual and tactile locations in parietal cortex*. Marie Avillac, et all. 8, s.l. : nature neuroscience, online, 2005, Vol. 1.
4. *Models and processes of multisensory cue combination*. Seilheimer, Robert L, Rosenberg, Ari and Angelaki , Dora E. s.l. : Elsevier, Theoretical and computational neuroscience, Current Opinion in Neurobiology, Vols. Volume 25, Pages 38–46.
5. Bayro-Corrochano, Eduardo, Buchholds and Sven. Geometric neural networks. [book auth.] Gerald Somer and Koenderink Jan J. *Algebraic Frames for the Perception-Action Cycle*. Kiel, Germany : Springer Berlin Heidelberg, 1997, pp. 379-394.
6. Intuitive Surgical Inc. Intuitive Surgical. [Online] 2014. [Cited: Junio 8 de Junio, 2014.] <http://www.intuitivesurgical.com/>.
7. Resch, Klaus Dieter Maria. *Trasendoscopic Ultrasound for Neurosurgery*. Germany : Springer, 2006.
8. E., Nuño and L., Basañez. Teleoperación: técnicas, aplicaciones, entorno sensorial, y teleoperación inteligente. *PhD. thesis, universidad politécnica de cataluña*. Cataluña : s.n., 2006.
9. Médica Sur, hospital universitario privado de alta especialidad. Medica Sur. [Online] [Cited: Abril 22, 2013.] [http://www.medicasur.com/es\\_mx/ms/Cirugia\\_de\\_Minima\\_Invasion](http://www.medicasur.com/es_mx/ms/Cirugia_de_Minima_Invasion).
10. Tavakoli, M, et al., et al. *Haptic for teleoperated surgical robotic systems*. s.l. : World Scientific, 2008.
11. Salamanca, María Luisa Pinto. Análisis e Implementación de una Interfaz Háptica en entornos virtuales. *Tesis para obtener el grado de magister en ingeniería automatización industrial*. Bogotá, Colombia : Universidad Nacional de Colombia, Facultad de Ingeniería, 2009.
12. Corrochano, Eduardo Bayro. *Geometric Computing: for Wavelet Transforms, Robot Vision, Learning, Control and Action*. s.l. : Springer, 2010.
13. *Applications of Grassmann's Extensive Algebra*. Clifford, W. K. s.l. : The John Hopkins University Press, 1878, American Journal of Mathematics, Vol. 1, pp. 350-358.
14. Li, H., Hestenes, D., y Rockwood, A. Generalized Homogeneous Coordinates for Computational Geometry. [book auth.] Gerald Sommer. *Geometric Computing with Clifford Algebras*. 1. s.l. : Springer, 2000, pp. 27-58.

15. *On how we know universals: The perception of auditory and visual forms.* **McCulloch, Warren and Pitts, Walter.** 9, s.l. : Bulletin of Mathematical Biophysics, 1947.
16. **Rojas, Raúl.** *Neural Networks- A Systematic Introduction.* Berlín : Springer-Verlag, 1996.
17. *A General Regression Neural Network.* **Specht, Donald F.** s.l. : IEEE TRANSACTIONS ON NEURAL NETWORKS, 1991, Vol. 2.
18. *On Estimating Regression.* **Nadaraya, E. A.** 1, s.l. : Theory Probab. Appl., 1963, Vol. 9.
19. **Bauer, M.** *General Regression Neural Network for Technical Use. Master's Thesis.* s.l. : University of Wisconsin-Madison, 1995.
20. **GeoKernels, research project of the Swiss National Science Foundation.** [Online] <http://www.geokernels.org/grnnposteronegu06.pdf>.
21. *The use of Geometric Algebra for 3D Modeling and Registration of Medical Data.* **Bayro-Corrochano, Eduardo and Rivera-Rovelo, Jorge.** 1, s.l. : Journal of Mathematical Imaging and Vision, 2009, Vol. 34.
22. *Marching cubes: A high resolution 3d surface construction alghoritm.* **E., Lorensen W. and Cline, Harvey E.** 4, s.l. : ACM Computer Graphics, 1987, Vol. 21.
23. *A Unified Approach for Physically-Based Simulations and Haptic Rendering.* **Weller, Rene and Zachmann, Gabriel.** s.l. : Proceedings of the 2009 SIGGRAPH Symposium on Video Games, 2009.
24. **Perwass, Dr. Chrisitan B. U. and Grest, Daniel.** CLUCalc V4.3.2, lenguaje que permite programar de manera natural en el lenguaje de las Álgebras Geométricas con interfaz gráfica para visualizaciones.



# **CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N. UNIDAD GUADALAJARA**

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

**Modelo Neuronal de Percepción Táctil y Visual con Aplicaciones  
Hápticas para Sistemas de Cirugía de Mínima Invasión**

del (la) C.

**Efraín CASTILLO MUÑIZ**

el día 27 de Junio de 2014.

**Dr. Eduardo José Bayro Corrochano**  
Investigador CINESTAV 3D  
CINESTAV Unidad Guadalajara

**Dr. Juan Luis Del Valle Padilla**  
Investigador CINESTAV 3C  
CINESTAV Unidad Guadalajara

**Dr. Mario Angel Siller González**  
Pico  
Investigador CINESTAV 3A  
CINESTAV Unidad Guadalajara

**Dra. Susana Ortega Cisneros**  
Investigador CINESTAV 2C  
CINESTAV Unidad Guadalajara

**Dr. Luis Enrique González Jiménez**  
Profesor Investigador  
ITESO





CINVESTAV - IPN  
Biblioteca Central



SSIT0012579