



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Ingeniería Eléctrica

Sección de Computación

**Modelos Deformables para Caracterizar
Macromoléculas Biológicas**

Tesis que presenta

Jorge Eduardo Ramírez Flores

para obtener el Grado de

Maestro en Ciencias

en la Especialidad de

Ingeniería Eléctrica

con Opción en Computación

Director de la Tesis

Dr. Luis Gerardo de la Fraga

Resumen

Un algoritmo iterativo para obtener una reconstrucción tridimensional a partir de un conjunto de proyecciones de un espécimen, es:

- i) Se comparan las proyecciones con un modelo 3D de referencia para asignar los valores de orientación a cada proyección.
- ii) Se hace una reconstrucción 3D con el conjunto de proyecciones.
- iii) Se usa la reconstrucción generada en (ii) como modelo de referencia y se vuelve al paso (i) hasta llegar a la convergencia.

En este trabajo pretendemos investigar qué tan crítica resulta la selección del *modelo 3D de referencia* para que el algoritmo iterativo de reconstrucción converja.

Se usan modelos deformables basados en mallas de simplejos para realizar varios modelos iniciales de entrada al algoritmo iterativo de reconstrucción. Estos modelos iniciales van desde una esfera –que es un un modelo sin forma (todas sus proyecciones son iguales)–, hasta el objeto destino que es un fantasma de una macromoléculas biológica a media resolución. De este objeto destino generamos las proyecciones iniciales con las que iniciamos nuestro algoritmo iterativo de reconstrucción.

Las proyecciones y los especímenes usados por el algoritmo iterativo están especificados en pixels y voxels, respectivamente. Los modelos deformables están compuestos por politopos de *género 0*, y esto mismo resulta en una representación alternativa para una macromolécula biológica a mediana resolución.

Abstract

An iterative algorithm to obtain a three-dimensional reconstruction from a set of projections of a specimen is:

- i) The projections are compared with a reference 3D model to assign the orientation values for each projection.
- ii) With the set of projections with orientation values assigned the step (i), a 3D reconstruction is made.
- iii) The reconstruction generated in (ii) is used now as the reference model, and then return to the step (i) until to get convergence.

In this work we pretend to research how critic is the selection of the *3D reference model* in order to get convergence of the iterative reconstruction algorithm.

We use deformable models based in simplex meshes to construct the initial models used as input to the iterative reconstruction algorithm. These initial models start with a sphere –which is a formless model (all its projections are equals)–, and ends with the final model which is a biological macromolecule’s phantom at low resolution. With this phantom we generated the set of projections to initialize the iterative reconstruction algorithm. The projections and the specimens used by the iterative algorithm are specified in pixels and voxels, respectively. The deformable models are composed by polytopes of *genus* 0, and these give us an alternative representation for the low resolution biological macromolecule.

Agradecimientos

A mi familia, Carlos Ramírez Mijangos, Lidia Flores Meza y Carlos Ramírez Flores por criarme, alimentarme y educarme, sin ustedes no sería la persona que soy y esto no habría sido posible.

A mi asesor el Dr. Luis Gerardo de la Fraga y al Dr. Jorge Buenabad Chavéz por apoyarme en los momentos que mas lo necesite.

Al Centro de Investigación y Estudios Avanzados de Instituto Politécnico Nacional por darme la facilidades para los estudios y elaboración de esta tesis de Maestría.

A mis amigos y compañeros de la sección de Computación por enriquecer mi vida de formas y maneras tan interesantes.

A la sección de Computación en general.

Y a la vida. . .

Índice general

Lista de Figuras	VIII
Lista de Tablas	XI
1. Introducción	1
1.1. Reconstrucción tridimensional de macromoléculas biológicas	1
1.1.1. Proteínas	1
1.1.2. Resolución	2
1.2. Métodos de reconstrucción	3
1.3. Un algoritmo iterativo para una reconstrucción tridimensional	5
1.4. Modelos deformables	7
1.4.1. Modelos continuos	7
1.4.2. Modelos discretos	9
2. Marco Teórico	13
2.1. Las mallas de simplejos	13
2.1.1. Simplejos	13
2.1.2. Mallas y celdas de simplejos	14
2.1.3. Geometría de las mallas de simplejos	14
2.1.4. Geometría de las malla-2 de simplejos	16
2.1.5. Esquema de visualización para mallas de simplejos.	18
2.2. Modelos iniciales	22
2.2.1. Creación de una esfera con mallas de simplejos	22
2.2.2. Creación de un cilindro	30
2.2.3. Creación de un toroide	31
2.3. Esquema de deformación	36
2.3.1. Computación de la fuerza interna	36
2.3.2. Computación de la fuerza externa	38
2.3.3. Computación de la fuerza externa en rangos de datos	38
2.3.4. Computación de la fuerza externa en imágenes volumétricas	39
2.4. Algoritmo de aproximación	40
3. Reconstrucción 3D Iterativa con Distintos Modelos Iniciales	47
3.1. Generación del modelo de una macromolécula biológica	47
3.2. Deformación sucesiva del modelo de la macromolécula	48
3.3. Generación del volumen a partir de la malla deformada	53
3.4. Pruebas con el algoritmo iterativo de reconstrucción 3D	54

3.4.1. Pruebas	57
4. Conclusiones	73
4.1. Mallas de simplejos	73
4.2. Pruebas al algoritmo iterativo de reconstrucción 3D	74
A. Demostraciones	77
A.1. Ángulo del simplejo	77
A.2. Función $L(r_i, d_i, \varphi_i)$	79
A.3. Coordenadas baricéntricas	80
A.3.1. Posición de un punto a partir de dos vecinos	81
A.3.2. Posición de un punto a partir de tres vecinos	82
A.4. Determinar el círculo que pasa a través de tres puntos en el espacio	83
A.5. Determinar la proyección de un punto dentro de un plano.	85
A.6. Inversión	87
A.7. Inversión en el espacio	89
A.8. Ángulo de la recta resultante al círculo invertido (dirección del vector perpendicular u_i a la recta resultante)	91
A.9. Distancia de un punto en el círculo a la curva invertida	93
B. Lista de Programas Realizados.	95
C. Algoritmo para el Trazo de una Línea en Tres Dimensiones	97
D. Poliedros	101

Índice de figuras

1.1.	Niveles de organización de una estructura proteica. La propia secuencia de aminoácidos recibe el nombre de <i>estructura primaria</i> . Las interacciones entre fragmentos contiguos de la secuencia forman estructuras localmente regulares; es lo que se conoce como <i>estructura secundaria</i> . La <i>estructura terciaria</i> está constituida por el empaquetamiento de dichos elementos estructurales en unidades más o menos compactas denominadas <i>dominios</i> . Finalmente, la proteína en su estado final puede contener varios dominios ensamblados en la <i>estructura cuaternaria</i>	2
1.2.	Un contorno formado alrededor de varias caras dentro de una malla de simplejos. Como se puede apreciar, en el vértice P_1 la función $PP_1(0)$ hará referencia al vértice P_2 y la función $PP_1(1)$ hará referencia al vértice P_0	10
1.3.	(a) Triangulaciones, (b) modelo elástico de masa.	10
1.4.	Las mallas de simplejos tienen una dualidad topológica con las triangulaciones.	11
2.1.	Simplejos de pertenecientes a distinto número de dimensiones.	14
2.2.	Celdas a diferente p	14
2.3.	(a) Definición de los vectores tangente normal al rededor del vértice P_i de una malla-1 de simplejos. (b) Definición de la curvatura κ_i y el círculo circunscrito	15
2.4.	(a) La esfera circunscrita S_2 de radio R_i y el círculo circunscrito S_i de radio r_i . (b) La proyección de la figura (a) en el plano formado por los puntos (O_i, C_i, P_i) . El ángulo del simplejo puede ser interpretado como un ángulo de geometría plana.	16
2.5.	El centro de inversión P_i transforma la esfera en un plano	17
2.6.	La proyección F_i dentro del plano formado por los tres vecinos de P_i	18
2.7.	Estructura de una malla de simplejos	20
2.8.	Los elementos de un mismo tipo se encuentran doblemente ligados entre si.	20
2.9.	Mallas de simplejos trazados a mano sobre un triángulo, un cuadrado y un pentágono. Detalles de como se generaron los vértices en el texto.	23
2.10.	Malla de simplejos dibujada sobre un pentágono regular. Se han numerado las circunferencias concéntricas con el índice i del procedimiento 1. Para la figura, $f = 5$, $n = 5$ (ó $c = 3$)	25
2.11.	Prisma pentagonal. La esfera se crea expandiendo cada cara del prisma y llenando el espacio con hexágonos.	26
2.12.	Los vértices ya seleccionados para construir una semiesfera. Para la figura, $f = 3$ y $c = 3$	27

2.13. Diagrama lateral de una esfera. El número de capas es de 3, por lo que el número de circunferencias, n , es de 5; θ será igual a $\pi/4n$	27
2.14. Ejemplo de las aristas creadas para una esfera. La unión de las dos semiesferas se muestra en gris.	28
2.15. Un cilindro construido según el procedimiento 7. El vértice a , que está en la base, solamente puede unirse a los vértices arriba de él que están marcados con un punto negro, para poder formar un simplejo. Las líneas punteadas indican la última circunferencia que debe ser la misma que la base para poder formar un toro.	33
2.16. Esquema para visualizar como se dobla un cilindro para realizar el toroide. Las circunferencias c'_i del toro corresponden a las c_i del cilindro. La circunferencia c_4 del cilindro no se usa, es la misma que la c_0 . El sentido del eje y es hacia dentro de la página.	34
2.17. Ejemplo de la construcción de un toro.	35
2.18. La vecindad $Q^{si}(P_i)$ alrededor del vértice P_i	38
2.19. La función de rigidez $G(x)$	39
2.20. (a) El modelo de una estrella del cual se muestran algunas rebanadas. (b) La representación 3D del volumen presentado en (a).	41
2.21. Algoritmo de aproximación.	42
2.22. Gráfica de magnitud de fuerzas contra número de iteraciones.	43
2.23. Iteraciones 0 a 16	44
2.24. La figuras muestran los modelos correspondientes a la gráfica presentada en la figura 2.22, basado en el modelo presentado en la seccion 3.1 del capítulo 3. Como se puede apreciar todo el proceso de reconstrucción se lleva a cabo entre las iteraciones 0 y 20, después el modelo mantiene una forma constante.	45
3.1. Modelo generado con el contenido del archivo <i>esp1.dat</i>	48
3.2. La visualización del Espécimen 1 por medio de rebanadas.	48
3.3. Pantalla de la herramienta Proto.	49
3.4. (a),(b) y (c) modelos de la esfera a baja, media y alta resolución. (d),(e) y (f) modelos del cilindro a baja, media y alta resolución.	50
3.5. El espacio de trabajo discretizado, cada cubo dentro de éste es un voxel.	53
3.6. El proceso de discretizacion de la malla.	54
3.7. (a) El modelo intermedio obtenido en la iteración 12 de la herramienta "Proto".(b) La imagen volumétrica obtenida en (a) visualizada con la herramienta "Scubes".	55
3.8. Los distintos modelos a los que ha sido sometido el algoritmo de reconstrucción 3D.	57
3.9. Gráfica derivada de la tabla 3.3.	58
3.10. Modelos intermedios obtenidos de las iteraciones al aplicar el algoritmo de reconstrucción basado en mallas de simplejos.	63
3.11. Gráfica derivada de la tabla 3.11.	67
3.12. Modelos intermedios obtenidos de las iteraciones al aplicar el algoritmo de reconstrucción basado en mallas de simplejos.	71
3.13. Gráfica derivada de la tabla 3.19.	71
3.14. Modelos intermedios obtenidos de las iteraciones al aplicar el algoritmo de reconstrucción basado en mallas de simplejos.	72

4.1. Las figuras a,b y c muestran como puede propagarse el error de una normal la cual es muy próxima al centro del modelo (a y b), al ser atraído este vértice a una parte del modelo la cual originalmente no debería provoca que la normal (en la figura c) se aleje más del centro, generando con esto en algunos casos que no tenga un punto de atracción en la dirección, positiva o negativa de su normal. En la figura d se muestra un punto con la dirección de su vector normal adecuada. 75

A.1. El ángulo $\angle\varphi_i$ se calcula por medio de las relaciones entre los dos triángulos isósceles formados por P_i y sus vecinos. 77

A.2. La función $L(r_i, d_i, \varphi_i)$ nos devuelve la magnitud de la distancia entre la proyección del punto P_i , que es F , y el mismo punto P_i 79

A.3. La definición del punto b a partir de los puntos a y c 81

A.4. La definición del punto e a partir de los puntos a , b y c 82

A.5. Los planos formados por los tres puntos p_1 , p_2 y p_3 y su círculo circunscrito con centro C 84

A.6. La proyección f del punto p_i en el plano formados por los 3 puntos p_1, p_2, p_3 . 86

A.7. La coordenada P , su círculo de inversión y su inversa P' 87

A.8. La coordenada P , su esfera de inversión y su inversa P' 89

A.9. Definición de nuestro problema: Queremos saber cual es el ángulo de la pendiente m de la recta Q 92

A.10.El círculo de inversión c_i , el círculo a invertir C y la curva invertida L . . . 94

C.1. La diferencias de una recta trazada en el espacio tridimensional. 99

D.1. Algunos poliedros simples con los valores de V , E y F 101

D.2. Un poliedro con hoyos donde se puede ver las características de la generalización de la formula de Euler. 102

Índice de cuadros

3.1. Magnitudes de las fuerzas totales, el error absoluto y el porcentaje de similitud.	51
3.2. Porcentaje de similitud entre el modelo inicial y el final, según el número de iteración. El porcentaje de similitud fue calculado según las ecuaciones (3.1) y (3.2).	52
3.3. Porcentaje de similitud entre el modelo inicial y el final del modelo con simetría cuatro.	59
3.4. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial esfera.	60
3.5. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial 1.	60
3.6. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial 2.	61
3.7. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial 3.	61
3.8. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial 4.	61
3.9. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial 5.	63
3.10. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo original.	63
3.11. Porcentaje de similitud entre el modelo inicial y el final del modelo con simetría tres.	64
3.12. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial esfera.	65
3.13. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial 1.	65

3.14. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial 2.	65
3.15. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial 3.	66
3.16. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial 4.	66
3.17. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial 5.	66
3.18. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo original.	66
3.19. Porcentaje de similitud entre el modelo inicial y el final del modelo sin simetría.	68
3.20. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 1297 en el modelo inicial esfera.	69
3.21. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 1297 en el modelo inicial 1.	69
3.22. Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 1297 en el modelo inicial 2.	71
4.1. Tabla referente a la primera iteración del algoritmo de reconstrucción 3D, para cada modelo intermedio de cada espécimen.	76

Capítulo 1

Introducción

1.1. Reconstrucción tridimensional de macromoléculas biológicas

Los seres vivos poseen en su composición estructuras internas intrincadas que contienen varias clases de moléculas complejas. Cada una de las partes componentes de un organismo vivo cumple un propósito o función específicos, por lo tanto, es natural preguntarse cual es la función (o funciones) de una molécula determinada. Las principales macromoléculas de la célula son los lípidos, los polisacáridos, las proteínas y los ácidos nucleicos. Todas ellas poseen estructuras complejas que les confieren unas propiedades características, habilitándolas para desempeñar la mayor parte de las funciones más importantes de la célula. Esta estructura es, a su vez, la encargada de controlar la interacción entre una macromoléculas y otra, o entre las distintas partes de una misma macromolécula [1].

1.1.1. Proteínas

Uno de los tipos de macromoléculas principales dentro de la célula son las proteínas. Las células están formadas en gran parte por proteínas y son estas las que determinan en gran medida la forma y la estructura de la célula, y en última instancia, la función celular.

Una de las funciones más importantes de las proteínas consiste en actuar como enzimas, catalizando reacciones químicas específicas. Otra función interesante es la de desempeñar el papel de elementos estructurales, proporcionando mecanismos de regulación de diversos procesos celulares. Las proteínas atienden a otras muchas funciones biológicas y son las más versátiles de las macromoléculas biológicas.

En principio, cualquier macromolécula proteica puede adoptar un número enorme de formas diferentes, llamadas conformaciones. Sin embargo, la mayoría de las proteínas de una célula se pliegan de forma estable adoptando una sola de estas conformaciones. Este comportamiento se debe a que casi toda la información que determina la conformación se halla contenida en la secuencia de *aminoácidos* (un tipo de molécula orgánica básica) que forma la proteína. Sin embargo, a pesar de este hecho, aún no se sabe predecir con exactitud la estructura tridimensional detallada de una proteína cuya secuencia de aminoácidos se conozca. Por consiguiente, la conformación solo puede ser determinada mediante los diversos métodos biofísicos de determinación de estructuras: cristalografía

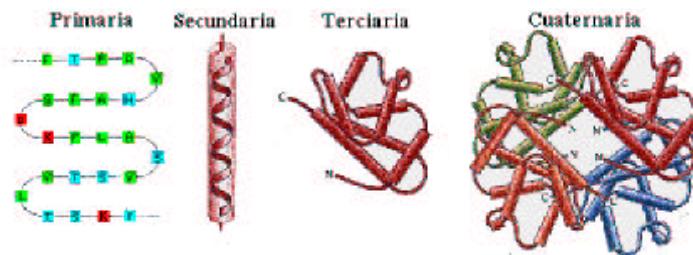


Figura 1.1: Niveles de organización de una estructura proteica. La propia secuencia de aminoácidos recibe el nombre de *estructura primaria*. Las interacciones entre fragmentos contiguos de la secuencia forman estructuras localmente regulares; es lo que se conoce como *estructura secundaria*. La *estructura terciaria* está constituida por el empaquetamiento de dichos elementos estructurales en unidades más o menos compactas denominadas *dominios*. Finalmente, la proteína en su estado final puede contener varios dominios ensamblados en la *estructura cuaternaria*.

de rayos X, resonancia magnética nuclear (RMN) y microscopía electrónica. Aunque el número de maneras diferentes en que una proteína se podría plegar es sumamente elevado, parece que existe una cierta lógica en la forma en que se produce este plegamiento: Existen cuatro niveles de plegamiento (véase la figura 1.1), y cada uno de ellos se construye de manera jerárquica a partir del anterior. La función biológica de una proteína depende en gran medida de las propiedades físicas y químicas de su superficie. La superficie de la estructura de la proteína es la que le confiere su especificidad al momento de interactuar con otra(s) macromolécula(s). Como consecuencia de esta especificidad de interacción, la proteína tendrá una determinada función dentro de la célula. El interior de la proteína a menudo es importante únicamente en cuanto que proporciona a la superficie de la molécula la forma y rigidez apropiadas, aunque hay importantes ejemplos de macromoléculas complejas con una estructura interna muy rica, como es el caso de los ribosomas.

Es evidente que, gracias a un conocimiento de la estructura de las proteínas, o en general de las macromoléculas biológicas, se podría conocer mejor la función que se desempeña dentro de la célula.

1.1.2. Resolución

Existen muchas definiciones de resolución, pero una de las más empleadas es aquella en la que la resolución se declara como la mínima distancia entre dos puntos que permita aún distinguirlos como entes separados. El orden de resolución de interés en el estudio de las proteínas está localizado entre 1 y 30-50 Å. Una resolución de unos 30 Å proporciona la forma global de una proteína. A unos 7 Å se pueden observar ya algunas estructuras secundarias. Las cadenas de aminoácidos comienzan a distinguirse en resoluciones de 3-4 Å y, finalmente, a 1-2 Å se pueden localizar los átomos individuales. La cristalografía de rayos X llega a resolución atómica (1-3 Å) para proteínas que forman cristales con un orden intrínseco alto. En cuanto a la microscopía electrónica, ésta cubre un amplio espectro de rangos de resolución, desde aproximadamente los 100 Å para grandes complejos hasta la resolución atómica, aunque esta última solo en algunos casos muy particulares (la resolución alcanzada depende enormemente del tipo de espécimen, del método de prepa-

ración empleado, del microscopio, etc.). El hecho de que la microscopía electrónica cubra esta amplitud de resoluciones hace posible que se alcance un continuo en resolución en el estudio de las estructuras biológicas. La microscopía electrónica en ese sentido, es el enlace vital entre la microscopía óptica empleada en biología celular y la cristalografía de rayos X.

1.2. Métodos de reconstrucción

Para la obtención de información acerca de las macromoléculas se hace uso de micrografías (fotografías tomadas con el microscopio electrónico), estas micrografías son tomadas con el objetivo de obtener medidas cualitativas, las cuales son por mencionar algunas: forma general, dimensiones, presencia de un componente dado, etc. Al ser estas fotografías o imágenes de un espécimen a estudiar no proporcionan directamente información acerca de la estructura tridimensional del espécimen, debido a que el tipo de imágenes que proporciona son imágenes, las cuales son proyecciones de un volumen, y es necesario la aplicación de técnicas de análisis de imagen para recuperar la tercera dimensión. Es posible reconstruir la estructura 3D de un objeto a partir de una serie de imágenes de proyección tomadas a diferentes ángulos de inclinación y ejes de rotación. La teoría matemática que sustenta esta afirmación fue desarrollada por Radon en 1917, y a partir de ella se han desarrollado algoritmos de reconstrucción en distintos campos de aplicación, tales como tomografía y radiología en medicina, radioastronomía y microscopía electrónica. El teorema de Radon afirma que en una función real en un espacio euclidiano n -dimensional que obedece a ciertas condiciones de regularidad esta definida unívocamente por sus integrales a lo largo de todos sus hiperplanos de dimensión $n - 1$. Esto significa que una función definida en tres dimensiones (el volumen) puede ser reconstruida a partir de sus proyecciones bidimensionales.

Los algoritmos de reconstrucción 3D de un objeto a partir de sus proyecciones 2D se han agrupado en dos grandes familias[1]:

- *Métodos directos*: Los métodos directos utilizan la relación matemática entre la densidad del objeto y sus proyecciones, y por lo tanto trabajan en el espacio real. Esta relación puede formularse como un sistema de ecuaciones lineales pero en la práctica el número de variables es tan grande que imposibilita la solución directa a este problema. Se utilizan por lo tanto, técnicas de aproximación iterativas, que tratan de llegar a una solución mediante la minimización de la discrepancia entre las proyecciones de un volumen estimado (que se actualiza en cada iteración) y las proyecciones experimentales.
- *Métodos de Fourier*: Estos métodos hacen uso del *Teorema de la Sección Central*. Este teorema establece que la transformada de Fourier de una imagen de proyección 2D es exactamente una sección central de la transformada de Fourier 3D del objeto. Según este teorema, podemos construir la transformada de Fourier 3D del objeto mediante el ensamblaje de las distintas secciones centrales que proponen las transformadas de Fourier de un número de proyecciones 2D del objeto. Finalmente, la distribución 3D de densidad del objeto puede obtenerse calculando la transformada inversa de Fourier 3D.

Dentro de los métodos que utilizan el Teorema de la Sección Central está el Método de Búsqueda de la Orientación usando conjuntos de proyecciones u *OSSP* por sus siglas en ingles [3], este método se utiliza varios atajos del método de fusión, el cual implica hacer una búsqueda entre pares de reconstrucciones efectuadas en el espacio real, este método está restringido para datos con gran inclinación, y los ángulos de Euler encontrados son inevitablemente parciales por la ausencia parcial de información angular. El problema de la determinación de la orientación en la reconstrucción de partículas no simétricas se resuelve de forma automática al aplicar el esquema de recolección de datos cónico-aleatorio. Este provee todos los ángulos de Euler necesarios para la reconstrucción en 3D. Dos de estos ángulos son determinados con gran precisión al aplicar geometría inclinada y el tercero se encuentra a través de la alineación de partículas extraídas de micrografías de especímenes sin inclinación. La aportación importante de esta aproximación se enfoca a que solo proyecciones muy similares son comparadas y el promedio resultante tiene un *SNR* (Relación Señal-Ruido) alta, lo cual facilita la determinación del tercer ángulo de Euler con una gran precisión. Las reconstrucciones obtenidas por este método son por lo tanto limitadas por la región angular perdida y se deben de realizar esfuerzos para llenar esta región usando más de una orientación. Por lo tanto el problema de la determinación de la orientación de la superficie puede ser reducido relacionando conjuntos de datos unos con otros. Este propósito se logra obteniendo la mejor concordancia entre dos conjuntos de proyecciones asumiendo que se conoce la geometría interna de cada conjunto. La recolección de datos cónico-aleatorios implica que las transformadas de Fourier de las proyecciones formen un conjunto de planos en el espacio de Fourier tangenciales al cono. De este modo, el problema de encontrar la orientación entre dos estructuras en 3D puede formularse como un problema de encontrar la orientación que coincida mejor entre dos conjuntos de planos tangenciales de Fourier a sus respectivos conos.

Otro método que se puede clasificar dentro de los métodos de Fourier es el *PFT* (o Transformadas Polares de Fourier)[2], el cual al igual que el *OSSP* parte de un modelo inicial en 3D. Este método es aplicado para modelos de virus icosaedricos, este comienza con un modelo 3D del cual se tienen m diferentes proyecciones para producir una base de datos de referencia. Para los icosaedros, estas vistas cubren una mitad de la estructura asimétrica de la unidad. Cada vista cartesiana (x, y) es interpolada en una rejilla polar (r, γ) , el cual subdivide los datos en series de anillos equi-espaciados desde $r = 0$ a un radio fuera del borde de la partícula. El ancho de cada anillo es igual al intervalo muestreado (tamaño del pixel) en los datos Cartesianos y todos los anillos son muestreados el mismo número de veces (dirección de γ) a un espacio lo suficiente grande para asegurarse que no ocurra perdida de información en valores de radio grandes. Los datos polares resultantes consistentes a los anillos organizado en renglones (de dirección r), están muestreados el mismo número de veces (dirección γ). Para cada una de las imágenes y proyecciones polares se transforman a través de la dirección azimutal para producir una base de datos de PFTs, para que cada PFT de las imágenes se correlacionen con esta base de datos. De esta forma la representación de la PFT son series simples de transformadas de Fourier de una dimensión o espectro de magnitud rotacional de cada anillo de datos. Este tipo de representación tiene la ventaja de ser invariante a la rotación, lo cual reduce considerablemente el número de subsecuente iteraciones requeridas para identificar los parámetros de orientación.

Para cada una de las imágenes PFT son correlacionadas por turnos con todas las PFT del modelo para identificar cual PFT del modelo es la que mejor coincide con cada imagen

PFT. Este proceso provee un estimado inicial de los valores de θ y $|\phi|$ para cada imagen. El ángulo de rotación ω , y el signo de ϕ , los cuales son consistentes con el modelo 3D, son derivados de las correlaciones rotacionales computadas entre la imagen polar j , la proyección del modelo $(\theta, +\phi, 0)_j$ y la proyección polar $(\theta, -\phi, 0)_j$. El conocimiento de $(\theta, \phi, \omega)_j$ permite un refinamiento más preciso del origen de la partícula vía la correlación cruzada translacional entre la imagen j y la proyección del modelo $(\theta, \phi, 0)_j$ después de que este es rotado por ω_j para obtener la proyección $(\theta, \phi, \omega)_j$. Con los orígenes y las direcciones de las vistas ya obtenidas para todas las n imágenes, una reconstrucción 3D puede ser computada por medio de procedimientos de Fourier-Bessel. Normalmente, solo aquellas imágenes que tienen la mejor correlación con los datos del modelo son usadas. Esta reconstrucción sirve como un nuevo modelo 3D para un ciclo adicional de refinamiento de los parámetros $\theta, \phi, \omega, x, y$. De manera alternativa uno puede usar los orígenes (x, y) determinados en el ciclo previo para computar cada imagen polar nueva, en lugar de recomputar un nuevo conjunto de orígenes (x_0, y_0) estimados. El refinamiento puede ser procesado calculando un nuevo conjunto de proyecciones del modelo en una rejilla fina de valores θ y ϕ .

El refinamiento es repetido hasta los incrementos en los coeficientes de la correlación no puedan ser obtenidos entre la imagen j y la proyección del modelo $(\theta, \phi, \omega)_j$, entre la PFT_j y la proyección $PFT(\theta, |\phi|)_j$ o entre la imagen polar j y la proyección polar $(\theta, |\phi|, 0)$. La restricción principal del método de la PFT es que se requiere un buen modelo 3D inicial.

1.3. Un algoritmo iterativo para una reconstrucción tridimensional

El esquema de comparación directa de las proyecciones desarrollado por *Penczek* [3] es iterativo: Las proyecciones de referencia (reproyecciones) son computadas a partir de una estructura prototipo (a baja resolución) de tal forma que cubran todo el espacio angular regularmente (equiespaciadamente). Como estructura prototipo se usa un volumen ya existente. Una proyección experimental dada es correlacionada con todas las proyecciones de referencia. El pico más grande de la Función de Correlación Cruzada (CCF) da los valores de los ángulos a la proyección. La CCF entre la proyección de referencia y la proyección actual da, al mismo tiempo, el desplazamiento y la orientación azimutal de la proyección de la partícula bajo consideración. Usando los nuevos parámetros para cada proyección experimental, una nueva reconstrucción es calculada, la cual normalmente tiene una resolución mejorada. La nueva reconstrucción es usada como un nuevo prototipo.

van Heel [4] propuso un método, en 1984, antes de tener la velocidad y recursos de las computadoras actuales, para obtener la orientación de una proyección usando la siguiente secuencia:

- Paso 0: Asignar orientaciones aleatorias a las proyecciones.
- Paso 1: Calcular la reconstrucción 3D.
- Paso 2: Proyectar la reconstrucción 3D en todas las direcciones en el espacio para comparar las proyecciones experimentales. Los parámetros a los cuales se ajustan mejor las comparaciones resultan en las nuevas orientaciones para las proyecciones;

calcular las diferencias entre las proyecciones del modelo actual y las proyecciones experimentales; si la suma del cuadrado de las diferencias es más grande que un valor predefinido entonces VOLVER al paso 1, de otro modo PARAR.

Este procedimiento contiene los ingredientes principales del algoritmo de Penczek et al., excepto por el cambio en el punto de arranque (paso 0), lo que resulta en la práctica es que el algoritmo encuentra difícilmente una solución satisfactoria excepto en el caso, totalmente fortuito donde el asignamiento aleatorio de los ángulos genere una distribución cercana a los valores correctos. En todos los otros casos, la reconstrucción inicial no se parecerá a una versión en baja resolución de la estructura real, y fallará en conducir de los nuevos asignamientos de orientación hacia la dirección correcta.

El algoritmo que se someterá a pruebas durante esta tesis esta basado en el algoritmo de van Heel, es el siguiente:

- Se debe tener un conjunto de proyecciones bidimensionales que hayan sido tomadas de una macromolécula; estas proyecciones las conoceremos como las *proyecciones experimentales*. También Se tiene de un modelo 3D de referencia (el prototipo).
1. Se proyecta el modelo 3D de referencia en todas direcciones en el espacio, a las proyecciones de referencia generadas se les llama *reproyecciones*.
 2. A las proyecciones experimentales se se les asigna los parámetros de translación y orientación comparándolas con las reproyecciones.
 3. Se realiza la reconstrucción en 3D con el conjunto de proyecciones experimentales con los parámetros ya asignados.
 4. La reconstrucción 3D generada en el punto anterior, que ya es generada con las proyecciones experimentales, se usa ahora como el prototipo.
 5. Calculamos las diferencias entre las proyecciones del modelo actual y las proyecciones experimentales, si la suma del cuadrado de las diferencias es más grande que un valor predefinido, se vuelve al paso 2.

En esta tesis se investigará que tanto depende la convergencia del algoritmo iterativo de la elección del modelo 3D de referencia, o del prototipo inicial.

Se pueden utilizar tres métodos distintos para realizar las comparaciones de las proyecciones experimentales con las reproyecciones:

- La función de correlación cruzada rotacional [3]
- Transformadas de Radon: Compara la transformada 2D de la proyección experimental con la transformada 2D de Radon de la proyección de referencia [5].
- Transformada Polar de Fourier (PFT): Se calcula la representación polar de las proyecciones del modelo de referencia y del modelo experimental, después se les calcula la transformada de Fourier, y finalmente se comparan [2].

1.4. Modelos deformables

Un modelo o superficie deformable es un objeto compuesto de la representación de superficie y las leyes de evolución que permiten la deformación de esta superficie para representar una forma diferente. Como se menciona en [6] los modelos deformables se pueden clasificar por medio a la naturaleza de su representación en dos categorías: *discretos* y *continuos*. En las representaciones discretas la geometría de la superficie es solamente conocida en un conjunto finito de puntos. Las representaciones continuas deben ser discretizadas para su implementación computacional pero ofrecen la habilidad de computar cantidades diferenciales tales como normales o curvaturas en casi cualquier parte de la superficie.

1.4.1. Modelos continuos

Los modelos continuos se clasifican por la ecuación por medio de la cual son representados, esta ecuación puede ser explícita o implícita y a continuación se definirá en que consiste cada una de estas representaciones.

Representación explícita

Cuando se utiliza una representación explícita, las superficies son descritas por funciones coordinadas, las cuales dependen de los parámetros de un vector de forma q como se ve en la ecuación 1.1.

$$\begin{aligned} \mathcal{S}_q : \quad \mathbb{R}^{n_q} \times \Omega &\rightarrow \mathbb{R}^3 \\ (q_1, \dots, q_{n_q}, r, s) &\mapsto p_q(r, s) \end{aligned} \quad (1.1)$$

En donde una superficie deformable \mathcal{S}_q es controlada por un vector con parámetros de forma: $q = (q_1, \dots, q_{n_q})$, el cual puede expresarse como $p_q(r, s)$, en donde los parámetros r y s son parámetros de superficie, $r \in [0, 2\pi]$ y $s \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Una distinción más a fondo entre estos modelos puede hacerse dependiendo del número y la naturaleza del conjunto de parámetros q . El número de parámetros tiene un impacto en la complejidad de la deformación. Otro criterio es el impacto de la deformación local o global de estos parámetros. Con parámetros controlando la deformación local se pueden obtener formas bastante complejas pero a costa de una complejidad computacional importante. De manera contraria, los parámetros de forma global nos dan un esquema numérico más estable cuando deben estimarse un número limitado de parámetros.

Dentro de los modelos que utilizan la representación explícita tenemos los siguientes:

- *Funciones de soporte polinomiales finitas*: En estas representaciones, los parámetros de forma corresponden a los coeficientes de una función de soporte polinomial finita. Cada parámetro tiene sólo un impacto local en la forma de la superficie. Esta categoría incluye los diferentes tipos de splines de contorno y superficie de los cuales los más usadas son las B-splines.
- *Superquadrics*: Los superquadrics son una generalización de los quadrics a superficies. Estos son ciertamente las superficies parametrizadas de uso más frecuente, en particular los superelipsoides representando superficies cerradas. Un superelipsoide

es definido por la ecuación:

$$Q_q(r, s) = \begin{cases} a_1 \operatorname{signo}(\cos r \cos s) |\cos r \cos s|^{\epsilon_1} \\ a_2 \operatorname{signo}(\cos r \operatorname{sen} s) |\cos r \operatorname{sen} s|^{\epsilon_2} \\ a_3 \operatorname{signo}(\operatorname{sen} s) |\operatorname{sen} s|^{\epsilon_3} \end{cases} \quad (1.2)$$

donde $q = (a_1, a_2, a_3, \epsilon_1, \epsilon_2, \epsilon_3)^T$ es el vector de parámetros de superficie, $r \in [0, 2\pi]$ y $s \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Los superquadrics permiten representaciones de formas relativamente variables, pero solo con ejes simétricos. Debido a su formulación primitiva estos no pueden representar formas complejas.

- *Descomposición modal:* Diferentes representaciones usan una descomposición modal del modelo. La descomposición base es un conjunto de diferentes armónicas de frecuencia. Esta representación es equivalente a un conjunto de superficies paramétricas de los cuales los parámetros son los valores de la moda. La suma de las primeras modas que componen la superficie nos da una aproximación burda de su forma. La aproximación se refina cuando las modas de alta frecuencia son sumadas a la representación. En la práctica es recomendable que se reduzca al mínimo el número de modas para obtener una representación compacta, sabiendo que unas cuantas modas nos permiten la representación de formas complejas.

Representación implícita

Contrarios a los modelos con representación explícita, la representación implícita involucra una ecuación implícita para localizar los puntos de la superficie. Una superficie implícita es definida de forma general como el conjunto de ceros de una función f evaluada en \mathbb{R} que es:

$$\mathcal{S} = \{p \in \mathbb{R}^3 | f(p) = 0\} \quad (1.3)$$

Algunas de la representaciones implícitas más comúnmente usadas son:

- *Superficies algebraicas:* Las superficies algebraicas son superficies que satisfacen la definición dada por la ecuación 1.3 donde f es un polinomio. Estas han sido ampliamente utilizadas en el campo de gráficas y modelado. Las superficies algebraicas tienen varias limitaciones, estas no necesariamente son cerradas, aunque es posible considerar solo algunos polinomios de orden par. Las superficies algebraicas no están listas para desplegarse en una pantalla gráfica. Aparte, resulta difícil computar la distancia de un punto a una superficie algebraica, lo cual que es requerido por algoritmos de correspondencia de superficies. El valor de $f(p)$ es a veces usado como una aproximación.
- *Superquadrics:* los superquadrics presentados en la sección 1.4.1 pueden ser representados por medio de la forma implícita, en particular los superelipsides con la ecuación implícita:

$$F_q(p) = \left(\left(\left(\frac{p_x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{p_y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{p_z}{a_3} \right)^{\frac{2}{\epsilon_1}} \right)^{\frac{\epsilon_1}{2}} = 1 \quad (1.4)$$

representando superficies cerradas.

- *Hiperquadrics*: Los hiperquadrics fueron propuestos como una extensión de los superquadrics. Estos permiten la representación de una extensa familia de superficies. Un hiperquadric es definido por la ecuación:

$$f_q(p) = \sum_{j=1}^{n \geq 3} |a_j p_x + b_j p_y + c_j p_z + d_j|^{\epsilon_j} = 1 \quad (1.5)$$

donde $\epsilon_j \geq 0$ para toda j , y $q = (a_1, b_1, c_1, \epsilon_1, \dots, a_n, b_n, c_n)^T$. Esta ecuación no tiene alguna representación explícita cuando $n > 3$. La superficie es encerrada en la cerradura convexa definida por el conjunto de planos $a_j p_x + b_j p_y + c_j p_z + d_j = \pm 1$. Cuando el coeficiente ϵ_j se incrementa la superficie se fusiona con su cubierta convexa. Así como la representación modal, los hyperquadrics son una formulación extensiva permitiendo la adición de términos de forma iterativa que refinan la descripción de la superficie. Sin embargo un hyperquadric debe permanecer homomorfo a una esfera.

1.4.2. Modelos discretos

La representación de modelos deformables como entidades discretas se basa naturalmente en el problema de caracterización de las representaciones continuas. De cualquier forma esta aproximación permite un gran número de grados de libertad a la superficie y por lo tanto a menudo es necesario forzar el espacio de posibles deformaciones. Los modelos discretos se pueden clasificar por su naturaleza en dos tipos: *mallas discretas* y *sistemas de partículas*.

Mallas discretas

Las mallas discretas se definen como un conjunto con alguna relación de conectividad, entre las más comúnmente usadas están:

- *Contornos Discretos*: Un contorno deformable discreto \mathcal{C} es un conjunto de vértices $\{p_0, \dots, p_{d-1}\}$ conectados para producir una línea poligonal cerrada o abierta. Hagamos $PP_i : \{0, 1\} \rightarrow [0, d-1]$ ser una relación de vecindad del vértice i :

$$\begin{cases} \forall i \in [1, d-2], PP_i(0) = i+1 \text{ y } PP_i(1) = i-1 \\ PP_0(0) = 1 \text{ y } PP_{d-1}(1) = d-2 \\ PP_0(1) = d-1 \text{ y } PP_{d-1}(0) = 0 \text{ si } \mathcal{C} \text{ es cerrada.} \end{cases} \quad (1.6)$$

Los contornos discretos son definidos como un conjunto de vértices con vecindades tal que un vértice dentro de un contorno tenga dos y solo dos vecinos que pertenezcan a ese contorno, los contornos pueden ser definidos al rededor de cualquier cara de una malla (vease la Fig. 1.2). Como se puede ver en el conjunto de propiedades definidas en 1.6 La función $PP_i(x)$ expresa una relación de vecindad la cual nos indica que vértice es el anterior o el siguiente del vértice i del contorno, en esta función x puede tomar dos distintos valores: 0 o 1, este valor dependerá de a que vértice deseamos hacer referencia, al vértice a la derecha del vértice i (x igual a 0) o al de la izquierda ($x = 1$), en donde las dos ultimas propiedades del conjunto de propiedades 1.6 son validas únicamente para contornos cerrados.

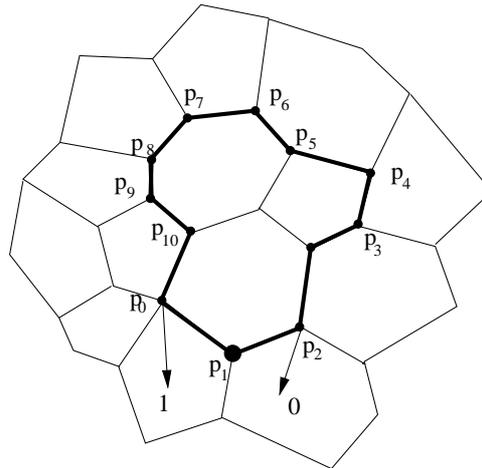


Figura 1.2: Un contorno formado alrededor de varias caras dentro de una malla de simplejos. Como se puede apreciar, en el vértice P_1 la función $PP_1(0)$ hará referencia al vértice P_2 y la función $PP_1(1)$ hará referencia al vértice P_0 .

- *Triangulaciones:* Una superficie puede ser también representada por un conjunto de vértices y una relación de vecindad que incluye dependencias topológicas. Una representación común son las triangulaciones, en donde la superficie está compuesta por un conjunto de triángulos adyacentes (véase la figura 1.3). Cada triángulo de una triangulación comparte al menos una de sus aristas con un triángulo vecino.

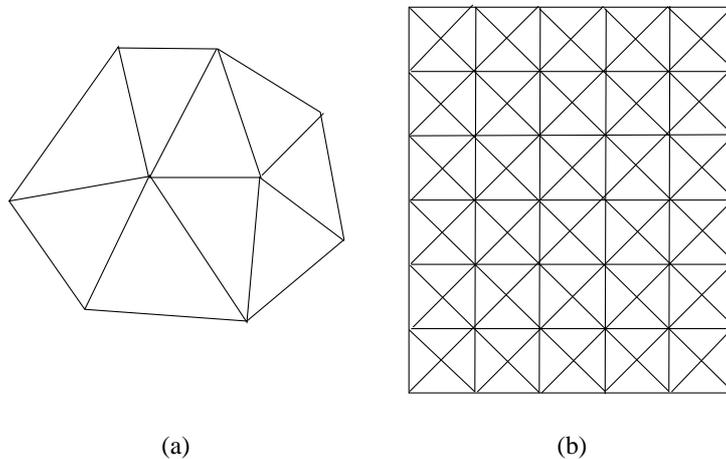


Figura 1.3: (a) Triangulaciones, (b) modelo elástico de masa.

- *Modelo elástico de masa:* Los modelos elásticos de masa están compuestos por un conjunto de nodos conectados por resortes en el contexto de reconstrucción de superficies. La longitud en reposo de los resortes determina la forma en reposo del modelo aunque los modelos elásticos de masa tienen varias posiciones de equilibrio. La consistencia de los resortes controla las propiedades elásticas del modelo. Es posible agregar resortes entre vértices que no son vecinos directos de forma tal que se mejore la estabilidad de la malla y para proveer un mejor comportamiento volumétrico.

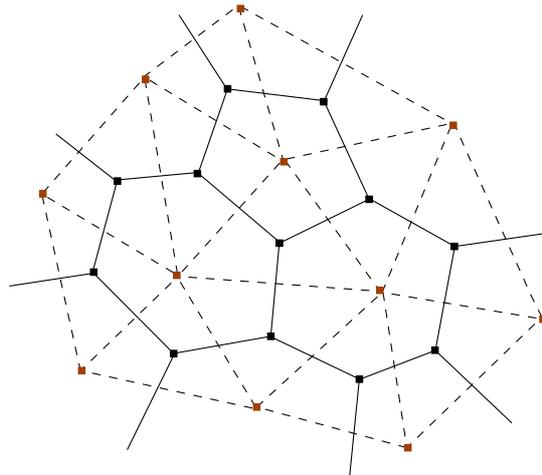


Figura 1.4: Las mallas de simplejos tienen una dualidad topológica con las triangulaciones.

- *Mallas de simplejos*: Una malla de simplejos es un modelo discreto con una conectividad de vértices constante. Las mallas-2 de simplejos son utilizadas para representar superficies. Cada vértice de una malla-2 de simplejos está conectado con tres y sólo tres vecinos. Las mallas-2 de simplejos son topológicamente duales a las triangulaciones, como se ilustra en la figura 1.4 de donde podemos notar que existe un triángulo dual por cada vértice de la malla, y el vértice de una triangulación dual por cada cara de la malla. Debido a su conectividad constante, la geometría de las mallas de simplejos es bastante simple. La noción de superficie local permite la definición de condiciones de memoria de forma.

Sistemas de partículas

Los sistemas de partículas son modelos discretos con una representación fundamentalmente diferente a las mallas discretas. Estos están compuestos por un conjunto de objetos elementales llamado partículas. Las partículas son descritas por su localización, velocidad, aceleración, masa y algún otro parámetro necesario para alguna aplicación en particular. Cada partícula se desarrolla acorde a las leyes mecánicas de la Física Newtoniana. Las partículas interfieren una con otras a través de las fuerzas de atracción y repulsión que tiende a ordenarlas en una rejilla regular (rejilla hexagonal) cuando las fuerzas externas se desvanecen. Los sistemas de partículas están diseñadas para representar sólidos viscoelásticos y fluidos. Los sistemas de partículas no permiten una computación directa de las propiedades geométricas de las superficies y son bastante complejas de visualizar.

Capítulo 2

Marco Teórico

En este capítulo se dará una introducción teórica a los *modelos deformables*. Es específico se desarrollarán tres temas:

1. Las mallas de simplejos.
2. La construcción tres cuerpos geométricos simples, la esfera, el cilindro y el toro, con mallas de simplejos. Estos cuerpos geométricos simples serán usados como *modelos iniciales* a los cuales se les induce una deformación para que alcancen otro modelo objetivo (que será una macromolécula biológica en baja resolución).
3. El esquema de deformación que se usa en las mallas de simplejos, y
4. El algoritmo de aproximación el cual es utilizado para obtener el modelo objetivo a partir de un modelo inicial.

2.1. Las mallas de simplejos

Esta sección se abocará a dar principios y definiciones de las mallas de simplejos, las explicaciones y demostraciones de carácter formal se darán en el apéndice A, al final de esta tesis.

2.1.1. Simplejos

Los simplejos son la generalización de una región tetraédrica en el espacio a n dimensiones. Un simplejo- k , de dimensión k está formado de $k + 1$ caras-0 (vértices), $k(k + 1)/2$ caras-1 (aristas) y $\binom{k+1}{i+1}$ caras- i , en donde $\binom{n}{k}$ es un coeficiente binomial (para ver una explicación más general de esta relación véase el apéndice D). Los simplejos se pueden representar por medio del símbolo $\underbrace{\{3, \dots, 3\}}_{n-1}$. En una dimensión el simplejo es una línea, en dos dimensiones el simplejo es la cubierta convexa de un triángulo equilátero. En tres dimensiones el simplejo $\{3, 3\}$ es la cubierta convexa de un tetraedro, tal como se aprecia en la figura 2.1, de tal forma que es posible apreciar cómo un simplejo-1 se forma a través de dos simplejos-0, un simplejo-2 se forma a partir de tres simplejos-1 y tres simplejos-0, obteniendo de esta forma una definición recursiva de simplejos.

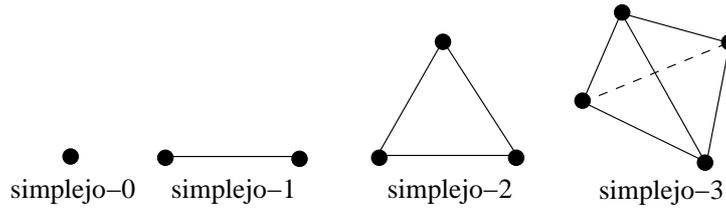


Figura 2.1: Simplejos de pertenecientes a distinto número de dimensiones.

2.1.2. Mallas y celdas de simplejos

Definiremos una malla- k de simplejos como la unión de varias celdas- p . Dado que estas celdas son a su vez mallas- p de simplejos, la definición de celda es recurrente:

Definición 1. Se define una celda-0 de \mathbb{R}^d como un punto P en \mathbb{R}^d , y una celda-1 de \mathbb{R}^d como una línea en \mathbb{R}^d , por ejemplo un par de vértices sin ordenar (P, M) . Definimos de forma recursiva una celda- p ($p \geq 2$) C en \mathbb{R}^d como una unión de celdas- $(p-1)$ tal que:

1. Cada vértice perteneciente a C pertenecerá a p distintas celdas- $(p-1)$.
2. La intersección de dos celdas- $(p-1)$ puede ser el vacío o una celda- $(p-2)$.
3. Teniendo dos vértices de C , debe existir un camino que ligue a ambos vértices.

Una celda-2 es por lo tanto un conjunto de líneas que tiene uno y solamente un vértice en común. Por lo tanto una celda-2 es una línea poligonal cerrada en \mathbb{R}^d . La tercera condición garantiza que una celda- p sea simple conectada.

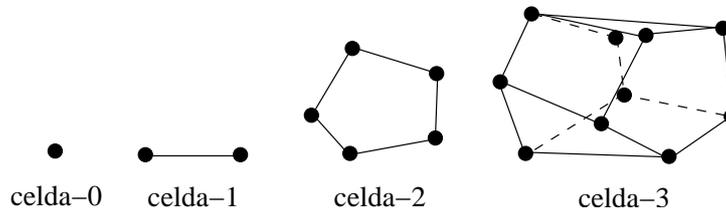


Figura 2.2: Celdas a diferente p

Definición 2. Una malla- k de simplejos \mathcal{M} en \mathbb{R}^d es una celda- $(k+1)$ de \mathbb{R}^d .

Una malla- k de simplejos es por lo tanto una unión de celdas- k que cumplen las propiedades de la definición 1.

Propiedad 1. Una malla- k de simplejos es una malla- $(k+1)$ conectada: cada vértice tiene $k+1$ vértices vecinos.

2.1.3. Geometría de las mallas de simplejos

En esta sección se presentaran las principales relaciones geométricas existentes en una malla-1 de simplejos en \mathbb{R}^2 y una malla-2 de simplejos en \mathbb{R}^3 . Teniendo la ecuación 2.4 como ecuación principal, y gracias a esta podemos obtener la posición de un vértice por medio de la posición de sus vecinos. También definiremos otras entidades geométricas tales como el ángulo del simplejo y sus parámetros métricos. La geometría plana de una malla-1 de simplejos $\mathcal{M} \in \mathbb{R}^2$ es descrita por el ángulo del simplejo y sus parámetros

métricos. Primero se definirá una tangente local t_i y un vector normal m_i al rededor del vértice P_i como:

$$t_i = \frac{\overline{P_{i-1}P_{i+1}}}{\|P_{i-1}P_{i+1}\|} \quad m_i = t_i^\perp \quad (2.1)$$

La curvatura local κ_i se define como la inversa del radio del círculo circunscrito a (P_{i-1}, P_i, P_{i+1}) , así como también definimos r_i como la distancia media entre P_{i-1} y P_{i+1} : $r_i = \frac{\|P_{i-1}P_{i+1}\|}{2}$. A continuación definiremos el ángulo del simplejo y los parámetros para P_i .

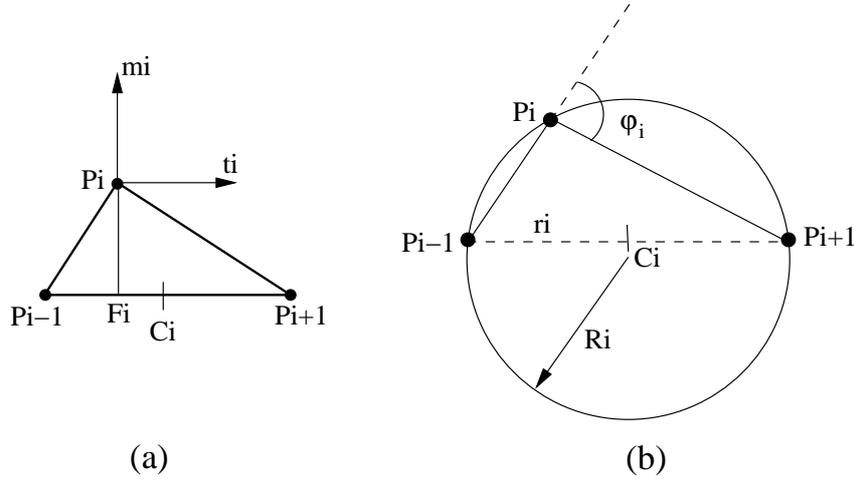


Figura 2.3: (a) Definición de los vectores tangente normal al rededor del vértice P_i de una malla-1 de simplejos. (b) Definición de la curvatura κ_i y el círculo circunscrito

Definición 3. Sea P_i un vértice de una malla-1 de simplejos y sea F_i la proyección ortogonal de P_i , dentro de la línea $[P_{i-1}P_{i+1}]$. Entonces los parámetros métricos en P_i , (ϵ^1, ϵ^2) son definidos por las coordenadas baricéntricas de F_i relativas a P_{i-1} y P_{i+1} :

$$\epsilon^1 P_{i-1} + \epsilon^2 P_{i+1} = F_i \quad (2.2)$$

$$\epsilon^1 + \epsilon^2 = 1 \quad (2.3)$$

Definición 4 El ángulo del simplejo φ_i en P_i es el ángulo orientado, que existe entre los dos segmentos adyacentes $[P_{i-1}P_i]$ y $[P_iP_{i+1}]$. La siguiente relación fundamental nos da la posición de vértice P_i en función de sus dos vecinos P_{i-1} y P_{i+1} y los tres parámetros $(\epsilon^1, \epsilon^2, \varphi_i)$:

$$P_i = \epsilon^1 P_{i-1} + \epsilon^2 P_{i+1} + L(r_i, |2\epsilon^1 - 1| r_i, \varphi_i) m_i \quad (2.4)$$

donde:

$$L(r_i, d_i, \varphi_i) = \frac{(r_i^2 - d_i^2) \tan \varphi_i}{\epsilon \sqrt{r_i^2 + (r_i^2 - d_i^2) \tan^2 \varphi_i} + r_i} \quad (2.5)$$

con:

$$\begin{aligned} \epsilon &= 1 & \text{si} & \quad \|\varphi_i\| < \frac{\pi}{2} \\ \epsilon &= -1 & \text{si} & \quad \|\varphi_i\| > \frac{\pi}{2} \end{aligned}$$

2.1.4. Geometría de las malla-2 de simplejos

En una malla-2 de simplejos $\mathcal{M} \in R^3$, definimos un vector normal n_i como:

$$n_i = \frac{P_{N_1(i)} \times P_{N_2(i)} + P_{N_2(i)} \times P_{N_3(i)} + P_{N_3(i)} \times P_{N_1(i)}}{\|P_{N_1(i)} \times P_{N_2(i)} + P_{N_2(i)} \times P_{N_3(i)} + P_{N_3(i)} \times P_{N_1(i)}\|} \quad (2.6)$$

La definición del ángulo del simplejo se generaliza de la definición dada en la sección 2.1.3. Si introducimos una esfera S_2 de centro O_i y radio R_i , circunscrita a los cuatro vértices $(P_i, P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$. El ángulo del simplejo $\varphi_i = \angle(P_i, P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$ en P_i es definido por las siguientes dos ecuaciones:

$$\varphi_i \in [-\pi, \pi] :$$

$$\text{sen } \varphi_i = \frac{r_i}{R_i} \text{signo}(P_i P_{N_1(i)} \cdot n_i) \quad (2.7)$$

$$\text{cos } \varphi_i = \frac{\|O_i C_i\|}{R_i} \text{signo}(O_i C_i \cdot n_i) \quad (2.8)$$

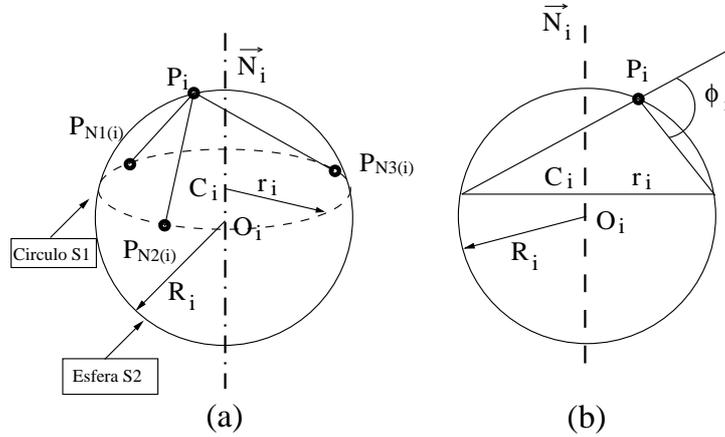


Figura 2.4: (a) La esfera circunscrita S_2 de radio R_i y el círculo circunscrito S_i de radio r_i . (b) La proyección de la figura (a) en el plano formado por los puntos (O_i, C_i, P_i) . El ángulo del simplejo puede ser interpretado como un ángulo de geometría plana.

Donde $\|O_i C_i\|$ representa la distancia al plano $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)},)$ desde el centro de la esfera O_i . Desde que se supone que los tres vecinos de cada vértice son linealmente independientes el círculo no se degenera ($r_i \neq 0$). El ángulo del simplejo es independiente de la posición de $P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)}$ en el círculo S_i y de P_i en el hemisferio de S_2 . Es nulo cuando P_i está en el plano definido por sus tres vecinos. Las relaciones mostradas en la ecuación 2.7 y 2.8 se vuelven explícitas al dibujar la proyección de la esfera en el plano

definido por (O_i, C_i, P_i) (como en la figura 2.4). En esta figura el ángulo del simplejo φ_i aparece como un ángulo en el sentido de geometría plana entre los segmentos que unen P_i a la proyección del círculo. Para una descripción detallada del ángulo del simplejo φ_i consultese el apéndice A.

La definición de φ_i dada en las ecuaciones 2.7 y 2.8 no es conveniente para computarse debido a que el radio de la esfera circunscrita R_i puede tomar valores muy altos cuando P_i se aproxime al plano definido por sus tres vecinos. En lugar de esto, éste puede ser fácilmente computado cuando se define a través de una inversión de centro P_i (para una descripción detallada véase el apéndice A). Desde que una inversión a través de P_i transforma cualquier esfera en un plano la esfera circunscrita es transformada en un plano pasando a través de los tres vecinos transformados (véase la figura 2.5). Si escribimos D_i^*

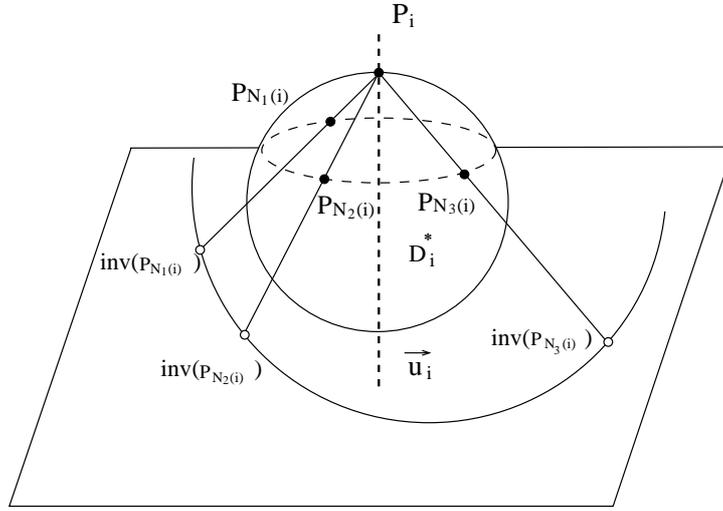


Figura 2.5: El centro de inversión P_i transforma la esfera en un plano

(véase apéndice A) como la distancia de P_i al plano transformado y \vec{u}_i (véase apéndice A) como el vector normal unitario a este plano, entonces tendremos:

$$D_i^* = \frac{1}{2R_i} \quad O_i = P_i + \frac{\vec{u}_i}{2D_i^*} \quad (2.9)$$

Las ecuaciones 2.7 y 2.8 se pueden escribir como:

$$\text{sen } \varphi_i = 2r_i D_i^* \text{signo}(\overrightarrow{P_i P_{N_1(i)}} \cdot \vec{N}_i) \quad (2.10)$$

$$\text{cos } \varphi_i = \|2r_i D_i^* \overrightarrow{P_i C_i}\| \text{signo}(\overrightarrow{O_i C_i} \cdot \vec{N}_i) \quad (2.11)$$

El ángulo del simplejo tiene muchas propiedades interesantes, es importante notar que hay una relación simple entre el ángulo del simplejo φ_i y la curvatura $H_i = \frac{1}{R_i}$, también llamada la *curvatura media* en el vértice P_i :

$$H_i = \frac{\varphi}{r_i} \quad (2.12)$$

Desde el vértice P_i introducimos la proyección F_i (véase apéndice A) de P_i en el triángulo formado por la vecindad $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$ (como se puede ver en la figura 2.6). Los

parámetros métricos en el vértice P_i son las coordenadas baricéntricas de F_i con respecto al triángulo $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$:

$$F_i = \epsilon^1 P_{N_1(i)} + \epsilon^2 P_{N_2(i)} + \epsilon^3 P_{N_3(i)} \quad (2.13)$$

$$\epsilon^1 + \epsilon^2 + \epsilon^3 = 1 \quad (2.14)$$

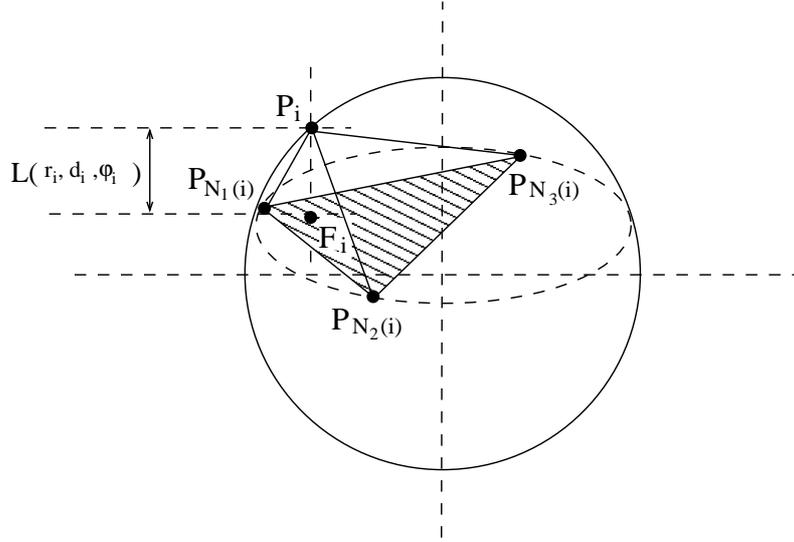


Figura 2.6: La proyección F_i dentro del plano formado por los tres vecinos de P_i .

El ángulo del simplejo con sus dos parámetros (debido a la igualdad mostrada en la ecuación 2.14) representa la posición de P_i con respecto a sus tres vecinos:

$$P_i = \epsilon^1 P_{N_1(i)} + \epsilon^2 P_{N_2(i)} + \epsilon^3 P_{N_3(i)} + L(r_i, d_i, \varphi_i) n_i \quad (2.15)$$

donde:

- r_i es el radio de círculo circunscrito en el triángulo $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$.
- d_i es la distancia entre F_i y el centro C_i del círculo circunscrito.
- $L(r_i, d_i, \varphi_i)$ es una función descrita en la ecuación 2.5, en la pág. 15 (para una explicación más detallada véase el apéndice A).

2.1.5. Esquema de visualización para mallas de simplejos.

Para poder visualizar en un sistema computacional a una malla de simplejos, se le tiene que indicar al sistema que parte de ésta se tiene que mostrar en pantalla, cuáles son los polígonos que forman a las caras de una malla, el orden que tienen. En general, un modelo en 3D está formado por medio de polígonos, estos polígonos a su vez están formados por caras, las caras están formadas por aristas, y finalmente las aristas están

definidas por vértices, y se debe de indicar cada uno de los componentes de la malla y que relación tienen entre sí. Teóricamente con tener la vecindad entre los vértices de una malla bastaría para presentar en la pantalla nuestro modelo, hacer esto nos llevaría a programar un algoritmo, el cual a parte de ser complicado, gastaría mucho tiempo maquina calculando y mapeando los vértices, las aristas y las caras de los polígonos que forman nuestra malla. Para evitar estas situaciones, Delingette [7] propuso en su tesis doctoral una serie de estructuras de datos, las cuales son útiles para visualizar los modelos que estén basados en mallas de simplejos. Estas estructuras representan las partes de un polígono, anteriormente mencionadas. En esta tesis hemos retomado esas estructuras y las hemos modificado a nuestra conveniencia para poder visualizar en pantalla modelos 3D que estén basados en mallas de simplejos. Como ya se mencionó anteriormente, las estructuras representan partes de un polígono, las cuales son:

- Malla.
- Caras.
- Aristas.
- Vértices (puntos).

estos elementos que forman polígonos tienen relaciones entre sí, las cuales son las siguientes:

- Cada vértice tiene tres vértices vecinos.
- Un vértice solo puede pertenecer a tres caras.
- Una arista está formada por dos vértices.
- Un vértice puede pertenecer solamente a tres aristas.
- Una cara puede tener n aristas pertenecientes a esta.
- Una malla puede estar formada por m caras

Las relaciones como se puede ver en la figura 2.7 son:

- Una Malla esta formada por polígonos.
- Los polígonos están formados por caras.
- Las caras están formadas por aristas.
- Las aristas están formadas por puntos (como podemos ver en la figura 2.7).

Los puntos serán nuestra unidad mínima, o nuestro elemento básico con el cual construiremos las mallas. Los elementos de un mismo tipo están guardados en listas doblemente ligadas, como se puede ver en la figura 2.8, por lo tanto todas las caras estarán dentro de una lista de caras, de manera similar, las aristas y los puntos están dentro de sus respectivas listas. Esto nos facilita la inserción de cualquiera de estos elementos dentro de su lista, agiliza las búsquedas de un elemento en particular, y la eliminación de un elemento también es más sencilla.

Las estructuras de datos utilizadas en esta tesis (creadas en lenguaje C) son:

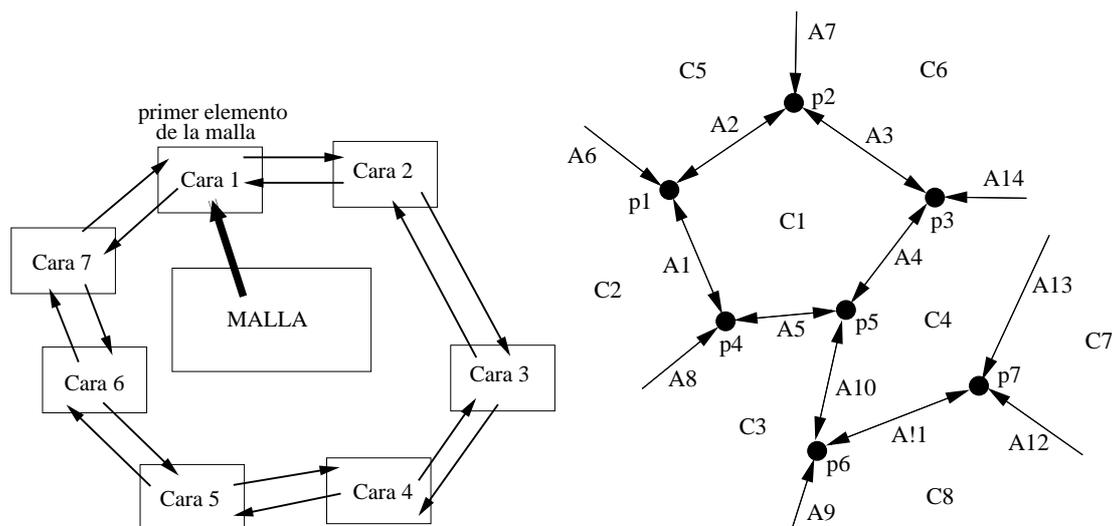


Figura 2.7: Estructura de una malla de simplejos

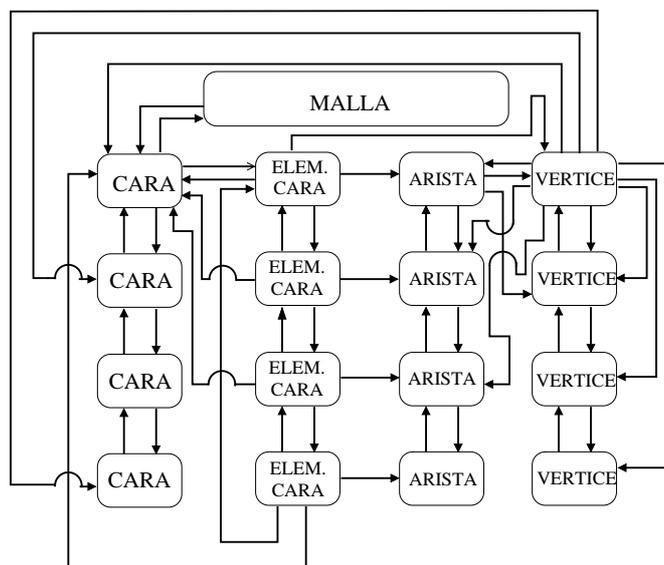


Figura 2.8: Los elementos de un mismo tipo se encuentran doblemente ligados entre si.

```

typedef struct _Malla
{
    struct _Punto *   pri_Punto; /* El primer elemento de la lista de puntos*/
    struct _Arista *  pri_Arista; /* El primer elemento de la lista de aristas */
    struct _Cara *   pri_Cara; /* El primer elemento de la lista de caras */
    double cet[3]; /* La coordenada del centro de la malla en */
                  /* coordenadas cartesianas en 3D*/
}*_Malla;

typedef struct _Punto
{
    struct _Punto *vecinos[3]; /* Los 3 vértices adyacentes */
    struct _Arista *arista[3]; /* Las 3 aristas adyacentes */
    struct _Cara *cara[3]; /* Las 3 caras adyacentes */
    int cual_vecino[3]; /* El índice de adyacencia del vértice para un */
                      /* vértice adyacente */
    double cp[3]; /* La coordenada del vértice en coordenadas */
                 /* cartesianas en 3D */
}

```

```

double norm[3];          /* La normal del vértice en coordenadas
                        /* cartesianas en 3D */
double _Di;             /* La distancia del vértice a sus 3 vértices */
                        /* vecinos transformados a través de la */
                        /* inversión de puntos */
double sin_phi;        /* El seno del ángulo de este vértice en */
                        /* particular al formar un simplejo con sus */
                        /* vértices vecinos */
double C[3];           /* El centro del círculo circunscrito a
                        /* nuestros 3 vecinos */
double r;              /* El radio del círculo circunscrito a */
                        /* nuestros 3 vecinos */
double F[3];          /* La proyección de este vértice en el plano */
                        /* formado por sus vecinos */
double Hi;            /* La curvatura de nuestro simplejo */
struct _Punto *ant;   /* El vértice que le precede en la lista de */
                        /* vértices */
struct _Punto *sig;   /* El vértice que le sigue en la lista de */
                        /* vértices */

}*Punto;

typedef struct _Arista
{
    struct _Punto *vecinos[2]; /* Los dos vértices adyacentes */
    struct _Cara *cara[2];    /* Las dos caras adyacentes */
    struct _Arista *ant;     /* La arista precedente de la lista */
    struct _Arista *sig;     /* La arista siguiente de la lista */
}*Arista;

typedef struct _Cara
{
    struct _Elemento_Cara *pri_elemento; /* El primer elemento de la cara */

    struct _Cara *ant; /* La cara precedente de la lista */
    struct _Cara *sig; /* La cara siguiente de la lista */
    double NCcp[3]; /* La normal de esta cara */
    double cent[3]; /* El centroide de esta cara */
    float color[3]; /* Color de la cara en RGB */
}*Cara;

typedef struct _Elemento_Cara
{
    struct _Punto * punto; /* El vértice adyacente */
    struct _Arista * arista; /* La arista compuesta por el vértice adyacente */
                        /* y el vértice que le sigue */
    struct _Cara * cara; /* La cara adyacente a la arista, distinta a la */
                        /* cara a que pertenece */
    struct _Elemento_Cara *ant; /* El elemento precedente de la cara */
    struct _Elemento_Cara *sig; /* El elemento siguiente de la cara */
}*Elemento_Cara;

```

Debido a que las caras pueden estar formadas por n aristas, cada cara debe de tener una lista en donde se especifique cual es el conjunto de aristas, para esto se creó la estructura `_Elemento_Cara`. Además `_Elemento_Cara` es una lista doblemente ligada, donde se especifican las aristas y los vértices que pertenecen a éstas.

Estos vértices, serán los primeros de cada arista, para que de tal manera, al visualizar nuestra cara, se pueda especificar de una manera fácil cuál es el siguiente vértice que forma la cara. Como el `_Elemento_Cara` es una lista de aristas, dentro de la estructura `_Cara` se guardará el primer elemento de esta lista.

2.2. Modelos iniciales

Para aplicar un esquema de deformación a una malla de simplejos necesitamos primero generar unos modelos tridimensionales básicos hechos con mallas de simplejos. A estos modelos tridimensionales básicos les llamaremos *modelos iniciales*. En esta tesis proponemos los siguientes modelos iniciales:

- La esfera
- El cilindro, y
- El toroide

En esta sección describiremos los procedimientos necesarios para construir cada uno de estos modelos en base a una malla de simplejos. De hecho, los modelos iniciales serán también *poliedros* y serán aproximaciones del modelo tridimensional básico.

El procedimiento general para construir los modelos iniciales consta de los siguientes pasos:

1. Crear los vértices (puntos en el espacio) del modelo.
2. Crear las aristas, uniendo los vértices generados en el paso anterior, y
3. Crear las caras del modelo inicial a partir de las aristas.

Primero se tratará el caso más difícil que es la creación de una esfera por medio del rellenado de su superficie con una malla de simplejos. La creación de los otros dos modelos iniciales –el cilindro y el toroide– son casos especiales del procedimiento para crear una esfera.

El paso 3 del procedimiento general para la creación de los modelos iniciales es idéntico para los tres modelos, por lo que sólo se explicará en detalle para el caso de la construcción de un esfera.

2.2.1. Creación de una esfera con mallas de simplejos

Vamos a tratar el procedimiento para la creación de una esfera rellenando su superficie con una malla de simplejos. Como ya se mencionó el procedimiento sigue tres pasos principales: la generación de los vértices, la generación de aristas y la generación de caras. Iniciaremos con un paso más general de la creación de vértices, para después llegar al procedimiento particular para generar los vértices que formarán una esfera.

Creación de vértices

La creación de vértices consiste en asignar valores a la tripleta de coordenadas (x, y, z) , que representa la posición de un punto en el espacio. Con todos los valores para cada vértice podemos formar una *lista* de vértices.

Iniciemos imaginando cómo se podría rellenar un plano con una malla de simplejos. Para generar esta malla no se puede asignar un conjunto de valores arbitrarios a los vértices que la conforman. Un vértice perteneciente a una malla de simplejos debe de contar con otros tres vértices vecinos (ó cada vértice pertenece a tres aristas, ver la Sec.

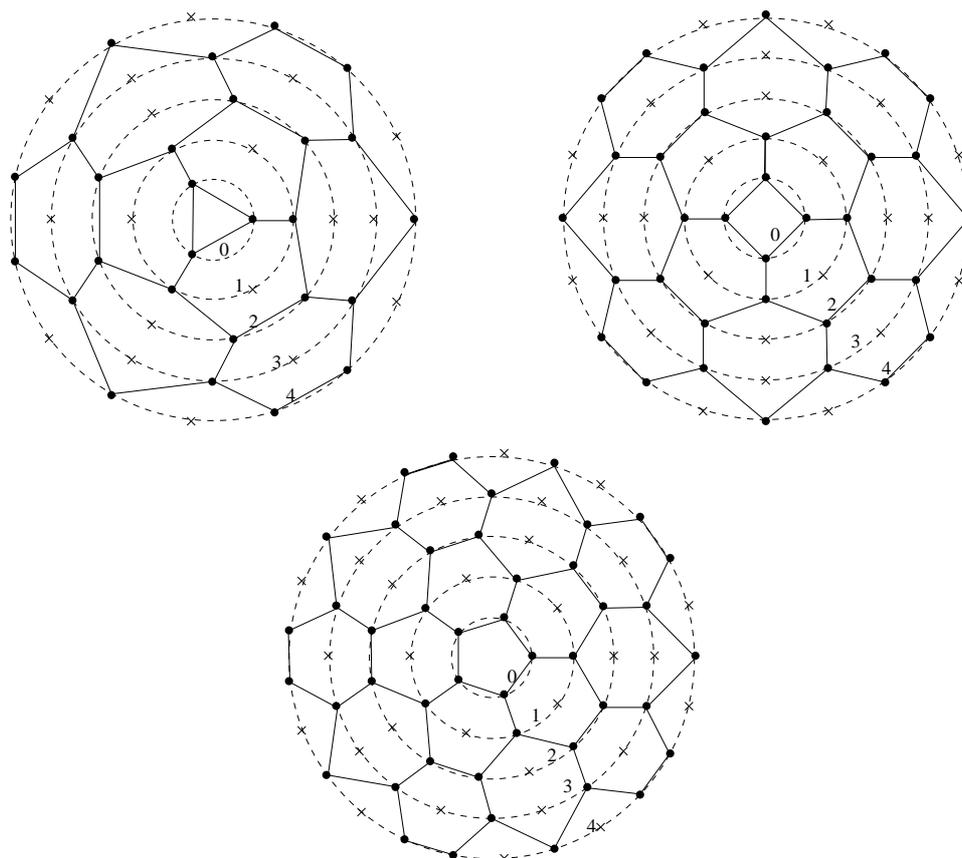


Figura 2.9: Mallas de simplejos trazados a mano sobre un triángulo, un cuadrado y un pentágono. Detalles de como se generaron los vértices en el texto.

2.1 en la página 13. La figura bidimensional que cumple más fácilmente esta condición es el hexágono. Al rellenar una superficie con hexágonos, sus aristas y vértices generan una malla-2 de simplejos.

Ahora bien, para generalizar la idea descrita en el párrafo anterior, se puede rellenar un plano con una malla de simplejos con el siguiente procedimiento: podemos partir de un polígono regular de f lados e ir pegando un hexágono a cada uno de sus lados. Por supuestos, estos hexágonos serán irregulares. Esta idea puede verse en la Fig. 2.9, donde se han trazado “a mano” tres mallas de simplejos sobre un triángulo, un cuadrado y un pentágono. Estas mallas se trazaron sobre vértices que enseguida pasaremos a explicar como se generaron.

Los vértices mostrados en la Fig. 2.9 se generaron de la siguiente manera: el polígono regular inicial está inscrito en una circunferencia. Este polígono inicial tienen f lados, por lo que están creados f vértices. Sobre la circunferencia inicial se generaron otras circunferencias concéntricas, todas con el mismo origen; sobre cada una de estas circunferencias se trazaron cf vértices, donde c es el número del orden de la circunferencia, empezando por la más interna que es la número 1. Así, la circunferencia inicial en la misma Fig. 2.9, tiene 3 vértices, la segunda tiene $3 \cdot 2$ vértices, la tercera $3 \cdot 2$ vértices, etc. Esta relación en el número de vértices para cada circunferencia, cf , tiene que ver con que se está llenando la superficie con hexágonos. Por lo tanto, el número de vértices, n_v^m , correspondientes a

la circunferencia m , será:

$$n_v^m = mf, \quad \text{para } 1 \leq m \leq n \quad (2.16)$$

En la Ec. 2.16, n representa el número total de circunferencias concéntricas que se trazan.

Definiremos c como el número de capas de la malla de simplejos. Para las mallas dibujadas en la Fig. 2.9 el número de capas es 3 y significa el número de capas de hexágonos que rodean el polígono inicial más el polígono inicial, que se considera la primera capa. La ecuación que relaciona el número de capas, c , con el número de circunferencias concéntricas, n , es:

$$n = 2c - 1 \quad (2.17)$$

Para la Fig. 2.9, el número de capas c es igual a 3 y el número de circunferencias concéntricas es 5. Si vemos la Ec. (2.17), el número de circunferencias, n , será siempre un número impar. Esto nos ayudará a unir varias mallas, ya que de esta forma, si se unen en sus bordes, se seguirán formando simplejos.

Los valores de las coordenadas (x, y) para cada vértice pueden calcularse fácilmente: si el radio de la circunferencia más externa es r , entonces el radio de la circunferencia m será

$$r_m = \frac{rm}{n}, \quad \text{para } 1 \leq m \leq n$$

y los valores de las coordenadas serán

$$\begin{aligned} x_k &= r_m \cos\left(\frac{2\pi k}{n_v^m}\right) \\ y_k &= r_m \operatorname{sen}\left(\frac{2\pi k}{n_v^m}\right) \end{aligned}$$

para $0 \leq k < n_v^m$ y cada circunferencia m .

Vértices para la malla de simplejos

En la Fig. 2.9 se han trazado todos los vértices con el proceso explicado en la sección anterior. Ahora, observando la misma Fig. 2.9, vemos que no se usan todos los vértices (los marcados con \times) para trazar la malla-2 de simplejos. Es por ello que necesitamos ahora un procedimiento para *seleccionar* los vértices que sólo pertenecen a la malla-2 de simplejos. Tal procedimiento se muestra en el número 1.

Para entender el procedimiento 1 debe observarse la Fig. 2.10: se tiene que en las circunferencias con i cero e impar se seleccionan los vértices de forma equiespaciada. Y se sigue la secuencia del Proc. 1 para las circunferencias con i par. Todos los vértices seleccionados están marcados en la Fig. 2.10 con puntos negros.

Vértices para una esfera

Ahora reproduciremos una parte de la Fig. 2.9 para poder explicar el procedimiento para la obtención de los vértices para una esfera.

Observando de nueva cuenta la Fig. 2.10, podemos imaginarnos el procedimiento para la generación de una esfera: si doblamos la malla de simplejos sobre una semiesfera, generando otra semiesfera de la misma forma y uniendo ambas semiesferas se formaría la esfera que buscamos. La segunda semiesfera se podría crear de manera simultánea a la

Procedimiento 1 Selecciona los vértices para la malla-2 de simplejos

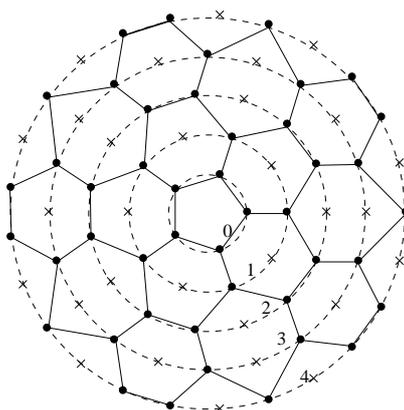
Require: $c \geq 1$ y $f \geq 3$ **Ensure:** Listas de vértices \mathcal{L}_v^i por cada circunferencia i $n = 2c - 1$ **for** $i = 0$ to $n - 1$ **do** $n_v^i \leftarrow (i + 1) * f$ **for** $j = 0$ to $n_v^i - 1$ **do****if** $i \bmod 2 = 0$ **then** $\{i \text{ es par}\}$ $\text{cont} \leftarrow (j + i/2) \bmod (i + 1)$ **else** $\{i \text{ es impar}\}$ $\text{cont} \leftarrow j + i/2$ **end if****if** $\text{cont} \bmod 2 = 0$ **then** $\mathcal{L}_v^i \leftarrow \mathcal{L}_v^i + \text{vertice}(i, j)$ **end if****end for****end for**

Figura 2.10: Malla de simplejos dibujada sobre un pentágono regular. Se han numerado las circunferencias concéntricas con el índice i del procedimiento 1. Para la figura, $f = 5$, $n = 5$ (ó $c = 3$)

primera, invirtiendo los valores de los vértices sobre alguno de los ejes (x , y ó z). Si se escoge el eje z , entonces un vértice con coordenadas (x, y, z) sobre la primera semiesfera, tendrá los valores $(x, y, -z)$ sobre la segunda semiesfera. Para la notación usaremos v para un vértice sobre la primera semiesfera y usaremos v' para designar un vértice sobre la segunda semiesfera. Si se observa con cuidado la Fig. 2.10, se puede deducir que si se unen los bordes de dos semiesferas sucede lo siguiente

- se forman automáticamente simplejos al unir los vértices de los bordes v con el respectivo v' .
- se forman cuadrados en los bordes, en la proyección de cada lado del polígono inicial, siguiendo los hexágonos perpendiculares a cada lado. Así nuestro procedimiento genera una esfera con una malla de simplejos a partir de un prisma como el de la Fig. 2.11

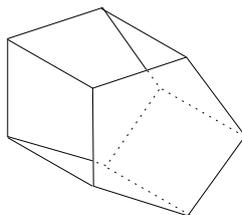


Figura 2.11: Prisma pentagonal. La esfera se crea expandiendo cada cara del prisma y llenando el espacio con hexágonos.

El procedimiento 2 muestra cómo se crean los vértices sobre la superficie de una esfera.

Procedimiento 2 Procedimiento para crear los vértices para una esfera

Require: $c \geq 1$

Ensure: L_v^i y $L_{v'}^i$, listas de vértices para la semiesfera anterior y posterior, para cada circunferencia

$n = 2c - 1$

for $i = 0$ to n **do**

$n_v^i \leftarrow f * (i + 1)$

for $j = 0$ to n_v^i **do**

if $i \bmod 2 = 0$ **then** $\{i$ es par $\}$

$\text{cont} \leftarrow (j + i/2) \bmod (i + 1)$

else

$\text{cont} \leftarrow j + i/2$

end if

if $\text{cont} \bmod 2 = 0$ **then** $\{\text{cont}$ es par $\}$

$\mathcal{L}_v^i \leftarrow \mathcal{L}_v^i + \text{vertice}(0, i, j)$

$\mathcal{L}_{v'}^i \leftarrow \mathcal{L}_{v'}^i + \text{vertice}(1, i, j)$

end if

end for

end for

En la práctica, cada vértice que se genera se guarda en una lista doblemente ligada de vértices. Las listas \mathcal{L}_v^i y $\mathcal{L}_{v'}^i$ no existen realmente, sino que son listas de punteros a la lista de vértices.

La función $\text{vertice}(flag, i, j)$, del procedimiento 3, regresa las coordenadas (x, y, z) del vértice dependiendo de los valores de los índices i y j . El valor de su primer argumento nos sirve para seleccionar si es un vértice sobre la semiesfera que vemos o sobre la que está “detrás”. Si consideramos el punto (x_0, y_0, z_0) como el origen de la esfera, la función vertice quedaría de la siguiente manera:

Los valores para los radios de las circunferencias, tanto para el plano xy como para su valor en z se calculan considerando el diagrama de la Fig. 2.13. Como se ve en esa figura, las circunferencias se sitúan equiespaciadas sobre la superficie de la esfera, por lo que el valor del ángulo θ será $(\pi/2)/2n$ ó $\pi/4n$.

Creación de aristas

Una vez que se han creado los vértices para la esfera usando el procedimiento 2, el siguiente paso es formar aristas entre ellos. Así mismo se guardará la vecindad de los

Procedimiento 3 Procedimiento para asignar valores a un vértice de la esfera
vertice(flag, i, j)

Require: flag, i, j . Los valores de c y r son globales

Ensure: Coordenadas (x, y, z) del vértice (i, j)

$$n = 2c - 1$$

$$n_v^i \leftarrow f * (i + 1)$$

$$rc \leftarrow r * \text{sen}(\pi(2i + 1)/4n)$$

$$x \leftarrow x_0 + rc * \cos(2\pi * j/n_v^i)$$

$$y \leftarrow y_0 + rc * \text{sen}(2\pi * j/n_v^i)$$

$$rc \leftarrow r * \cos(\pi(2i + 1)/4n)$$

if flag=0 **then**

$$z \leftarrow z_0 + rc$$

else

$$z \leftarrow z_0 - rc$$

end if

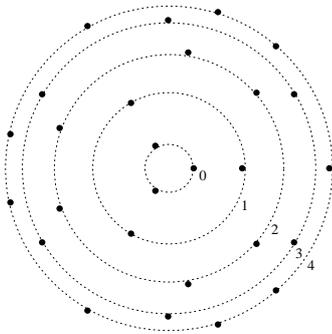


Figura 2.12: Los vértices ya seleccionados para construir una semiesfera. Para la figura, $f = 3$ y $c = 3$

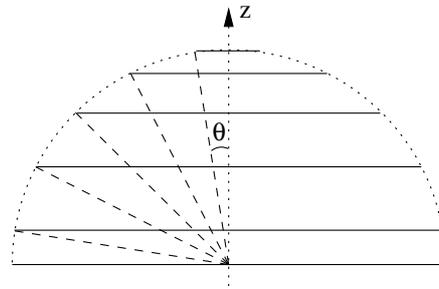


Figura 2.13: Diagrama lateral de una esfera. El número de capas es de 3, por lo que el número de circunferencias, n , es de 5; θ será igual a $\pi/4n$

vértices que forman aristas, en las estructuras de datos de nuestro esquema de visualización. Para la creación de las aristas suponemos que contamos con las listas de vértices seleccionados, \mathcal{L}_v^i y $\mathcal{L}_{v'}^i$, para cada circunferencia i , que son la salida del procedimiento 2. Un ejemplo de los vértices seleccionados para una esfera (viendo la proyección sobre el eje z) se muestra en la figura 2.12.

Se tienen cuatro casos para crear las aristas para una esfera:

1. Se une un vértice j de la circunferencia i , con el vértice j de la circunferencia anterior, $i - 1$.
2. Sobre alguna circunferencia i , se une un vértice i, j con el vértice $i, j - 1$.
3. Para las circunferencias $i \geq 3$, existe la situación donde se crean dos aristas al mismo tiempo: una como en el caso 1 y la otra uniendo el mismo vértice j de la circunferencia i , con el vértice *anterior*, $j - 1$ de la circunferencia anterior $i - 1$.
4. Para la circunferencia inicial se repite sucesivamente el caso 2 y el último vértice se une con el vértice 0.

El procedimiento para la creación de aristas para una esfera es el 4. En este procedimiento la función $arista(v_a, v_b)$ llena la estructura de datos `_Arista`; al mismo tiempo para cada uno de los vértices se asignan como `vecinos` en a estructura de datos `_Punto`, es decir, al final de la creación de aristas quedan hechos los simplejos para cada vértice. Aquí hace falta un procedimiento muy sencillo para ir ordenando los vértices de los simplejos en el sentido de las manecillas del reloj; se va recorriendo cada uno de los vértices y para el simplejo asociado se hace lo siguiente:

1. Se *normalizan* las coordenadas de dos primeros vértices vecinos, de forma que el vértice central del simplejo quede en la coordenada $(0, 0, 1)$ [8].
2. Se descarga el valor z de los dos vértices asociados.
3. Se ordenan estos dos vértices según el valor del producto cruz [9, Sec. 11.3]. Este ordenamiento es usado para crear la cubierta convexa de una serie de puntos usando el algoritmo de Graham [9, Sec. 11.3]. Como están dispuestos de forma circular, el tercer vértice ya queda ordenado. ¡Para ordenar tres puntos de forma circular solo necesitamos realizar una comparación!

El ordenamiento hace muy sencillo el procedimiento para la creación de caras, como veremos posteriormente.

Un ejemplo de las aristas creadas para una esfera, y de los lados de hexágonos y cuadrados que se crean al unir las dos semiesferas, puede verse en la Fig. 2.14.

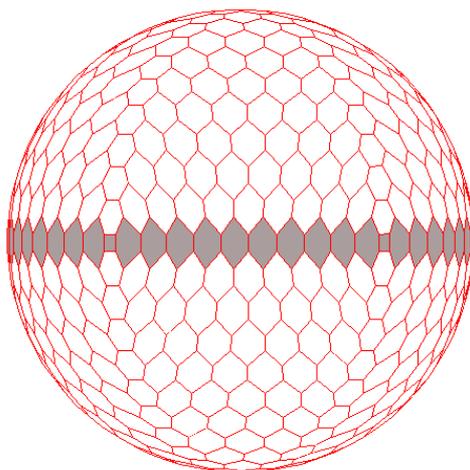


Figura 2.14: Ejemplo de las aristas creadas para una esfera. La unión de las dos semiesferas se muestra en gris.

Creación de caras

La creación de caras a partir de la lista de aristas es un problema más sencillo que la creación de vértices y aristas.

La creación de caras se podría hacer al mismo tiempo que las aristas, pero no sería tan sencillo como se mostrará aquí.

El procedimiento es muy sencillo aprovechando que los tres vértices, que rodean al vértice central de un simplejo, están ordenados en el sentido de las manecillas del reloj

Procedimiento 4 Crea la lista de las aristas de una esfera

Require:**Ensure:** $\mathcal{L}_a \leftarrow null$ **for** $i = 0$ to n **do** {Cada circunferencia} **for** $j = 0$ to $n_v^i - 1$ **do** {Cada vértice seleccionado de la circunferencia i } **if** i es par **then** **if** $i > 1$ **then**

{Se calcula el índice al vértice de la circunferencia anterior}

 $k \leftarrow (j + 1 - \frac{2j+i/2+1}{i+2} - (\frac{i}{2} \bmod 2)) \bmod (f \frac{i}{2})$ **end if** $d \leftarrow (j + 1 + i/4) \bmod (i/2 + 1)$ **if** $d = 0$ **then** **if** $i > 1$ **then** $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v_{i,j}, v_{i-1,k})$ $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v'_{i,j}, v'_{i-1,k})$ **end if** **else if** $d = 1$ OR $(i = 0$ AND $j > 0$ **then** $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v_{i,j}, v_{i,j-1})$ $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v'_{i,j}, v'_{i,j-1})$ **else** $\{d > 1\}$ $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v_{i,j}, v_{i-1,k})$ $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v'_{i,j}, v'_{i-1,k})$ $l = (k + 1) \bmod (fi/2)$ $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v_{i,j}, v_{i-1,l})$ $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v'_{i,j}, v'_{i-1,l})$ **end if** **if** $i = 0$ AND $j = f - 1$ **then** {último vértice de la primera circunferencia} $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v_{i,j}, v_{i,0})$ $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v'_{i,j}, v'_{i,0})$ **end if** **else** $\{i$ es impar} $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v_{i,j}, v_{i-1,k})$ $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v'_{i,j}, v'_{i-1,k})$ **end if** **if** $i = (n - 1)$ **then** {Se unen las dos tapas para formar la esfera} $\mathcal{L}_a \leftarrow \mathcal{L}_a + arista(v_{i,j}, v'_{i,j})$ **end if** **end for****end for**

(visto el simplejo desde “arriba”). Entonces, para cada vértice en la lista de vértices se escoge la su primera arista asociada y se le pone una marca; en la arista se llega al siguiente vértice que pertenece a otro simplejo y se sigue la arista siguiente en el orden y se repite esto hasta llegar a la arista de origen. No se inicia con una arista que ya esté marcada.

De esta forma hallamos todas las caras de la malla y la esfera está terminada. El pseudocódigo de este procedimiento para hallar las caras se ve el el procedimiento 5.

Las aristas que se van recorriendo se van guardando en la estructura de datos `_Elemento_Cara` (en realidad se guardan punteros a los datos de las aristas, esto es, punteros a la estructura `_Punto`).

Procedimiento 5 Procedimiento para la creación de caras

Require: La lista de vértices

Ensure: La lista de caras $\mathcal{L}_{\text{caras}}$

$\mathcal{L}_{\text{caras}} \leftarrow \text{null}$

for Cada vértice en la lista de vértices **do**

for Para cada arista (son tres) asociada al vértice **do**

if La arista no está marcada **then**

$\mathcal{C} \leftarrow \text{null}$

 Se marca la arista

repeat

 Para el vértice extremo de la arista:

$\mathcal{C} \leftarrow \mathcal{C} + a_{\text{sig}}$

until Se llega al vértice inicial

$\mathcal{L}_{\text{caras}} \leftarrow \mathcal{C}$

end if

end for

end for

2.2.2. Creación de un cilindro

Creación de vértices

La creación de vértices para el caso de un cilindro es muy similar al caso de la esfera, de hecho este es un caso especial de aquel, en donde el tamaño del radio no se modifica, las circunferencia ya no son concéntricas y tampoco se aumenta el número de vértices de forma consecutiva. Todos los círculos tendrán el mismo número de vértices, n_v , y el mismo radio r . También cambia la relación entre el número de círculos, n , y el número de capas, c , como se expresa en la Ec. (2.18).

$$n = 2c, \quad \text{para } c \geq 3 \quad (2.18)$$

c en la Ec. (2.18) debe ser mayor ó igual a tres para que se construya el cilindro con al menos un hexágono en su envoltura.

Contemplando estas modificaciones, obtendremos el siguiente algoritmo:

Los valores de las coordenadas del vértice se calculan de la siguiente manera. z aumenta de forma lineal y consideramos que el centro del cilindro está a la mitad de su altura h .

Procedimiento 6 Procedimiento para la selección de vértices de un cilindro

Require: $c \geq 3$ y n_v **Ensure:** La lista de vértices del cilindro, L_v $n = 2c$ **for** $i = 0$ to $n - 1$ **do** **for** $j = 0$ to $2n_v - 1$ **do** $cont \leftarrow j + \left(\frac{i}{2} \bmod 2\right)$ **if** $cont \bmod 2 = 0$ **then** $L_v \leftarrow L_v + \text{vertice}(i, j)$ **end if** **end for****end for**

Por lo tanto z se calcula como:

$$z = z_0 - \frac{h}{2} + \frac{ih}{(n-1)}$$

j e i son las mismas que las del procedimiento 6. Las circunferencias están sobre el plano xy , y de hecho sólo existen dos valores posibles: como se ve en el Proc. 6, se duplican los posible índices para j y se escogen para un par de circunferencias consecutivas, los valores pares de j y para otro par de circunferencias se escogen los impares, esto quiere decir que para las circunferencias 0 y 1 se escogen los valores pares de j y para las circunferencias 2 y 3 los valores impares, debido a que $cont = j + ((i/2) \% 2)$, los vértices se seleccionan cuando $cont \% 2 = 0$. En general los valores para x y y se calculan como:

$$x = x_0 + r \cos\left(\frac{\pi j}{n_v}\right)$$

$$y = y_0 + r \sin\left(\frac{\pi j}{n_v}\right)$$

para el radio r del cilindro.

Creación de aristas

El procedimiento para la creación de las aristas de un cilindro está basado en el procedimiento 4 para la creación de vértices de una esfera; pero se tienen menos casos, debido a que en el cilindro todas las circunferencias tienen el mismo radio. El procedimiento queda como se ve en el Proc. 7.

2.2.3. Creación de un toroide**Creación de vértices**

Para la creación de los vértices de un toroide, nos basaremos totalmente en el algoritmo descrito en la sección referente al modelo del cilindro. Un toroide se puede ver como un cilindro unido por sus dos extremos. Si observamos la Fig. 2.15, vemos que no cualquier número de capas permite formar simplejos al unir los extremos del cilindro. Para que se puedan formar simplejos, el número de capas, c , tiene que ser un número múltiplo de dos.

Procedimiento 7 Procedimiento para la creación de aristas de un cilindro

Require: $c \geq 3$ y n_v
Ensure: La lista de aristas del cilindro, \mathcal{L}_a
 $n = 2c$
 $\mathcal{L}_a \leftarrow \text{null}$
for $i = 0$ to $n - 1$ **do** {Para cada circunferencia}

for $j = 0$ to $n_v - 1$ **do** {Para cada vértice ya seleccionado}

if i es par ó está en el último círculo **then**
if $(i = 0 \text{ AND } j > 0) \text{ OR } (i = (n - 1) \text{ AND } j > 1)$ **then**
 $\mathcal{L}_a \leftarrow \mathcal{L}_a + \text{arista}(v_{i,j}, v_{i,j-1})$
end if
if $i > 0 \text{ AND } i < (n - 1)$ **then**
 $k \leftarrow (n_v + \frac{i}{2} \bmod 2) \bmod n_v$
 $\mathcal{L}_a \leftarrow \mathcal{L}_a + \text{arista}(v_{i,j}, v_{i-1,k})$
 $k \leftarrow (n_v - (\frac{i}{2} + 1) \bmod 2) \bmod n_v$
 $\mathcal{L}_a \leftarrow \mathcal{L}_a + \text{arista}(v_{i,j}, v_{i-1,k})$
end if
if $(i = 0 \text{ OR } i = (n - 1)) \text{ AND } (j = 2(n_v - 1) \text{ OR } j = (2n_v - 1))$ **then**
 $\mathcal{L}_a \leftarrow \mathcal{L}_a + \text{arista}(v_{i,j}, v_{i,0})$
end if
else { i es impar}

 $\mathcal{L}_a \leftarrow \mathcal{L}_a + \text{arista}(v_{i,j}, v_{i-1,j})$
end if
end for
end for

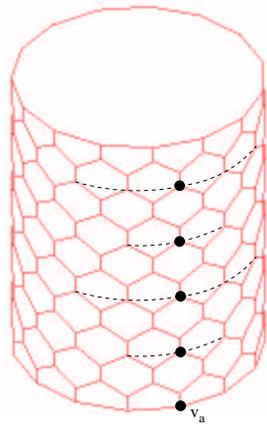


Figura 2.15: Un cilindro construido según el procedimiento 7. El vértice a , que está en la base, solamente puede unirse a los vértices arriba de él que están marcados con un punto negro, para poder formar un simplejo. Las líneas punteadas indican la última circunferencia que debe ser la misma que la base para poder formar un toro.

También, observando la Fig. 2.15, vemos que los vértices de la última circunferencia (que forma la tapa del cilindro) son exactamente los mismos que los vértices de la base; por lo tanto el número de circunferencias, n , será

$$n = 2c \quad (2.19)$$

La Ec. (2.19) es exactamente la misma que la Ec. (2.18) (en la pág. 30) pero tiene distinto significado: c para el toro solo puede ser un número par mayor o igual a 2; si hacemos $c = 4$, tenemos que $n = 8$ pero esto sería equivalente a realizar un cilindro con 9 circunferencias (que no es posible dada la Ec. (2.18)). En sí el algoritmo para crear las aristas del toro crea las aristas de un cilindro para un número impar de circunferencias.

El procedimiento para seleccionar los vértices para construir un toro es exactamente el mismo que para seleccionar los vértices para un cilindro (Proc. 6).

Ahora calcularemos los valores de los vértices del toroide en función de los vértices del cilindro. Para esto nos apoyaremos de la Fig. 2.16. En esta figura se ha dibujado un cilindro con 5 circunferencias sobre el sistema de coordenadas xz ; su número de caras es 2 y el número de circunferencias, según la Ec. 2.19 es 4. El cilindro tiene que doblarse para formar el toro, que esta representado con círculos punteados. El sistema de coordenadas para el toro es el xz' . El radio del cilindro es $(r_e - r_i)/2$, donde r_e es el radio externo y r_i el radio interno del toro. La circunferencia c_0 del cilindro si se traslada $-(r_e + r_i)/2, 0, 0$, es parte del toro. La circunferencia c_1 tiene que doblarse para formar la circunferencia c'_1 del toro y así sucesivamente para las demás circunferencia del cilindro. La circunferencia c_4 del cilindro no se toma en cuenta en el algoritmo para crear vértices porque es la misma que la circunferencia c_0 y es la que “cierra” el toro.

Si suponemos que el vector $[x, y, 0]^T$ son las coordenadas de los vértices sobre la circunferencia c_0 del cilindro (ver la Fig. 2.19). La circunferencia c_0 además que trasladarse $[-(r_{\text{ext}} + r_{\text{int}})/2, 0, 0]$ y rotarse para generar las siguientes circunferencias del toro. Traduciendo las coordenadas quedan como:

$$\begin{aligned} x' &= x - \frac{r_{\text{ext}} + r_{\text{int}}}{2} \\ y' &= y \\ z' &= 0 \end{aligned}$$

y luego tiene que rotarse $i\theta$ grados, donde θ para $0 \leq i < n$, es

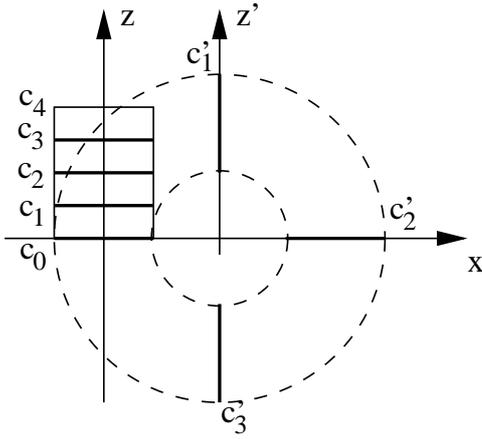


Figura 2.16: Esquema para visualizar como se dobla un cilindro para realizar el toroide. Las circunferencias c'_i del toro corresponden a las c_i del cilindro. La circunferencia c_4 del cilindro no se usa, es la misma que la c_0 . El sentido del eje y es hacia dentro de la página.

$$\theta = \frac{2\pi}{n} \quad (2.20)$$

Por lo que los puntos se transforman como

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} \cos(i\theta) & 0 & \text{sen}(i\theta) \\ 0 & 1 & 0 \\ -\text{sen}(i\theta) & 0 & \cos(i\theta) \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 0 \end{bmatrix}$$

lo que nos resulta en

$$\begin{aligned} x'' &= x' \cos(i\theta) \\ y'' &= y' \\ z'' &= -x' \text{sen}(i\theta) \end{aligned}$$

y los valores de x , y y z se calculan como en el caso del cilindro:

$$\begin{aligned} x &= \frac{r_{\text{ext}} - r_{\text{int}}}{2} \cos\left(\frac{\pi j}{n_v}\right) \\ y &= \frac{r_{\text{ext}} - r_{\text{int}}}{2} \text{sen}\left(\frac{\pi j}{n_v}\right) \\ z &= 0 \end{aligned}$$

Finalmente las ecuaciones quedan como:

$$\begin{aligned} x'' &= x_0 + \left[\frac{r_{\text{ext}} - r_{\text{int}}}{2} \cos\left(\frac{\pi j}{n_v}\right) + \frac{r_{\text{ext}} + r_{\text{int}}}{2} \right] \cos\left(\frac{2\pi i}{n}\right) \\ y'' &= y_0 + \frac{r_{\text{ext}} - r_{\text{int}}}{2} \text{sen}\left(\frac{\pi j}{2n_v}\right) \\ z'' &= z_0 - \left[\frac{r_{\text{ext}} - r_{\text{int}}}{2} \cos\left(\frac{\pi j}{n_v}\right) + \frac{r_{\text{ext}} + r_{\text{int}}}{2} \right] \text{sen}\left(\frac{2\pi i}{n}\right) \end{aligned} \quad (2.21)$$

donde el punto (x_0, y_0, z_0) sería un nuevo origen para el toro.

Creación de aristas

El procedimiento para la creación de las aristas de un toroide está basado en gran parte en el procedimiento 7 para crear las aristas de un cilindro. Un toroide se puede considerar como un cilindro que tiene unidos sus dos extremos, dado esta diferencia, el procedimiento se tiene que modificar para crear aristas entre la tapa y la base del cilindro.

El procedimiento, en pseudocódigo, para crear las aristas de un toro es el 8.

Procedimiento 8 Algoritmo para la creación de aristas de un toroide

Require: $c \geq 2$, el número de caras, c , debe ser par, y n_v (el número de vértices sobre una circunferencia)

Ensure: La lista de aristas del toroide, \mathcal{L}_a

$n = 2c$

for $i = 0$ to $n - 1$ **do** {Para cada circunferencia}

for $j = 0$ to $n_v - 1$ **do** {Para cada vértice ya seleccionado}

if i es par **then**

if $i > 0$ **then**

$k \leftarrow (j + \frac{i}{2} \bmod 2) \bmod n_v$

$\mathcal{L}_a \leftarrow \mathcal{L}_a + \text{arista}(v_{i,j}, v_{i-1,k})$

$k \leftarrow (j - (\frac{i}{2} + 1) \bmod 2) \bmod n_v$

$\mathcal{L}_a \leftarrow \mathcal{L}_a + \text{arista}(v_{i,j}, v_{i-1,k})$

end if

else { i es impar}

$\mathcal{L}_a \leftarrow \mathcal{L}_a + \text{arista}(v_{i,j}, v_{i-1,j})$

if i está en la última circunferencia **then** {Se unen la tapa y la base del cilindro}

$k \leftarrow (j + \frac{i}{2} \bmod 2) \bmod n_v$

$\mathcal{L}_a \leftarrow \mathcal{L}_a + \text{arista}(v_{i,j}, v_{i-1,k})$

$k \leftarrow (j - (\frac{i}{2} + 1) \bmod 2) \bmod n_v$

$\mathcal{L}_a \leftarrow \mathcal{L}_a + \text{arista}(v_{i,j}, v_{i-1,k})$

end if

end if

end for

end for

El procedimiento para la creación de las caras del toroide es exactamente el mismo que para crear las caras de una esfera (Proc. 4, en la pág. 29). Un ejemplo de un toro creado por los procedimientos aquí explicados puede verse en la Fig. 2.17

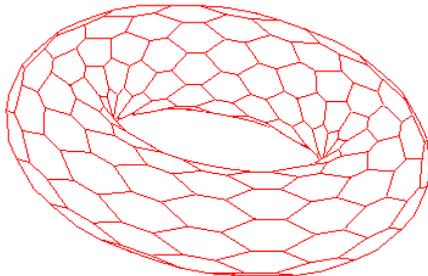


Figura 2.17: Ejemplo de la construcción de un toro.

2.3. Esquema de deformación

En esta sección vamos a describir el esquema de deformación de las mallas de simplejos planteada por H. Delingette en sus artículos [7], [10], [11] y en su tesis doctoral [12]. Al contrario que en las representaciones paramétricas, la geometría de una malla está definida solamente por la posición de sus vértices. La deformación no se basa en la noción de derivadas parciales, sino en la posición relativa de un vértice con respecto a sus vecinos; para ser más específicos, con respecto al valor del ángulo del simplejo y a sus parámetros métricos. Como en la mayoría de los esquemas deformables, todos los vértices de una malla de simplejos son considerada como una masa sujeta a las leyes de la física de Newton, consistiendo de fuerza interna y externa:

$$m \frac{d^2 P_i}{dt^2} = -\gamma \frac{dP_i}{dt} + F_{int} + F_{ext} \quad (2.22)$$

donde m es la unidad de masa del vértice y γ es el factor de amortiguamiento. F_{int} es la fuerza interna que impone la continuidad de la forma de la malla. F_{ext} es la fuerza externa que fuerza a la malla a ajustarse a un conjunto de datos tridimensionales. Se discretiza el tiempo t en orden para computar la evolución de la malla de simplejos bajo la leyes de movimiento descrita en la ecuación 2.22. Usando diferencias finitas centrales con un esquema explícito y tomando el valor de m igual a uno, la ley de movimiento es discretizada como:

$$P_i^{t+1} = P_i^t + (1 - \gamma)(P_i^t - P_i^{t-1}) + F_{int} + F_{ext} \quad (2.23)$$

Ambas fuerzas F_{int} y F_{ext} son computadas a un tiempo t , con la ecuación 2.23, la fuerza interna y externa tienen la dimensión de un desplazamiento.

2.3.1. Computación de la fuerza interna

La fuerza interna que actúa sobre una malla de simplejos se computa basándose en la minimización de un criterio local, la fuerza interna es derivada de la minimización de la energía local que es proporcional a la desviación de una posición ideal. Tenemos que \mathcal{S}_i es el criterio local para minimizar en el vértice P_i :

$$\mathcal{S}_i = \frac{\alpha_i}{2} (P_i P_i^*)^2 \quad (2.24)$$

Entonces la fuerza interna será:

$$F_{int} = \frac{\partial \mathcal{S}_i}{\partial P_i} = \alpha_i P_i P_i^*$$

en donde :

- α_i es un escalar ($0 < \alpha_i < 0.5$) que regula el valor de la fuerza interna con respecto a la fuerza externa. Valores de α_i por encima de 0.5 dan como resultado una deformaciones inestables.
- P_i^* es definido con respecto a los vecinos de P_i , con el ángulo del simplejo φ_i^* y los parámetros métricos $(\epsilon_i^{1*}, \epsilon_i^{2*}, \epsilon_i^{3*})$.

Usando la ecuación 2.15, la fuerza interna puede expresarse como:

$$F_{int} = \alpha_i(\epsilon_i^{1*} P_i P_{N_1(i)} + \epsilon_i^{2*} P_i P_{N_2(i)} + \epsilon_i^{3*} P_i P_{N_3(i)} + L(r_i, d_i, \varphi_i^*) n_i) \quad (2.25)$$

El ángulo del simplejo φ_i^* puede que no sea constante entre cada iteración, esto depende del tipo de condición de regularidad que deba ser utilizado. A continuación se dan los tipos de condiciones que pueden ser utilizados:

- *Condición de continuidad de posición.* Hacemos $\varphi_i^* = \varphi_i$. La superficie puede doblarse libremente alrededor del vértice P_i . Esto es útil para representar discontinuidades en la orientación de las superficies.
- *Condición de continuidad para orientación de superficies.* Simplemente asignamos $\varphi_i^* = 0$. Por lo tanto, la fuerza interna puede escribirse como:

$$F_{int} = \alpha_i(\epsilon_{1i} P_{N_1(i)} + \epsilon_{2i} P_{N_2(i)} + \epsilon_{3i} P_{N_3(i)} - P_i) \quad (2.26)$$

Cuando los parámetros métricos son iguales a $\frac{1}{3}$, obtendremos:

$$F_{int} = \frac{\alpha_i}{3}(P_{N_1(i)} + P_{N_2(i)} + P_{N_3(i)} - 3P_i) \quad (2.27)$$

El vértice P_i es atraído hacia el centro del triángulo formado por sus vecinos. La continuidad de la normal se garantiza con esto, pero no la continuidad del ángulo del simplejo o la curvatura media.

- *Condición de forma.* Teniendo la constante φ_i^0 se asigna $\varphi_i^* = \varphi_i^0$ así condicionamos el ángulo del simplejo en P_i a φ_i^0 . La malla es deformada a través de alguna forma de referencia descrita por el ángulo del simplejo φ_i^0 y los parámetros métricos $(\epsilon_i^{1*}, \epsilon_i^{2*}, \epsilon_i^{3*})$.
- *Condición de continuidad del ángulo del simplejo.* φ_i^* es definida como:

$$\varphi_i^* = \sum_{j \in Q^{s_i}(P_i)} \varphi_j \quad (2.28)$$

φ_i^* es el valor medio de los ángulos de los simplejos en la vecindad $Q^{s_i}(P_i)$ de tamaño s_i alrededor de P_i . Esta vecindad es definida de manera recursiva. Definiremos $Q^0(P_i) = \{P_i\}$ y $Q^1(P_i) = \{P_i, P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)}\}$. La vecindad $Q^{s_i} P_i, s_i > 1$ es la unión de $Q^{s_i-1}(P_i)$ con los vértices de \mathcal{M} que tengan un vértice vecino en $Q^{s_i-1}(P_i)$. s_i corresponde intuitivamente a la noción de rigidez, o deformación de escala. En un modelo altamente rígido, el modelo es suavizado a través de una escala larga, y por lo tanto una condición externa podría implicar una pequeña desviación sobre una larga extinción de la superficie. La rigidez también tiene efectos sobre la dinámica del modelo deformable. Esto debido a que la escala de deformación influencia la velocidad de la condición de propagación de la superficie. Una malla flexible con una rigidez pequeña necesita más iteraciones para alcanzar una posición estable que una malla con alta rigidez.

En la realización de esta tesis se ha utilizado la *condición de continuidad para orientación de superficies* para la implementación de este modelo de deformación, esto debido a que al mantener el ángulo $\varphi_i^* = 0$ y al hacer los parámetros $(\epsilon_{1i} = \epsilon_{2i} = \epsilon_{3i} = \frac{1}{3})$, se obtiene estabilidad en la distribución de los vértices deformados, y a su vez se ahorra bastante tiempo máquina al resultar más ligeros los cálculos para cada vértice.

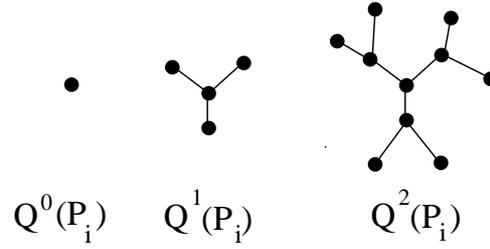


Figura 2.18: La vecindad $Q^{s_i}(P_i)$ alrededor del vértice P_i .

2.3.2. Computación de la fuerza externa

La fuerza externa F_{ext} es proporcional a la distancia de P_i al conjunto de datos al que se quiere aproximar. La fuerza computada es dirigida a través de la dirección de la normal n_i en el vértice P_i en donde la fuerza es aplicada.

La colinearidad de F_{ext} con n_i es importante por dos razones:

1. Esta implica deformaciones suaves con fuerzas atractoras variando de manera sutil al rededor de la malla. Desde que se usa un proceso iterativo es importante garantizar un comportamiento estable y fluido entre iteraciones. De hecho los desplazamientos tangenciales pueden crear vértices con parámetros negativos (o para ser más claros con la proyección F_i fuera del triángulo formado por los tres vecinos del vértice P_i) o hasta una malla con intersecciones entre sus paredes.
2. La dirección de la normal en la fuerza externa asegura que la forma resultante de la malla tenga transiciones suaves hasta en aquellos casos en que los datos existentes estén muy dispersos entre si. Si la fuerza de atracción ha sido dirigida a través de la dirección de la de unión del vértice P_i con el dato más cercano, la malla en equilibrio no tendrá una estructura y forma intuitiva.

La complejidad en el computación de la fuerza externa es lineal con respecto al número de vértices de la malla.

2.3.3. Computación de la fuerza externa en rangos de datos

La computación de la fuerza externa es dependiente del concepto del punto más cercano. Para cada vértice P_i de la malla, se busca el punto más cercano de nuestro conjunto de datos $M_{Cl(i)}$ y entonces la fuerza externa es computada como:

$$F_{ext} = \beta_i G \left(\frac{\|P_i M_{Cl(i)}\|}{D_{ref}} \right) (P_i M_{Cl(i)} \cdot n_i) n_i \quad (2.29)$$

donde:

- β_i es un parámetro para regular la fuerza.
- n_i es el vector normal a la malla en el vértice P_i .
- $G(x)$ es la función de rigidez.
- D_{ref} es una distancia de referencia.

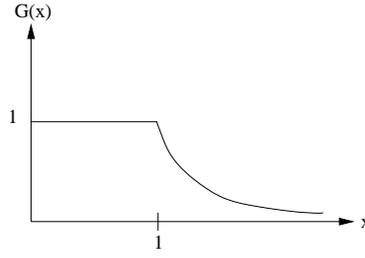


Figura 2.19: La función de rigidez $G(x)$.

La fuerza F_{ext} es calculada como la proyección del vector $P_i M_{Cl(i)}$ a través de la dirección del vector normal. La distancia D_{ref} es la máxima distancia de atracción de un punto en los datos, la función de rigidez $G(x)$ se asegura de que la fuerza decrezca bruscamente cuando la distancia entre $M_{Cl(i)}$ y P_i sea mayor que D_{ref} . De hecho es muy importante controlar el grado de atracción de los puntos en los datos, de forma tal que limite la influencia de puntos muy alejados. La distancia D_{ref} es computada de manera relativa al tamaño aproximado de el conjunto de datos, como por ejemplo del radio de la esfera que rodea al conjunto de datos. Durante la primera etapa de la deformación, escogemos un valor relativamente grande de D_{ref} (mayor al 20% del radio) de manera tal que pueda permitir grandes deformaciones de la malla. Después de varias iteraciones el valor de D_{ref} es decrementado (al menos al 8% del radio) con la intención de limitar el efecto de los puntos alejados y aumentar la rapidez de la búsqueda del punto más cercano. En el desarrollo de esta tesis $G(x)$ se planteó de la siguiente manera:

$$G(x) = \begin{cases} 1 & \text{para } 0 \leq x \leq 1 \\ \frac{1}{x} & \text{para } x > 1 \end{cases} \quad (2.30)$$

2.3.4. Computación de la fuerza externa en imágenes volumétricas

Las imágenes volumétricas son utilizadas ampliamente en el área de medicina, éstas son generadas por tomografías tomadas con máquinas de rayos-x y procesos de imágenes de resonancia magnética. En nuestro caso son resultado de una reconstrucción tridimensional a partir de proyecciones bidimensionales. Un conjunto de datos en una imagen volumétrica se puede entender como una colección de imágenes estáticas, conocidas como rebanadas. Las rebanadas representan varias secciones de un sujeto en varias posiciones a lo largo de la tercera arista (eje coordenado z). En imágenes volumétricas, la tarea de reconstrucción usualmente consiste en aislar regiones con valores de intensidad constante. Por lo tanto la intensidad del gradiente es la información principal en la cual se basa la fuerza externa. Para la computación de F_{ext} se combinan dos fuerzas, la intensidad del gradiente con la información de borde de tal forma que F_{ext} se puede expresar de la siguiente forma:

$$F_{ext} = F_{grad} + F_{borde} \quad (2.31)$$

La intensidad del gradiente es usada para la máxima desviación de la malla alrededor de los voxels de máxima variación en intensidad. La información de borde por otra parte, corresponde al valor máximo del gradiente e implica grandes deformaciones en la malla. La imagen de borde es una imagen binaria extraída de la intensidad de la imagen del

gradiente a través del uso de la umbralización. En [10] se propone una expresión para la fuerza del gradiente que elude todas las oscilaciones y no es afectada por el valor máximo local de intensidad del gradiente. La fuerza F_{grad} en el vértice P_i depende de una búsqueda en la vecindad de P_i , del voxel con la mayor intensidad del gradiente. Si \mathcal{V} es el voxel más cercano de P_i entonces, se inspeccionan todos los voxels en una ventana de $m \times m \times m$ alrededor de \mathcal{V} por el voxel central G_i , que contenga el valor más alto de la intensidad del gradiente. La expresión de esta fuerza es entonces:

$$F_{grad} = \beta_i^{grad} (P_i G_i \cdot n_i) n_i \quad (2.32)$$

β_i es un factor de regulación con: $0 \leq \beta_i^{grad} \leq 1$. Desde que la fuerza es proporcional al verdadero vector de desviación $P_i G_i$ (y no el vector del gradiente), esta fuerza implica que no habrá oscilaciones de la malla. La imagen de borde se construye por medio de umbralizar los valores de intensidad del gradiente. La aproximación utilizada consiste en encontrar el voxel de borde más cercano en la dirección normal de la malla. En el vértice P_i encontramos el voxel más cercano \mathcal{V} , y una línea tridimensional de voxels es examinada en la dirección de n_i . El número máximo de voxel es examinado dependiendo del valor de la distancia de referencia D_{ref} , determinada como un porcentaje del radio total de la imagen de borde. Una versión tridimensional del algoritmo de Bresenham ha sido implementada (véase apéndice) para la buscar de manera eficiente el voxel a través de la línea. Si E_i es el voxel más cercano a lo largo de la línea normal a la malla n_i , entonces la fuerza de borde está dada por:

$$F_{borde} = \beta_i^{borde} P_i E_i \quad (2.33)$$

El vector $P_i E_i$ es colineal con n_i y β_i^{borde} corresponde a la rigidez de la fuerza de borde: $0 \leq \beta_i^{borde} \leq 1$. En nuestro caso en particular, las imágenes volumétricas que utilizamos para realizar las pruebas son binarias. Por lo tanto cada rebanada de alguno de nuestros especímenes de pruebas constara de 2 posibles valores (como se puede apreciar en la figura 2.20 (a)) 0 y 1. Al ser de esta naturaleza nuestras imágenes, el cálculo y la suma de la fuerza de gradiente no tiene sentido alguno de manera tal que la ecuación de la fuerza externa será:

$$F_{ext} = F_{borde} = \beta_i^{borde} P_i E_i \quad (2.34)$$

La búsqueda del voxel E_i , se limitara a la búsqueda del voxel más cercano con valor igual a 1 en la dirección del vector normal n_i .

2.4. Algoritmo de aproximación

Originalmente el esquema planteado por [11] muestra un algoritmo el cual consta de tres etapas, las cuales se comentarán a continuación de manera breve:

1. Se crea la malla con un modelo inicial (de los ya anteriormente mencionados) y ésta se transforma de su forma original a una aproximación burda del modelo a reconstruir.

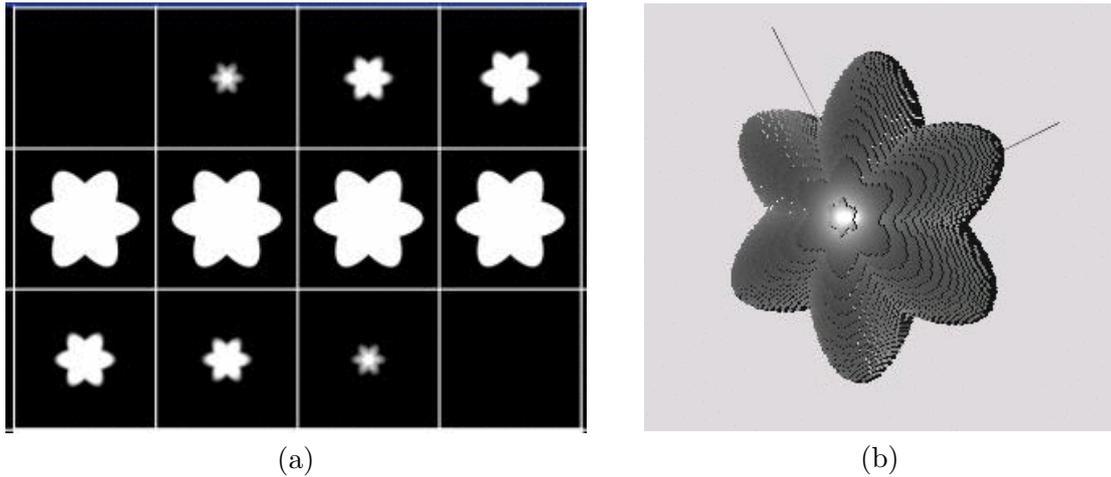


Figura 2.20: (a) El modelo de una estrella del cual se muestran algunas rebanadas. (b) La representación 3D del volumen presentado en (a).

2. Se crean hoyos y se cambia el *género* de nuestro modelo, cabe mencionar que esta es una acción asistida por un humano debido a que este es un problema bastante complejo por lo cual todavía no existe un algoritmo que de forma automática discrimine y elimine las caras que están dentro de un hoyo, y a su vez una las restantes.
3. Nuevamente se deforma para aproximar lo mejor posible la malla al modelo final, si es necesario se le aplicará a la malla un algoritmo de adaptación o refinamiento el cual puede consistir en concentrar los vértices en las zonas con mayor curvatura, o en aumentar el número de caras subdividiendo las caras que se encuentren en zonas con mayor curvatura.

El algoritmo utilizado por nosotros es muy similar al planteado originalmente, con la diferencia que en nuestra reconstrucción solo maneja un *género* igual a cero, resultando con esto en la eliminación de la fase dos del algoritmo anteriormente presentado, debido a la relativa complejidad de los algoritmos de adaptación y a la naturaleza de los modelos manejados en esta tesis no es necesario la implementación de estos y se ha optado por dejarlo como una mejora a futuro de la implementación realizada dentro de esta tesis. En el siguiente diagrama podemos apreciar el funcionamiento de nuestro algoritmo.

El número de iteraciones necesarias para llegar al modelo final se realizó de la siguiente manera. Se calcula la suma de las magnitudes de la fuerza externa y de la fuerza interna (la fuerza es un vector) relacionadas cada simplejo:

$$FTE_i = \sum_{j=0}^{N_v-1} |F_{ext}i_j| \quad (2.35)$$

$$FTI_i = \sum_{j=0}^{N_v-1} |F_{int}i_j| \quad (2.36)$$

La suma de las magnitudes de la fuerza externa la llamamos *fuerza total externa* (FTE), en la Ec. (2.35). La suma de las magnitudes de la fuerza interna la llamamos *fuerza total interna* (FTI), en la Ec. (2.36).

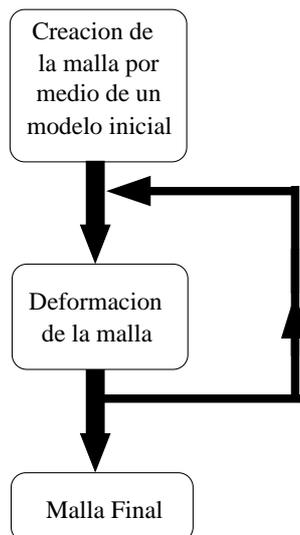


Figura 2.21: Algoritmo de aproximación.

Ahora, podemos tomar dos medidas para el error total en cada iteración, el error cuadrático medio (RMS):

$$\text{error}_{\text{RMS}}^i = \sqrt{\text{FTE}_i^2 + \text{FTI}_i^2} \quad (2.37)$$

para cada iteración i ; ó también podemos calcular el error absoluto ($\text{error}_{\text{abs}}^i$):

$$\text{error}_{\text{abs}}^i = |\text{FTE}_i - \text{FTI}_i| \quad (2.38)$$

Hemos usado la gráfica de error absoluto, en la ecuación (2.38), como criterio de paro para el algoritmo de deformación. Una gráfica típica para los valores de FTE, FTI, $\text{error}_{\text{RMS}}$ y $\text{error}_{\text{abs}}$, puede verse en la Fig. 2.22. En esta gráfica se puede observar que después de alrededor de veinte a cuarenta iteraciones la magnitud de las fuerzas empieza a ser las mismas de manera periódica, por lo tanto cuando la magnitud de la fuerza total interna y externa empieza a oscilar, el modelo entra a en un estado de equilibrio y su forma ya no variará de manera visible (como se muestra en las imágenes 2.23 y 2.24), por lo que al momento en que se entra en este estado será el momento ideal para dar por terminado el algoritmo.

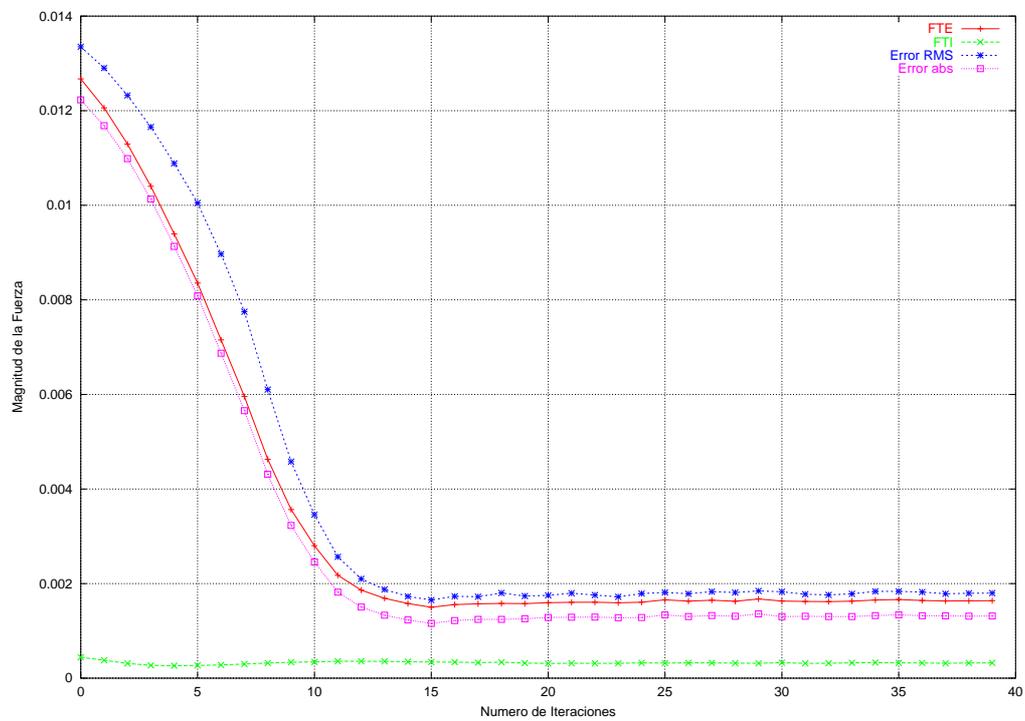


Figura 2.22: Gráfica de magnitud de fuerzas contra número de iteraciones.

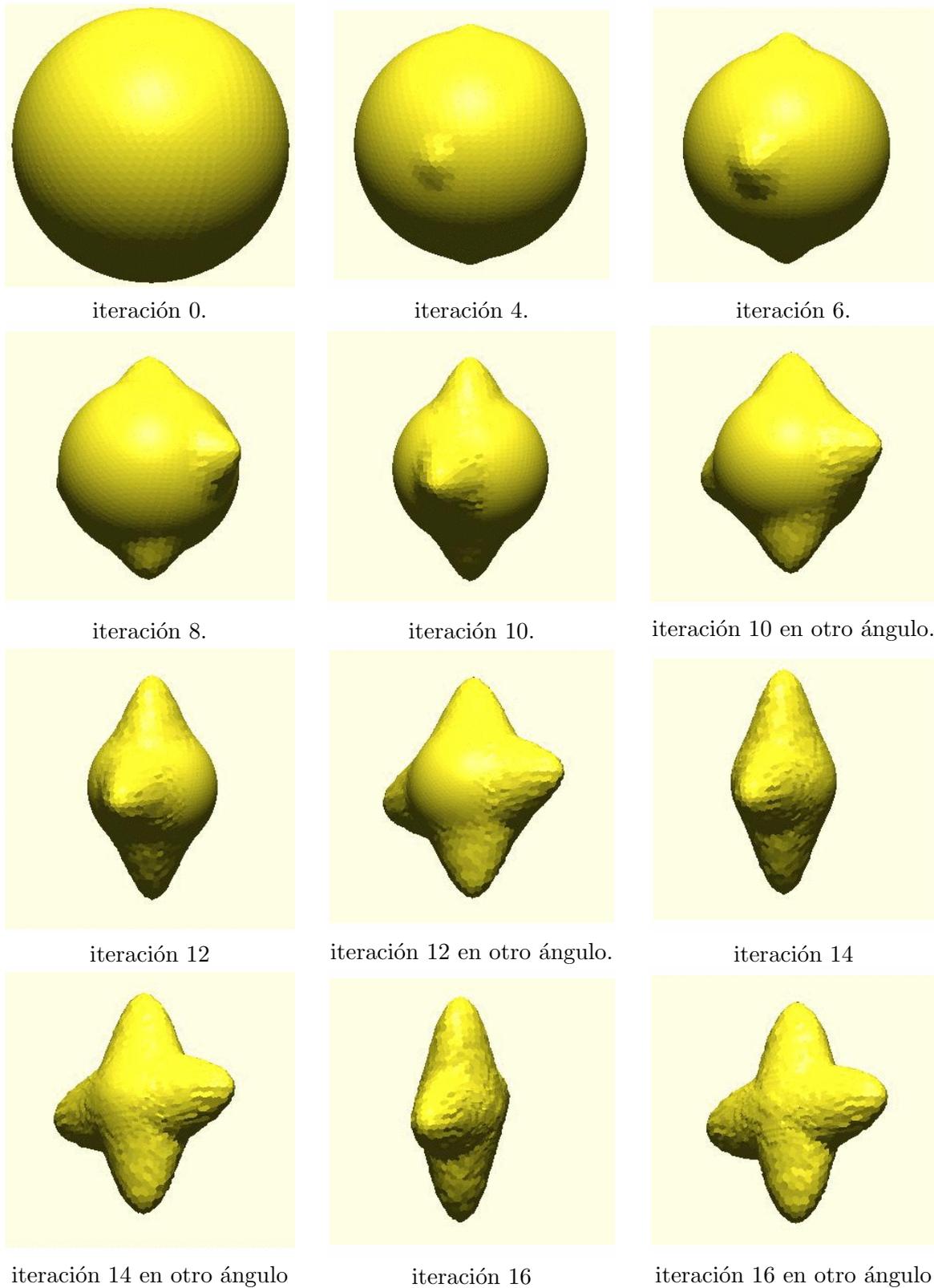


Figura 2.23: Iteraciones 0 a 16

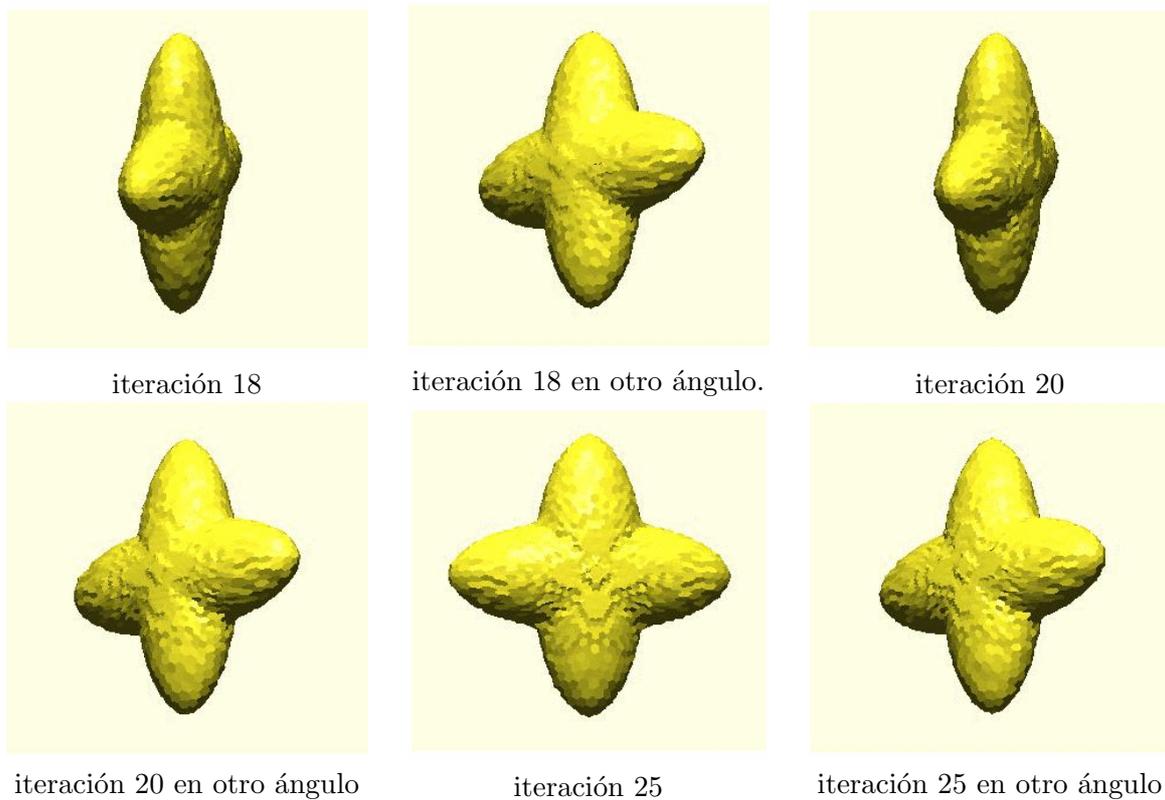


Figura 2.24: La figuras muestran los modelos correspondientes a la gráfica presentada en la figura 2.22, basado en el modelo presentado en la sección 3.1 del capítulo 3. Como se puede apreciar todo el proceso de reconstrucción se lleva a cabo entre las iteraciones 0 y 20, después el modelo mantiene una forma constante.

Capítulo 3

Reconstrucción 3D Iterativa con Distintos Modelos Iniciales

3.1. Generación del modelo de una macromolécula biológica

Para alimentar el algoritmo que se presento en la sección 1.3 del capítulo 1, en la pág. 1, se necesita un modelo inicial (que en este caso sería uno de los volúmenes obtenidos por medio de las mallas de simplejos), y un modelo final, debido a que en este trabajo esta enfocado a comprobar la robustez de un algoritmo de reconstrucción tridimensional se trabajara con una molécula sintética, la cual para fines prácticos es un volumen 3D modelado por medio del programa *phantom* de Xmipp [13] dentro del cual podemos generar uniones, y restas de tres distintos tipos de primitivas: cilindros, esferas, y elipsoides. El programa *phantom* nos arrojará un volumen situado en un espacio tridimensional de dimensiones $n \times m \times l$ (especificada por los parámetros), en donde nuestro volumen se sitúa. El siguiente texto es el contenido del archivo que sirve como ejemplo de la forma en la que se especifican los parámetros para crear un volumen basado en dos elipsoides, como se observa en la Fig. 3.1

```
#####Esp1.dat#####
# Phantom "Esp1.spi" description file, (generated with phantom)
# General Volume Parameters:
#      Xdim      Ydim      Zdim      Density
#      128       128       128       0.000000e+00
# Feature Parameters:
#Fea_N Sph_Cyl Or_And fDensity   iX_Cent iY_Cent iZ_Cent iX_Rad iY_Rad iZ_Rad iAlpha iBeta iGamma
0000   e       o       1.000000e+00  0.00   0.00   0.00  100.00  40.00  32.00 0.000 0.000 0.0000
0001   e       o       1.000000e+00  0.00   0.00   0.00  100.00  40.00  32.00 0.000 0.000 1.5708
```

Este modelo consiste en un conjunto de voxels en el espacio, el modelo generado es un sólido (como se ve en la figura 3.2 donde se muestran los cortes del Espécimen 1). Finalmente este modelo va ser utilizado como el modelo a reconstruir, de este se extraerán varias proyecciones (imágenes) y por medio de los modelos deformables se obtendrán los distintos modelos iniciales necesitados por el algoritmo de reconstrucción tridimensional.

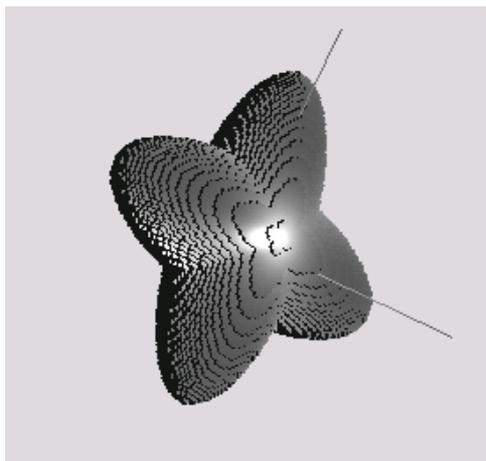


Figura 3.1: Modelo generado con el contenido del archivo *esp1.dat*.

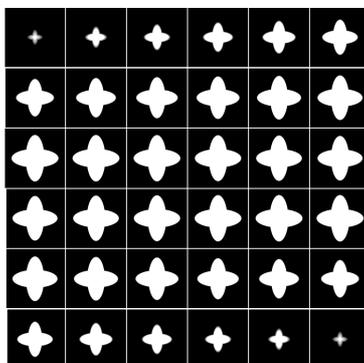


Figura 3.2: La visualización del Espécimen 1 por medio de rebanadas.

3.2. Deformación sucesiva del modelo de la macromolécula

Para el manejo de nuestro modelo deformable tratado en el capítulo 2 se creó una interfaz gráfica, dentro de la cual la tarea de crear y deformar un modelo inicial se vuelve algo relativamente sencillo. Nuestro programa de aplicación, llamado *Proto*, recibe como parámetros de entrada los siguientes datos:

- **Vol_fuente.raw**: Este es el volumen al cual nuestro modelo se va a aproximar, este debe de estar en formato raw o crudo, esto es una sucesión de bytes los cuales en su conjunto tienen el tamaño de una matriz tridimensional en donde los voxels con valor distinto a 0, serán voxels ocupados en el espacio.
- **Dimension_Cubica** (N^3): Son las dimensiones de nuestro espacio tridimensional en voxels, este está definido como un espacio de $N \times N \times N$ o N^3 , por lo cual sólo se debe de especificar el valor de N .
- **Umbral_voxels**: Es el valor que deben de sobrepasar los voxels para que sean detectados por nuestra aplicación, de lo contrario el programa chocara con un volumen vacío, por ejemplo si el valor que se le asigna a este parámetro es 50, los voxels que estén por debajo de este valor no serán tomados como parte del espacio.

- **Print_flag**: Si es 0, no imprime datos de salida, de lo contrario la información que se imprima puede ser utilizada para alimentar al programa “gnuplot” y obtener gráficas del comportamiento del modelo.
- **Vol_destino**: Es el prefijo que llevaran los volúmenes creados en formato raw a partir del modelo deformable que se este utilizando, de manera tal que si el valor del volumen destino es `vol_out`, los volúmenes que se generen a partir de nuestro modelo serán `vol_out0.raw`, `vol_out1.raw`, `vol_out2.raw`, etc.

Para la generación de un modelo inicial se ejecuta la herramienta **Proto**, los parámetros sugeridos para generar un volumen en un espacio cubico de 128^3 voxels imprimiendo información es:

```
Proto Esp2.raw 128 1 1 Esp2_out
```

Donde el espécimen a aproximar es `Esp2.raw`, y el prefijo de los nuevos volúmenes generados es `Esp2_out`. Al invocar a la herramienta **Proto** obtendremos la siguiente pantalla:

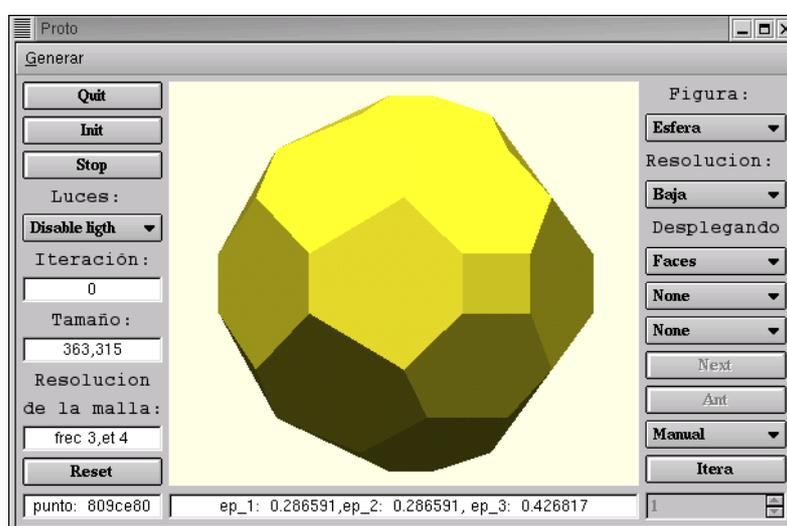


Figura 3.3: Pantalla de la herramienta Proto.

en donde podemos observar un poliedro el cual es una aproximación de una esfera en baja resolución. Como se mencionó en el capítulo 2 los posibles modelos iniciales son la esfera, el cilindro y el toroide, dentro del combobox *figura* se pueden escoger solamente dos de éstos debido a que dentro de esta tesis solamente se manejarán modelos con un *género* igual a cero. Dentro del combobox *Resolucion*, se pueden escoger entre tres posibles resoluciones del modelo, las cuales son: baja, media y alta, de las cuales se proporciona información de la malla en el textbox **Resolucion de la malla**.

Una vez que se ha escogido con cuál resolución se quiere aproximar al espécimen, se empieza a iterar con el botón “*itera*”, cabe mencionar que entre mas grande sea la resolución más vértices contiene el modelo, entre más alto sera el número de vértices más tiempo se tardará en procesar cada iteración, lo más recomendable para cualquier caso es elegir la resolución media, pero al elegirse la resolución alta se obtendrán modelos más detallados, también se le puede indicar al programa que haga varias iteraciones seguidas por medio del combobox situado arriba del botón “*itera*”, normalmente éste se encuentra en la opción “*Manual*”, pero si se escoge la opción “*n_iter*” el SpinBox que está debajo del botón “*itera*” se activará dejándonos ingresar un número determinado de iteraciones

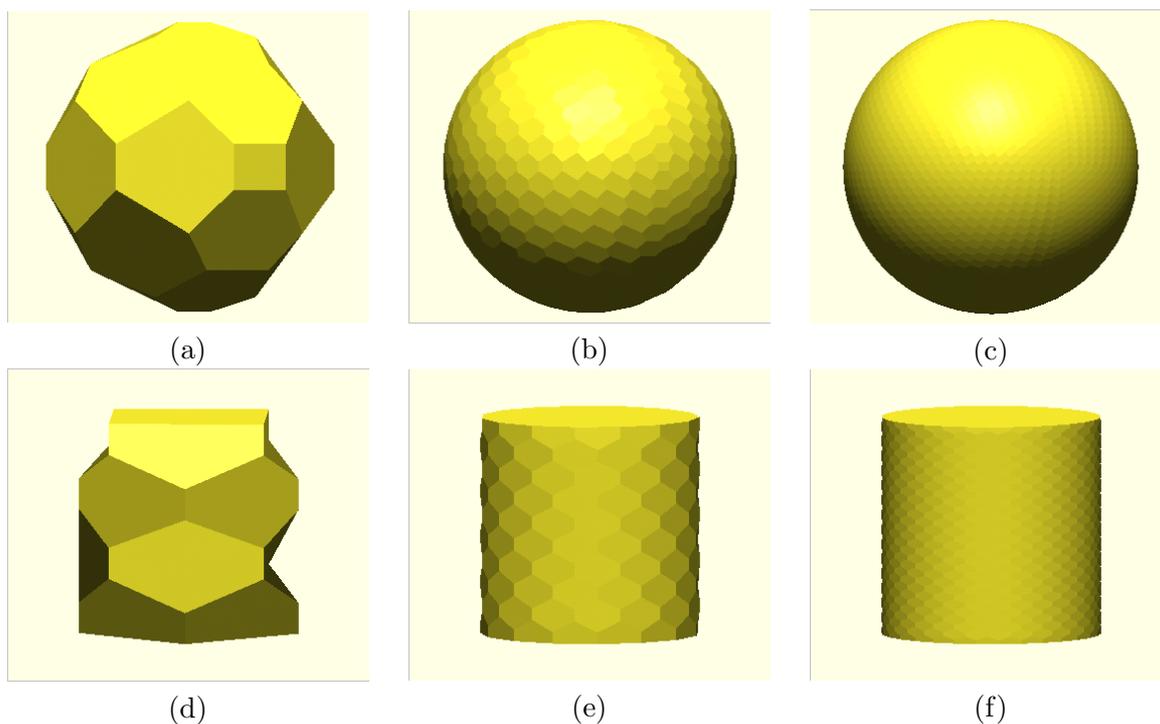


Figura 3.4: (a),(b) y (c) modelos de la esfera a baja, media y alta resolución. (d),(e) y (f) modelos del cilindro a baja, media y alta resolución.

sucesivas. Para que una aproximación se lleve a cabo sólo es necesario un determinado número de iteraciones, este número depende de cada modelo a aproximar y básicamente la manera de proceder adecuada es el imprimir la información de salida que nos proporcione el programa “Proto”, y cuando la magnitud de las fuerzas comience a oscilar será el tiempo propicio para detener al programa. Casi todos los modelos utilizan un número de entre 20 y 35 iteraciones para obtener un modelo muy aproximado al modelo final.

Para la condición de paro del algoritmo, creamos la medida del *porcentaje de similitud*, definida por la Ec. (3.1)

$$s_{\text{abs}}(i) = 1 - \frac{\text{error}_{\text{abs}}^i}{\text{error}_{\text{MAX}}} \quad (3.1)$$

En donde:

- $\text{error}_{\text{abs}}^i$ es el error absoluto en la iteración i calculado en la ecuación (2.38), pág. 42.
- $\text{error}_{\text{MAX}}$ es igual a error absoluto en la iteración 0, $\text{error}_{\text{abs}}^0$.

Aunque también podríamos haber usado el valor del error RMS (en la Ec. (2.37)) de la forma

$$s_{\text{RMS}}(i) = 1 - \frac{\text{error}_{\text{RMS}}^i}{\text{error}_{\text{MAX}}} \quad (3.2)$$

donde $\text{error}_{\text{MAX}}$ sería igual al error RMS en la iteración 0.

iteración	FTE	FTI	Error abs	s_{abs}
0	0.028189	0.000902	0.028297	0.027287
1	0.028136	0.000796	0.028243	0.027340
2	0.027966	0.000676	0.028095	0.027290
3	0.027662	0.000576	0.027859	0.027086
4	0.027216	0.000506	0.027536	0.026710
5	0.026593	0.000462	0.027104	0.026131
6	0.025780	0.000437	0.026564	0.025343
7	0.024803	0.000426	0.025902	0.024377
8	0.023498	0.000425	0.024978	0.023073
9	0.021814	0.000425	0.023721	0.021389
10	0.019555	0.000429	0.021956	0.019126
11	0.016943	0.000438	0.019786	0.016505
12	0.013871	0.000443	0.016879	0.013428
13	0.010302	0.000451	0.012774	0.009851
14	0.007478	0.000460	0.009272	0.007018
15	0.005443	0.000472	0.006699	0.004972
16	0.004062	0.000482	0.004932	0.003581
17	0.003032	0.000492	0.003617	0.002540
18	0.002406	0.000495	0.002893	0.001911
19	0.001921	0.000487	0.002219	0.001434
20	0.001675	0.000465	0.001949	0.001211
21	0.001607	0.000446	0.001929	0.001161
22	0.001554	0.000433	0.001783	0.001121
23	0.001582	0.000415	0.001779	0.001167
24	0.001609	0.000400	0.001779	0.001209
25	0.001640	0.000394	0.001819	0.001246
26	0.001645	0.000383	0.001849	0.001262
27	0.001602	0.000380	0.001775	0.001222
28	0.001606	0.000377	0.001781	0.001229
29	0.001643	0.000369	0.001802	0.001274
30	0.001628	0.000367	0.001792	0.001261
31	0.001668	0.000361	0.001850	0.001307

Cuadro 3.1: Magnitudes de las fuerzas totales, el error absoluto y el porcentaje de similitud.

iteración	s_{RMS}	s_{abs}
0	0.00	0.00
1	0.19	-0.19
2	0.71	-0.01
3	1.55	0.74
4	2.69	2.11
5	4.22	4.24
6	6.12	7.12
7	8.46	10.66
8	11.73	15.44
9	16.17	21.61
10	22.41	29.91
11	30.08	39.51
12	40.35	50.79
13	54.86	63.90
14	67.23	74.28
15	76.33	81.78
16	82.57	86.88
17	87.22	90.69
18	89.78	93.00
19	92.16	94.74
20	93.11	95.56
21	93.18	95.75
22	93.70	95.89
23	93.71	95.72
24	93.71	95.57
25	93.57	95.43
26	93.47	95.38
27	93.73	95.52
28	93.71	95.50
29	93.63	95.33
30	93.67	95.38
31	93.46	95.21

Cuadro 3.2: Porcentaje de similitud entre el modelo inicial y el final, según el número de iteración.
El porcentaje de similitud fue calculado según las ecuaciones (3.1) y (3.2).

3.3. Generación del volumen a partir de la malla deformada

Una vez que se ha generado un modelo intermedio entre el inicial y el final, debe de generarse un archivo el cual contenga la información de este volumen en formato raw.

Para trasladar una malla de simplejos a una imagen volumétrica hay que trasladar los vértices, aristas y caras de nuestro espacio *continuo* a la representación de imágenes volumétricas. El espacio manejado dentro de las imágenes volumétricas se le puede ver como un cubo o un poliedro el cual puede subdividirse en pequeños cubos llamados voxels (como se muestra en la figura 3.5), estos al igual que los pixels en imágenes son los componentes más pequeños e indivisibles de nuestro espacio de trabajo, el proceso para transformar nuestra malla en un modelo volumétrico es discretizar el espacio en el que se está trabajando, para hacer esto se debe calcular la posición de cada voxel dentro del espacio, lo cual es bastante sencillo. Una ves que se ha discretizado el espacio en una

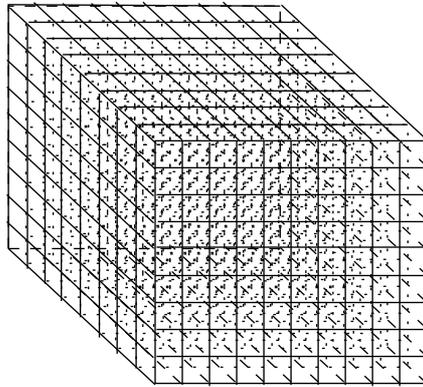


Figura 3.5: El espacio de trabajo discretizado, cada cubo dentro de éste es un voxel.

matriz cubica de $N \times N \times N$ el modelo de la malla también debe de ser discretizado a voxels, esto es aparentemente muy simple, pero si partimos del hecho del que la malla ya es una discretización de un modelo matemático, el cual se basa en tener guardada información de solamente algunos vértices y su relación entre ellos para poder construir un modelo tridimensional, entonces asumiremos que tenemos no tenemos disponible la información de la superficie de manera explícita. En un modelo continuo la información de la superficie está especificada por medio de la ecuación que lo define, en este caso para discernir que voxels están siendo ocupados por la superficie de la malla hemos diseñado el siguiente procedimiento:

1. Los únicos puntos que se tiene guardados como información son los vértices de la malla, se calculan dentro de cuales voxels están y estos se crean (fig 3.6 (a)).
2. Una vez que se tienen todos los vértices discretizados se calcula el centroide de cada una de las caras de la malla y se crea un voxel para este (fig 3.6 (a)).
3. Se crean los voxels para cada arista de la malla (fig 3.6 (b)).
4. Una vez que se tienen discretizadas los vértices, las arista y los centroides da cada una de las caras se traza una línea tridimensional de cada uno de los voxels creados al centroide de cada una de las caras de la malla (fig 3.6 (b) y (c)).

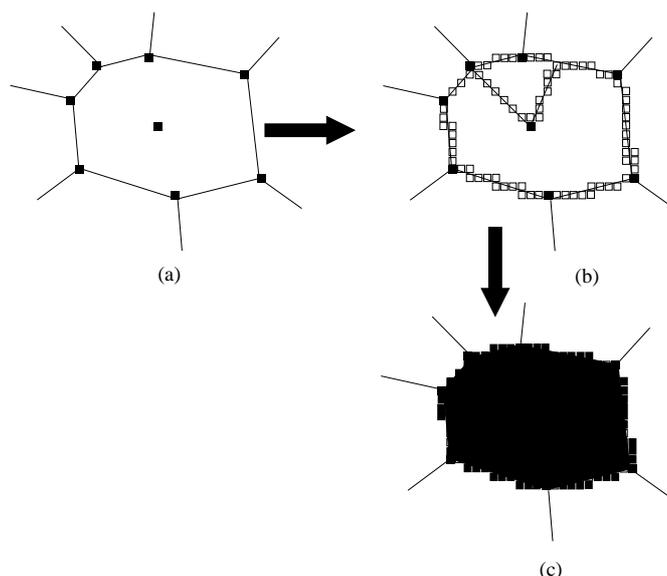


Figura 3.6: El proceso de discretización de la malla.

El volumen que se obtiene después de esta discretización es solamente la cubierta convexa del volumen, o lo que en lenguaje coloquial se conoce como el cascarón, obteniendo así un volumen hueco. Debido a que el algoritmo mostrado en la sección 1.3 del capítulo 1 necesita un volumen sólido, se necesita que el volumen obtenido de la aproximación sea rellenado, para este fin se ha utilizado la expansión a tres dimensiones del algoritmo de inundación comúnmente utilizado para imágenes, como ya es bien sabido el algoritmo de inundación es recursivo, al estar trabajando en un espacio tridimensional la complejidad que se eleva al cubo obteniendo como resultado un algoritmo poco óptimo para su utilización en máquinas con poca memoria, para resolver este problema se usó una implementación iterativa del algoritmo de inundación haciendo uso de una pila para sustituir el entorno recursivo. Dentro de la herramienta “Proto” el generar un volumen de salida es sencillo, simplemente se escoge del menú “Generar” la opción “imagen volumétrica” obteniendo con esto la imagen volumétrica del modelo que se está desplegando actualmente en la pantalla, una vez que se ha generado la imagen volumétrica en el mismo menú “Generar” se escoge la opción “Salida a Archivo” obteniendo con esto una imagen volumétrica en formato raw la cual se puede visualizar por medio del programa “scubes” [14] el cual nos ayuda a visualizar volúmenes en formato crudo.

Una vez que se ha generado el volumen en formato raw se pasa al formato spi de spider por medio del programa “spi2raw”, el cual es una adición al paquete Xmipp [13], para su posterior procesamiento. Este proceso se repite con todos los volúmenes intermedios, para alimentar al algoritmo de reconstrucción tridimensional, partiendo de cada uno de estos como modelo inicial.

3.4. Pruebas con el algoritmo iterativo de reconstrucción 3D

Para generar las pruebas con el algoritmo nos apoyamos en los programas realizados en el transcurso de la tesis del Doctor L.G de la Fraga [1], con lo cual para implementar

3.4. PRUEBAS CON EL ALGORITMO ITERATIVO DE RECONSTRUCCIÓN 3D 55

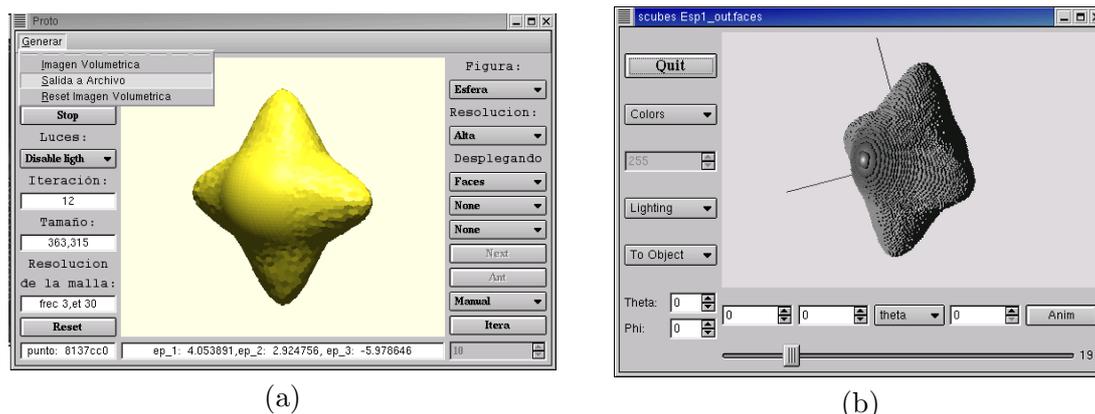


Figura 3.7: (a) El modelo intermedio obtenido en la iteración 12 de la herramienta “Proto”.(b) La imagen volumétrica obtenida en (a) visualizada con la herramienta “Scubes”.

el algoritmo de reconstrucción 3D mencionado en el capítulo 2 se debe de contar con un conjunto de proyecciones simulando las proyecciones del espécimen obtenidas del MET, para simular este estado se genero una lista de ángulos para obtener un conjunto de proyecciones de nuestro modelo original, se les agrega ruido y finalmente les cambiamos el ángulo a estas proyecciones por un valor aleatorio, tal y como se tendría en un conjunto de proyecciones extraídas de las micrografías generadas por el MET. En la practica este proceso es elaborado apoyándose en los siguientes programas:

- voea. Para generar una lista de ángulos de Euler
- proj3d. Para realizar las proyecciones
- addfnl. Para adicionar un ruido aditivo gaussiano a las proyecciones
- randangs.pl. Para generar una lista de ángulos de Euler con valores aleatorios.
- swriteangs. Para escribir los ángulos a las proyecciones.

Teniendo la fase preliminar completada se seguirá el algoritmo de reconstrucción 3D, a continuación presentaremos cada paso de este y el proceso seguido para su realización:

1. *Se les asigna los parámetros, de translación y orientación a las proyecciones del modelo real (la macromolécula en este caso).* Este proceso se logra a través de los siguientes programas de forma consecutiva:
 - refin3dsec
 - voea
 - vomdok
 - swriteangs

en donde refin3Dsec se ocupa de asignar la orientación de acuerdo al volumen del espécimen sintético, voea y vomdoc generan nuevos parámetros de orientación, y swriteang de introducir la información en las proyecciones

2. *Se realiza la reconstrucción en 3D.* Una vez que se tienen las diferentes proyecciones con los nuevos parámetros se procede a crear el nuevo modelo por medio de los programas:

- artkbcc.
- voxels.
- binvolume.

En donde artkbcc se encarga de hacer la reconstrucción 3D aplicando el algoritmo ART, dando como salida un modelo con voxels en diferentes valores de número flotante, voxels se encarga de informar cuantos voxels existe por debajo de un determinado valor, y binvolume se encarga de binarizar un volumen 3D dado un umbral. Lo que se busca al aplicar estos tres programas en secuencia es obtener un volumen reconstruido con la misma masa que el espécimen sintético, para esto se aplica el programa voxels, y una vez que se tiene el valor por medio del cual la masa del espécimen sintético y la reconstrucción son iguales, se alimenta a binvolume con este para obtener un modelo tridimensional binarizado con el cual es fácil trabajar.

3. *Se tiene de un modelo 3D de referencia.* Cuando este completa la reconstrucción 3D el nuevo modelo sera ahora nuestro modelo de referencia, de manera tal que este sustituye al modelo anterior creado por las mallas de simplejos.
4. *Se proyecta el modelo 3D en todas direcciones en el espacio para comparar las proyecciones experimentales.* Una vez que tenemos nuestro nuevo modelo de referencia se proyecta en todas las direcciones por medio del programa:

- refin3dsec

5. *Por cada proyección experimental se compara con las proyecciones de el modelo real, de esta forma los parámetros que se ajustan mejor las comparaciones resultan en la asignación de nuevos parámetros para las proyecciones.* En este paso nos auxiliaremos de los programas:

- vomdok
- swriteangs

En donde vomdok se encargara de asignar los parámetros y swriteangs de introducir los cambios en las proyecciones.

6. *Calculamos las diferencias entre las proyecciones del modelo actual y las proyecciones experimentales, si la suma del cuadrado de las diferencias es más grande que un valor predefinido, se vuelve al paso 2.* La comparación de las proyecciones se hace por medio del script :

- threshold.pl

El cual nos da una salida con la cuenta del número de proyecciones que tienen una diferencia de n grados ordenándolas de 5 en 5 grados.

3.4.1. Pruebas

Las pruebas se realizaron con tres distintos modelos, de los cuales dos tienen simetría cilíndrica de nivel 3 y 4 (como se puede ver en la figura 3.8) y otro carece completamente de ésta.

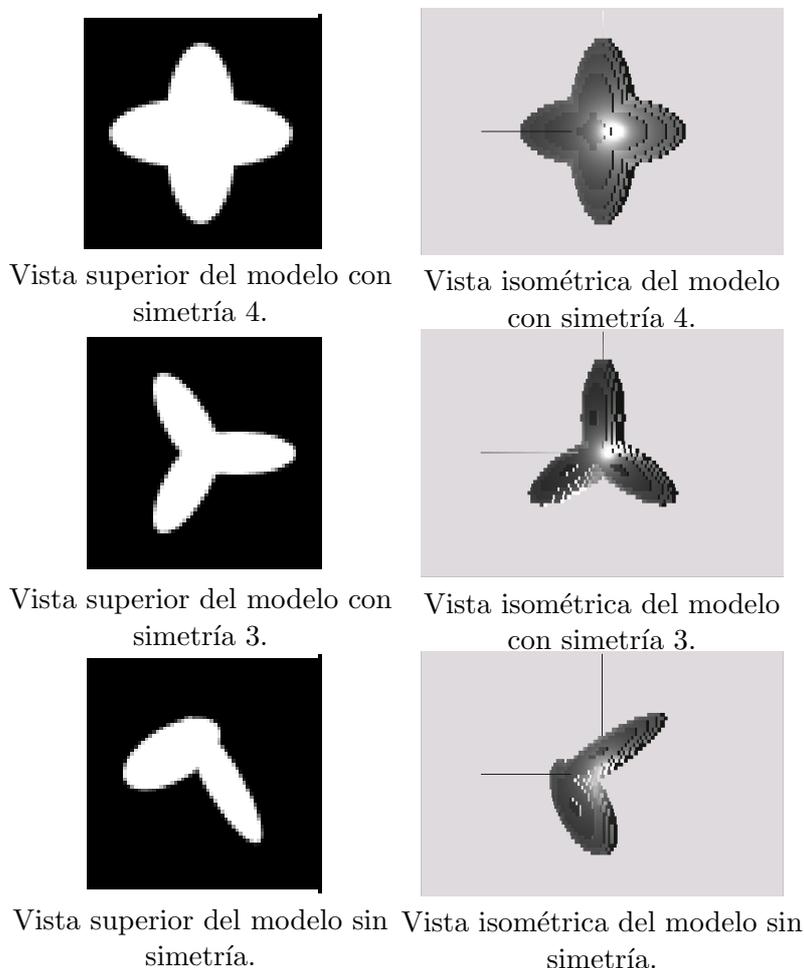


Figura 3.8: Los distintos modelos a los que ha sido sometido el algoritmo de reconstrucción 3D.

Para efectos de esta tesis, debido a las consideraciones tomadas en su planteamiento se decidió utilizar como medida de decisión el porcentaje de similitud extraído de la ecuación 3.2, para todos los modelos.

En dos de los tres modelos (específicamente en los modelos con simetría 3 y 4) se hicieron las pruebas de la siguiente forma:

Se alimentó el algoritmo de reconstrucción 3D con 7 modelos, el primero en todos los casos es la esfera, el séptimo es el mismo modelo de espécimen, y los cinco modelos intermedios restantes se sacan por medio de nuestro algoritmo de reconstrucción 3D por medio de mallas de simplejos, eligiendo de la gráfica generada en el proceso de reconstrucción 5 modelos con distintos grados de deformación. A cada uno de los modelos intermedios se les aplican 3 iteraciones del algoritmo de reconstrucción 3D y se analizan los resultados para establecer a partir de que porcentaje de similitud el algoritmo converge. Para el caso del modelo sin simetría solo se aplicaran las pruebas tomando como modelos de referencia a la esfera y dos de los cinco modelos intermedios para establecer que grado de similitud

es suficiente para que se logre una reconstrucción exitosa.

Modelo con simetría cuatro

El modelo con simetría igual a cuatro fue generado por el programa “phantom” a partir del siguiente archivo:

```
#####Esp1_64.dat#####
# Phantom "Esp1_64.spi" description file, (generated with phantom)
# General Volume Parameters:
#   Xdim   Ydim   Zdim   Density
#   64     64     64     0.000000e+00
# Feature Parameters:
#Fea_N Sph_Cyl Or_And fDensity   iX_Cent iY_Cent iZ_Cent iX_Rad iY_Rad iZ_Rad iAlpha iBeta iGamma
0000   e     o   1.000000e+00   0.00  0.00  0.00   50.00  20.00  10.00  0.000 0.000 0.0000
0001   e     o   1.000000e+00   0.00  0.00  0.00   50.00  20.00  10.00  0.000 0.000 1.5708
```

el cual al ser reconstruido por medio de mallas de simplejos nos da una gráfica y la tabla de salida 3.3, con los niveles de deformación que son:

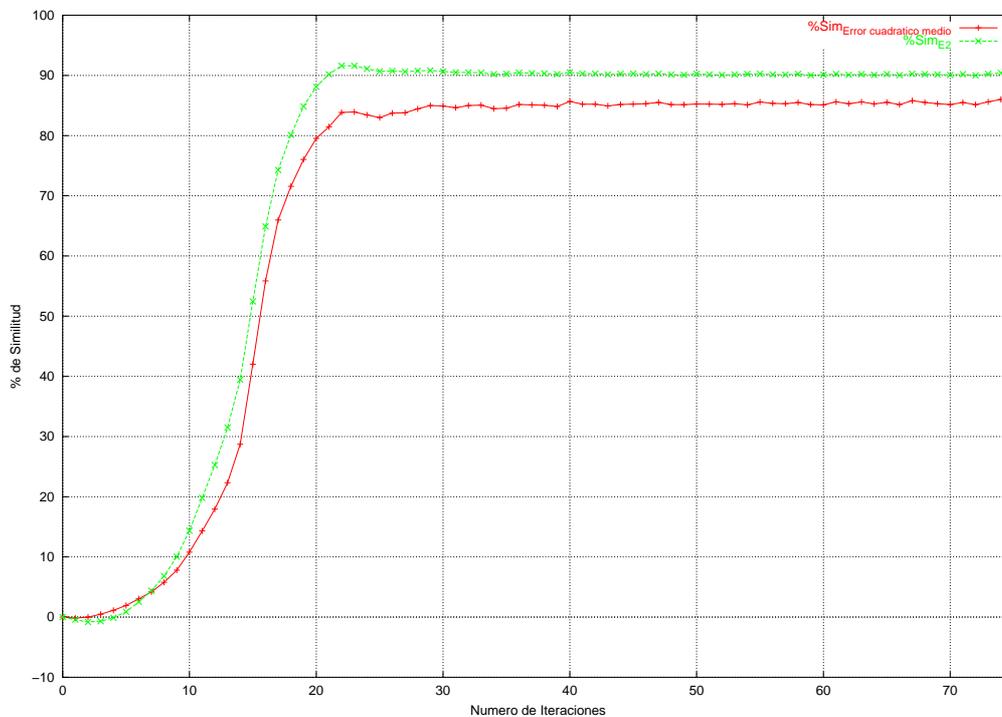


Figura 3.9: Gráfica derivada de la tabla 3.3.

En la tabla 3.3 podemos observar claramente que el porcentaje de similitud más alto es de aproximadamente 90.5%, para escoger 5 modelos de forma equitativa dividimos este número entre 5 y se escogen los modelos más aproximados al número resultado de esta división. Para este caso se han escogido los modelos correspondientes a la iteraciones: 11, 14, 15, 17, 74 (los cuales se pueden apreciar en la figura 3.10)

A continuación se presenta los resultados de cada iteración, para cada modelo se sacaron 334 proyecciones en donde para cada el parámetro e significa el error por diferencia de ángulos, en donde esas diferencias son el resultado de un histograma de las diferencias en el ángulo de inclinación, n el número de proyecciones con ese valor de error para cada diferencia y $n/N(\%)$ el porcentaje de voxels del número total de voxels en el modelo.

3.4. PRUEBAS CON EL ALGORITMO ITERATIVO DE RECONSTRUCCIÓN 3D 59

iteración	s_{abs}
0	0.00
1	-0.43
2	-0.82
3	-0.69
4	-0.11
5	0.88
6	2.52
7	4.37
8	6.82
9	10.04
10	14.36
11	19.82
12	25.27
13	31.50
14	39.45
15	52.44
16	64.91
17	74.28
18	80.10
19	84.85
20	88.12
21	90.18
22	91.61
23	91.59
24	91.10
25	90.67
26	90.74
27	90.61
28	90.75
29	90.82
30	90.66
31	90.50
32	90.49
33	90.46
34	90.21
35	90.25
36	90.46
37	90.36
38	90.33
39	90.21

iteración	s_{abs}
40	90.50
41	90.30
42	90.29
43	90.15
44	90.26
45	90.30
46	90.19
47	90.29
48	90.14
49	90.09
50	90.27
51	90.15
52	90.08
53	90.16
54	90.23
55	90.27
56	90.15
57	90.13
58	90.25
59	90.03
60	90.11
61	90.24
62	90.09
63	90.20
64	90.09
65	90.22
66	90.05
67	90.26
68	90.20
69	90.15
70	90.08
71	90.20
72	90.00
73	90.25
74	90.41

Cuadro 3.3: Porcentaje de similitud entre el modelo inicial y el final del modelo con simetría cuatro.

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	84	25.15	145	43.41	104	31.14
10	56	16.77	77	23.05	104	31.14
15	61	18.26	55	16.47	61	18.26
20	30	8.98	15	4.49	22	6.59
25	26	7.78	12	3.59	19	5.69
30	23	6.89	7	2.10	9	2.69
35	28	8.38	7	2.10	8	2.40
40	12	3.59	4	1.20	3	0.90
45	12	3.59	2	0.60	3	0.90
50	0	0.00	5	1.50	0	0.00
55	2	0.60	1	0.30	1	0.30
60	0	0.00	2	0.60	0	0.00
65	-	-	0	0.00	-	-
70	-	-	1	0.30	-	-
75	-	-	1	0.30	-	-
80	-	-	0	0.00	-	-

Cuadro 3.4: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial esfera.

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	26	7.78	246	73.65	315	94.31
10	46	13.77	20	5.99	13	3.89
15	48	14.37	20	5.99	3	0.90
20	36	10.78	14	4.19	2	0.60
25	35	10.48	21	6.29	1	0.30
30	35	10.48	4	1.20	0	0.00
35	42	12.57	5	1.50	-	-
40	22	6.59	1	0.30	-	-
45	22	6.59	2	0.60	-	-
50	15	4.49	1	0.30	-	-
55	4	1.20	0	0.00	-	-
60	0	0.00	-	-	-	-
65	2	0.60	-	-	-	-
70	1	0.30	-	-	-	-
75	0	0.00	-	-	-	-
No. Proy	334	-	334	-	334	-

Cuadro 3.5: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial 1.

3.4. PRUEBAS CON EL ALGORITMO ITERATIVO DE RECONSTRUCCIÓN 3D61

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	78	23.35	284	85.03	305	91.32
10	63	18.86	26	7.78	19	5.69
15	81	24.25	8	2.40	8	2.40
20	34	10.18	7	2.10	2	0.60
25	32	9.58	7	2.10	0	0.00
30	19	5.69	2	0.60	-	-
35	18	5.39	0	0.00	-	-
40	2	0.60	-	-	-	-
45	4	1.20	-	-	-	-
50	3	0.90	-	-	-	-
55	0	0.00	-	-	-	-

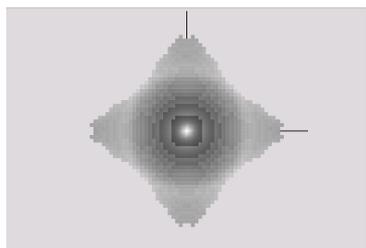
Cuadro 3.6: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial 2.

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	90	26.95	290	86.83	307	91.92
10	110	32.93	25	7.49	15	4.49
15	64	19.16	13	3.89	6	1.80
20	32	9.58	4	1.20	4	1.20
25	23	6.89	1	0.30	1	0.30
30	9	2.69	1	0.30	1	0.30
35	4	1.20	0	0.00	0	0.00
40	1	0.30	-	-	-	-
45	0	0.00	-	-	-	-
50	1	0.30	-	-	-	-
55	0	0.00	-	-	-	-

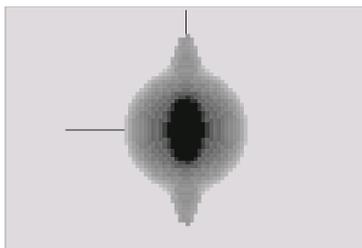
Cuadro 3.7: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial 3.

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	178	53.29	309	92.51	306	91.62
10	98	29.34	13	3.89	18	5.39
15	42	12.57	9	2.69	8	2.40
20	11	3.29	2	0.60	1	0.30
25	5	1.50	1	0.30	1	0.30
30	0	0.00	0	0.00	0	0.00

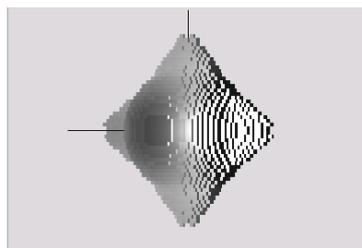
Cuadro 3.8: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial 4.



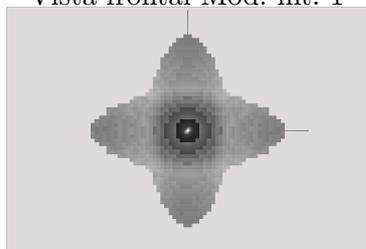
Vista frontal Mod. int. 1



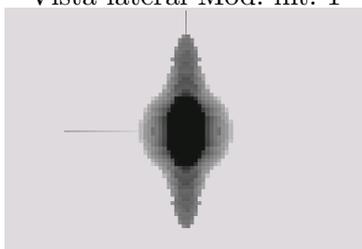
Vista lateral Mod. int. 1



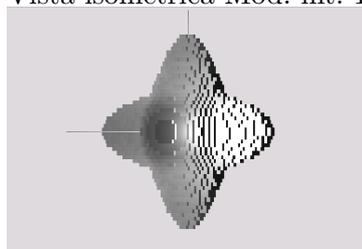
Vista isométrica Mod. int. 1



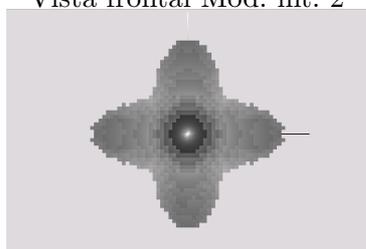
Vista frontal Mod. int. 2



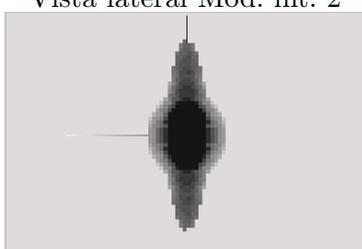
Vista lateral Mod. int. 2



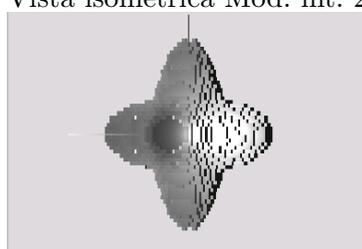
Vista isométrica Mod. int. 2



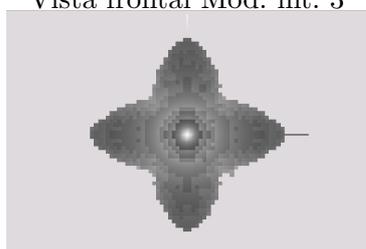
Vista frontal Mod. int. 3



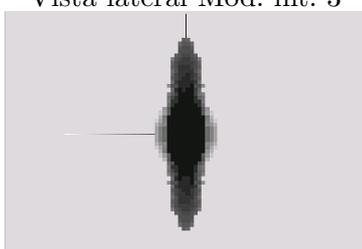
Vista lateral Mod. int. 3



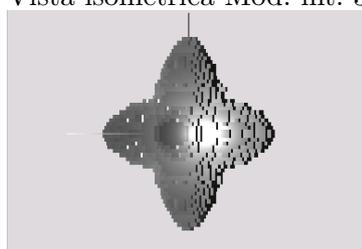
Vista isométrica Mod. int. 3



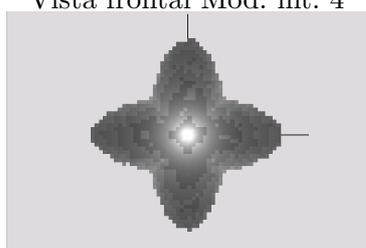
Vista frontal Mod. int. 4



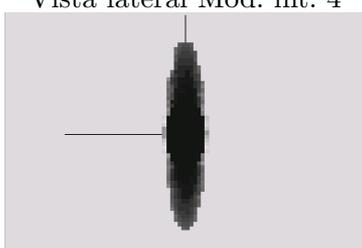
Vista lateral Mod. int. 4



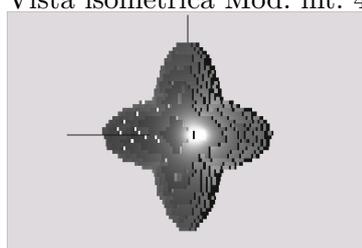
Vista isométrica Mod. int. 4



Vista frontal Mod. int. 5



Vista lateral Mod. int. 5



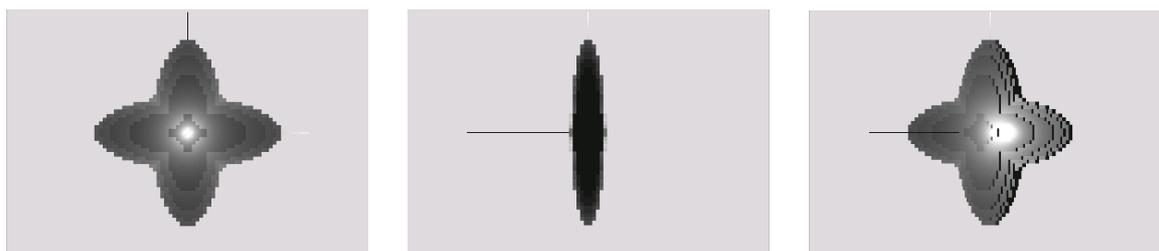
Vista isométrica Mod. int. 5

Modelo con simetría tres

Para el modelo con simetría tres se utilizo el siguiente archivo para generarlo:

```
#####Esp4_64.dat#####
# Phantom "Esp4_64.spi" description file, (generated with phantom)
# General Volume Parameters:
#   Xdim   Ydim   Zdim   Density
#     64     64     64   0.000000e+00
# Feature Parameters:
#Fea_N   Sph_Cyl Or_And fDensity   iX_Cent iY_Cent iZ_Cent iX_Rad iY_Rad iZ_Rad iAlpha iBeta iGamma
0000     e     o   1.000000e+00   18.0   0.00   0.00   32.00 12.0   10.00 0.000 0.000 0.0000
```

3.4. PRUEBAS CON EL ALGORITMO ITERATIVO DE RECONSTRUCCIÓN 3D 63



Vista frontal Mod. original. Vista lateral Mod. original. Vista isométrica Mod. original.

Figura 3.10: Modelos intermedios obtenidos de las iteraciones al aplicar el algoritmo de reconstrucción basado en mallas de simplejos.

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	248	74.25	308	92.22	312	93.41
10	59	17.66	17	5.09	13	3.89
15	15	4.49	8	2.40	6	1.80
20	6	1.80	1	0.30	3	0.90
25	5	1.50	0	0.00	0	0.00
30	1	0.30	-	-	-	-
35	0	0.00	-	-	-	-

Cuadro 3.9: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo inicial 5.

```
0001    e    o    1.000000e+00    -9.0    15.60    0.00    32.00    12.0    10.00    0.000    0.000    2.09440
0002    e    o    1.000000e+00    -9.0    -15.60    0.00    32.00    12.0    10.00    0.000    0.000    4.18880
```

con la gráfica 3.11 y tabla de salida 3.11 con los niveles de deformación.

En donde se puede observar que el nivel máximo de similitud es nuevamente 90.5 % por lo que cada incremento sera de 18.1, escogiendo los modelos que tengan los porcentajes más aproximados, donde para este caso serán los modelos con las iteraciones: 13, 15, 16, 18, 74.

Ahora veremos los resultados de cada iteración, para cada modelo se obtuvieron 441 proyecciones indicando los resultados de manera idéntica a las sección 3.4.1.

Modelo sin simetría

El modelo sin simetría fue creado con el propósito de probar en que medida el algoritmo de reconstrucción es dependiente de la simetría del modelo, o si este solamente necesita

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	271	81.14	311	93.11	318	95.21
10	37	11.08	18	5.39	11	3.29
15	18	5.39	5	1.50	4	1.20
20	4	1.20	0	0.00	0	0.00
25	4	1.20	-	-	-	-
30	0	0.00	-	-	-	-

Cuadro 3.10: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 334 en el modelo original.

iteración	s_{abs}
0	0.00
1	-0.46
2	-0.87
3	-0.97
4	-0.83
5	-0.30
6	0.53
7	1.42
8	2.79
9	4.49
10	6.56
11	9.16
12	12.42
13	17.05
14	23.78
15	35.06
16	50.22
17	63.31
18	72.34
19	78.82
20	84.22
21	87.91
22	90.41
23	91.78
24	91.72
25	91.62
26	91.22
27	91.04
28	91.18
29	90.83
30	90.87
31	90.91
32	90.60
33	90.57
34	90.35
35	90.35
36	90.22
37	90.42
38	90.32
39	90.07

iteración	s_{abs}
40	90.31
41	90.17
42	90.33
43	90.26
44	90.04
45	90.25
46	90.20
47	90.14
48	90.25
49	90.44
50	90.16
51	90.15
52	90.21
53	90.46
54	90.22
55	90.14
56	90.12
57	90.21
58	90.16
59	90.10
60	90.22
61	90.14
62	90.08
63	90.16
64	90.10
65	90.07
66	90.09
67	90.00
68	90.09
69	90.18
70	89.93
71	89.75
72	90.14
73	90.21
74	90.21

Cuadro 3.11: Porcentaje de similitud entre el modelo inicial y el final del modelo con simetría tres.

3.4. PRUEBAS CON EL ALGORITMO ITERATIVO DE RECONSTRUCCIÓN 3D 65

e	Iteración 1		Iteración 2		Iteración 3	
	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	136	30.84	170	38.55	116	26.30
10	88	19.95	81	18.37	56	12.70
15	72	16.33	95	21.54	71	16.10
20	38	8.62	30	6.80	35	7.94
25	45	10.20	32	7.26	48	10.88
30	16	3.63	10	2.27	37	8.39
35	22	4.99	12	2.72	22	4.99
40	12	2.72	5	1.13	14	3.17
45	4	0.91	2	0.45	13	2.95
50	2	0.45	1	0.23	6	1.36
55	3	0.68	1	0.23	6	1.36
60	2	0.45	0	0.00	8	1.81
65	0	0.00	2	0.45	5	1.13
70	1	0.23	-	-	1	0.23
75	-	-	-	-	1	0.23
80	-	-	-	-	1	0.23
85	-	-	-	-	1	0.23

Cuadro 3.12: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial esfera.

e	Iteración 1		Iteración 2		Iteración 3	
	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	207	46.94	406	92.06	406	92.06
10	113	25.62	26	5.90	25	5.67
15	85	19.27	6	1.36	10	2.27
20	22	4.99	2	0.45	-	-
25	13	2.95	1	0.23	-	-
30	1	0.23	-	-	-	-

Cuadro 3.13: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial 1.

e	Iteración 1		Iteración 2		Iteración 3	
	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	280	63.49	410	92.97	411	93.20
10	93	21.09	20	4.54	19	4.31
15	55	12.47	7	1.59	10	2.27
20	10	2.27	3	0.68	0	0.00
25	2	0.45	1	0.23	1	0.23
30	0	0.00	-	-	-	-
35	0	0.00	-	-	-	-
40	1	0.23	-	-	-	-
45	0	0.00	-	-	-	-

Cuadro 3.14: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial 2.

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	360	81.63	402	91.16	412	93.42
10	62	14.06	28	6.35	11	2.49
15	16	3.63	11	2.49	15	3.40
20	0	0.00	-	-	3	0.68
25	3	0.68	-	-	-	-
30	-	-	-	-	-	-

Cuadro 3.15: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial 3.

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	390	88.44	409	92.74	409	92.74
10	36	8.16	23	5.22	23	5.22
15	8	1.81	7	1.59	2	0.45
20	4	0.91	2	0.45	1	0.23
25	1	0.23	-	-	-	-
30	2	0.45	-	-	-	-
35	-	-	-	-	-	-

Cuadro 3.16: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial 4.

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	396	89.80	413	93.65	410	92.97
10	32	7.26	17	3.85	20	4.54
15	8	1.81	7	1.59	10	2.27
20	1	0.23	4	0.91	1	0.23
25	2	0.45	-	-	-	-
30	2	0.45	-	-	-	-
35	-	-	-	-	-	-

Cuadro 3.17: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo inicial 5.

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	395	89.57	401	90.93	422	95.69
10	34	7.71	27	6.12	9	2.04
15	9	2.04	8	1.81	8	1.81
20	1	0.23	3	0.68	1	0.23
25	2	0.45	2	0.45	0	0.00
30	-	-	-	-	1	0.23

Cuadro 3.18: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 441 en el modelo original.

3.4. PRUEBAS CON EL ALGORITMO ITERATIVO DE RECONSTRUCCIÓN 3D67

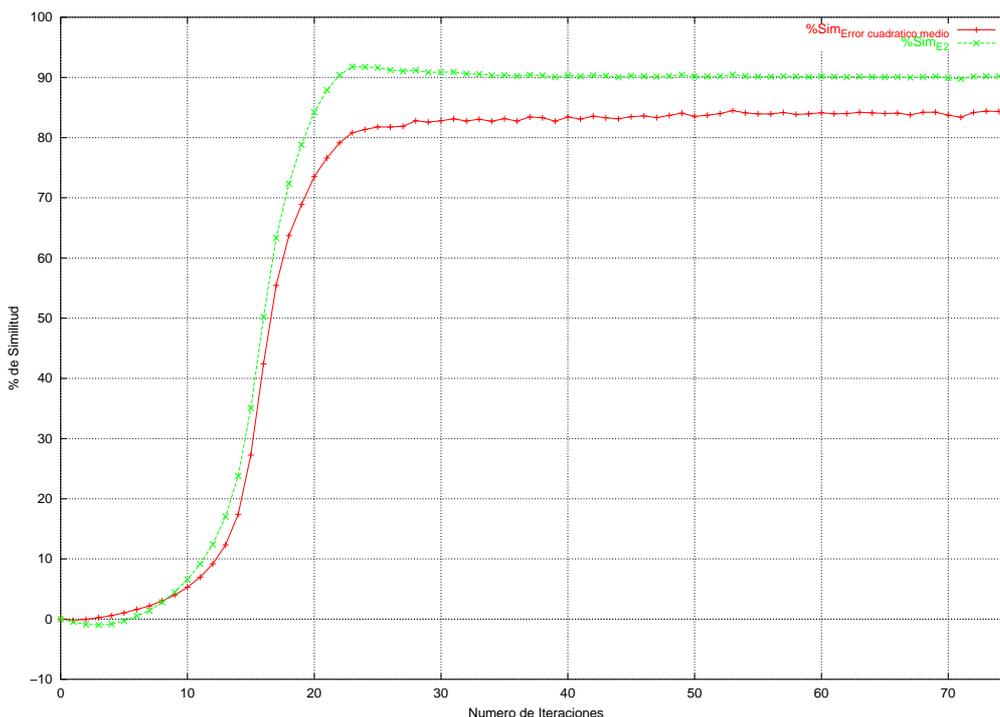


Figura 3.11: Gráfica derivada de la tabla 3.11.

el conjunto de proyecciones necesarias para poder reconstruir un modelo, por lo tanto ha este se le aplicaron solamente tres iteraciones, empezando con el modelo de la esfera, y los dos primeros modelos intermedios debido a que son los menos similares al modelo original, a continuación mostramos el archivo de parámetros con que se construyo el modelo.

```
#####Esp0_64.dat#####
# Phantom "Esp0_64.spi" description file, (generated with phantom)
# General Volume Parameters:
#      Xdim      Ydim      Zdim      Density
#      64        64        64        0.000000e+00
# Feature Parameters:
#Fea_N  Sph_Cyl Or_And fDensity      iX_Cent iY_Cent iZ_Cent iX_Rad iY_Rad iZ_Rad iAlpha iBeta iGamma
0000    e      o      1.000000e+00    11.06  -3.50   0.00   40.00  10.00  10.00 0.000 0.000 2.09439
0001    e      o      1.000000e+00   -17.05  11.00   0.00   30.00  15.00  10.00 0.000 0.000 0.52359
```

Con la gráfica 3.13 y tabla de salida 3.19 con los niveles de deformación.

En donde se puede observar que el nivel máximo de similitud es del 80% por lo que cada incremento sera de 16, escogiendo los modelos que tengan los porcentajes más aproximados, donde para este caso serán los modelos con las iteraciones:9 y 13.

Ahora veremos los resultados de cada iteración, para cada modelo se obtuvieron 1297 proyecciones indicando los resultados de manera idéntica a las sección 3.4.1.

iteración	s_{abs}
0	0.00
1	0.86
2	1.63
3	3.07
4	4.63
5	6.41
6	8.32
7	10.62
8	13.73
9	17.27
10	21.24
11	25.48
12	29.38
13	34.49
14	38.53
15	42.34
16	45.46
17	49.86
18	55.22
19	62.17
20	68.93
21	73.14
22	76.52
23	78.61
24	79.67
25	80.76
26	80.72
27	80.34
28	79.85
29	79.81
30	79.99
31	80.47
32	80.12
33	79.97
34	80.01
35	79.56
36	80.16
37	79.67
38	79.58
39	79.64

iteración	s_{abs}
40	79.77
41	79.26
42	79.56
43	79.71
44	80.04
45	79.95
46	79.95
47	79.98
48	79.57
49	79.64
50	79.60
51	79.46
52	80.00
53	79.86
54	79.59
55	79.42
56	79.66
57	79.69
58	79.68
59	79.84
60	79.54
61	79.61
62	79.64
63	79.50
64	79.57
65	79.44
66	79.76
67	79.64
68	80.10
69	79.79
70	79.50
71	79.55
72	79.64
73	79.65
74	79.49

Cuadro 3.19: Porcentaje de similitud entre el modelo inicial y el final del modelo sin simetría.

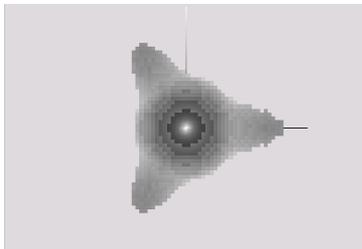
3.4. PRUEBAS CON EL ALGORITMO ITERATIVO DE RECONSTRUCCIÓN 3D 69

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	176	13.57	129	9.95	229	17.66
10	241	18.58	83	6.40	178	13.72
15	228	17.58	130	10.02	382	29.45
20	129	9.95	97	7.48	192	14.80
25	161	12.41	181	13.96	193	14.88
30	73	5.63	153	11.80	59	4.55
35	78	6.01	171	13.18	46	3.55
40	32	2.47	114	8.79	10	0.77
45	69	5.32	124	9.56	3	0.23
50	46	3.55	37	2.85	3	0.23
55	27	2.08	41	3.16	2	0.15
60	11	0.85	16	1.23	-	-
65	15	1.16	12	0.93	-	-
70	4	0.31	4	0.31	-	-
75	4	0.31	1	0.08	-	-
80	0	0.00	0	0.00	-	-
85	1	0.08	2	0.15	-	-
90	2	0.15	2	0.15	-	-
95	0	0.00	0	0.00	-	-

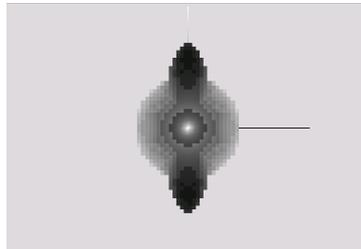
Cuadro 3.20: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 1297 en el modelo inicial esfera.

Iteración 1			Iteración 2		Iteración 3	
e	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	597	46.03	1174	90.52	1217	93.83
10	346	26.68	86	6.63	52	4.01
15	185	12.26	29	2.24	22	1.70
20	71	5.47	6	0.46	4	0.31
25	52	4.01	2	0.15	1	0.08
30	21	1.62	0	0.00	0	0.00
35	16	1.23	-	-	1	0.08
40	6	0.46	-	-	0	0.00
45	2	0.15	-	-	-	-
50	1	0.08	-	-	-	-
55	0	0.00	-	-	-	-

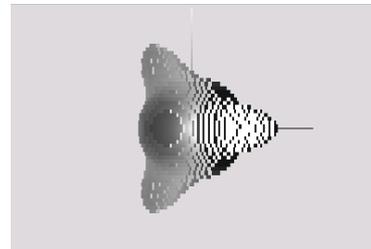
Cuadro 3.21: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 1297 en el modelo inicial 1.



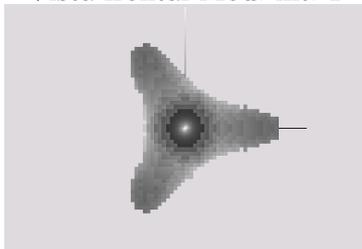
Vista frontal Mod. int. 1



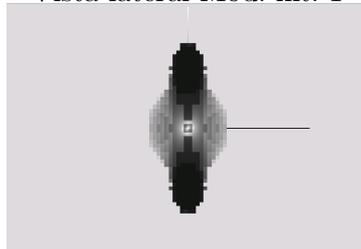
Vista lateral Mod. int. 1



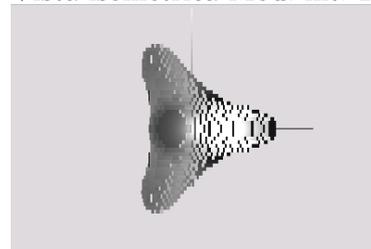
Vista isométrica Mod. int. 1



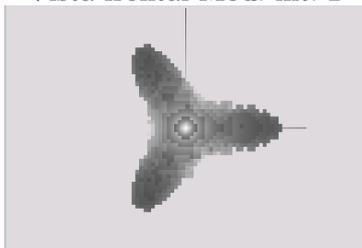
Vista frontal Mod. int. 2



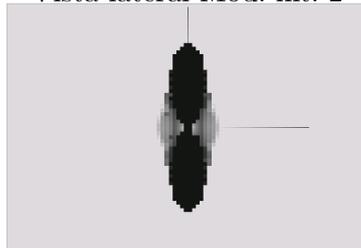
Vista lateral Mod. int. 2



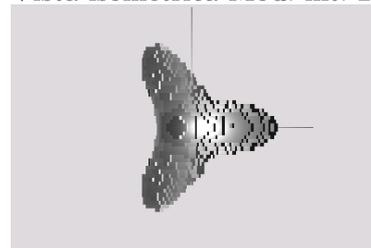
Vista isométrica Mod. int. 2



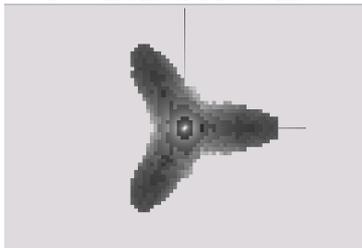
Vista frontal Mod. int. 3



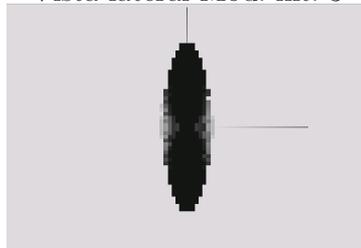
Vista lateral Mod. int. 3



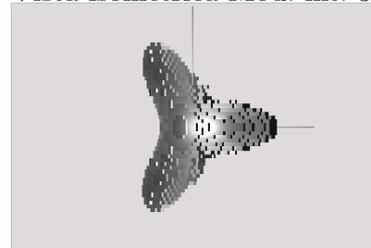
Vista isométrica Mod. int. 3



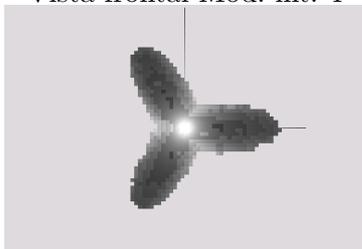
Vista frontal Mod. int. 4



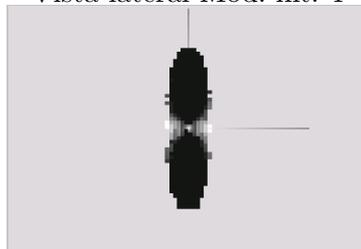
Vista lateral Mod. int. 4



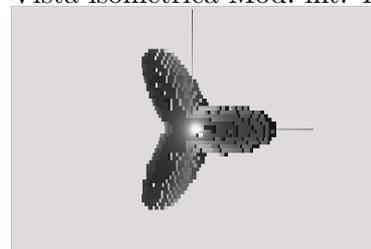
Vista isométrica Mod. int. 4



Vista frontal Mod. int. 5

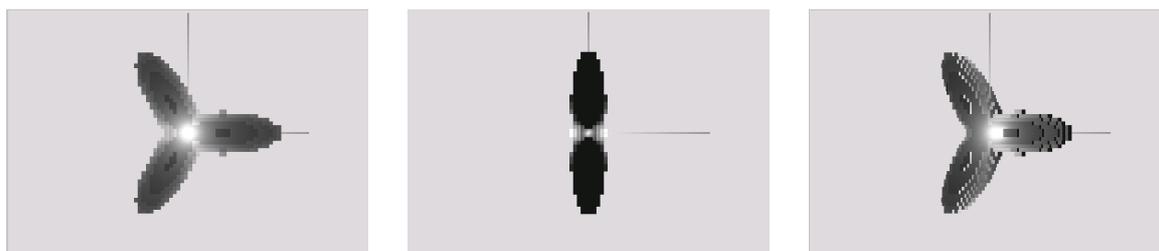


Vista lateral Mod. int. 5



Vista isométrica Mod. int. 5

3.4. PRUEBAS CON EL ALGORITMO ITERATIVO DE RECONSTRUCCIÓN 3D 71



Vista frontal Mod. original. Vista lateral Mod. original. Vista isométrica Mod. original.

Figura 3.12: Modelos intermedios obtenidos de las iteraciones al aplicar el algoritmo de reconstrucción basado en mallas de simplejos.

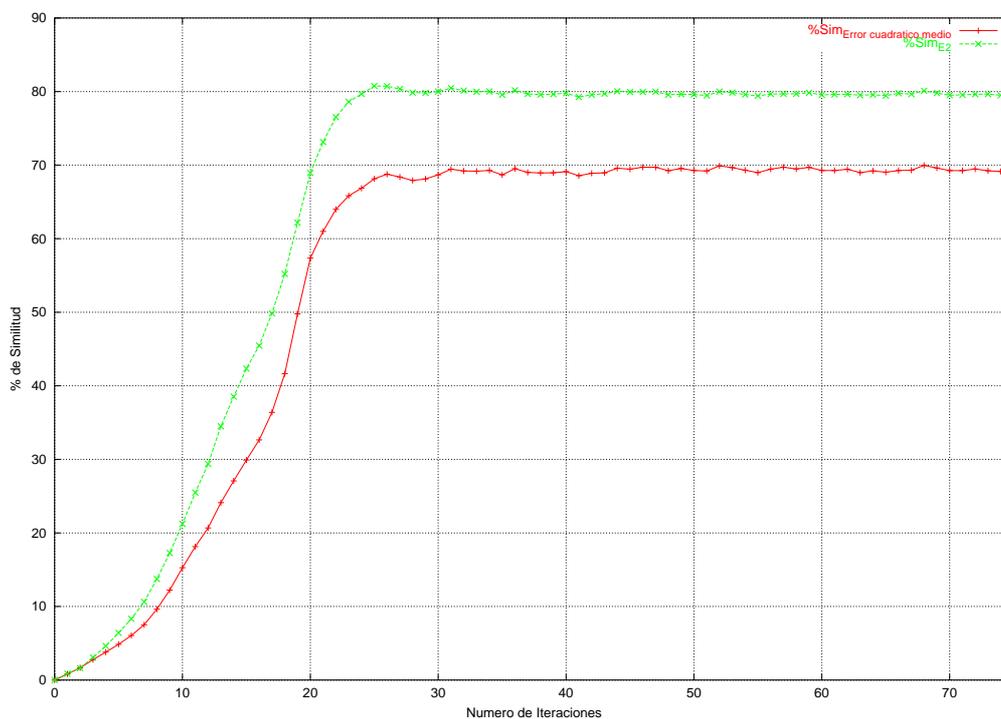


Figura 3.13: Gráfica derivada de la tabla 3.19.

e	Iteración 1		Iteración 2		Iteración 3	
	n	n/N (%)	n	n/N (%)	n	n/N (%)
5	1003	77.33	1165	89.82	1223	94.29
10	160	12.34	83	6.40	53	4.09
15	53	4.09	35	2.70	18	1.39
20	23	1.77	6	0.46	2	0.15
25	27	2.08	7	0.54	1	0.08
30	17	1.31	1	0.08	0	0.00
35	8	0.62	0	0.00	-	-
40	5	0.39	-	-	-	-
45	1	0.08	-	-	-	-
50	0	0.00	-	-	-	-

Cuadro 3.22: Tabla referente al histograma del valor absoluto de las diferencias en el valor del ángulo de inclinación de las proyecciones, el total de proyecciones es 1297 en el modelo inicial 2.

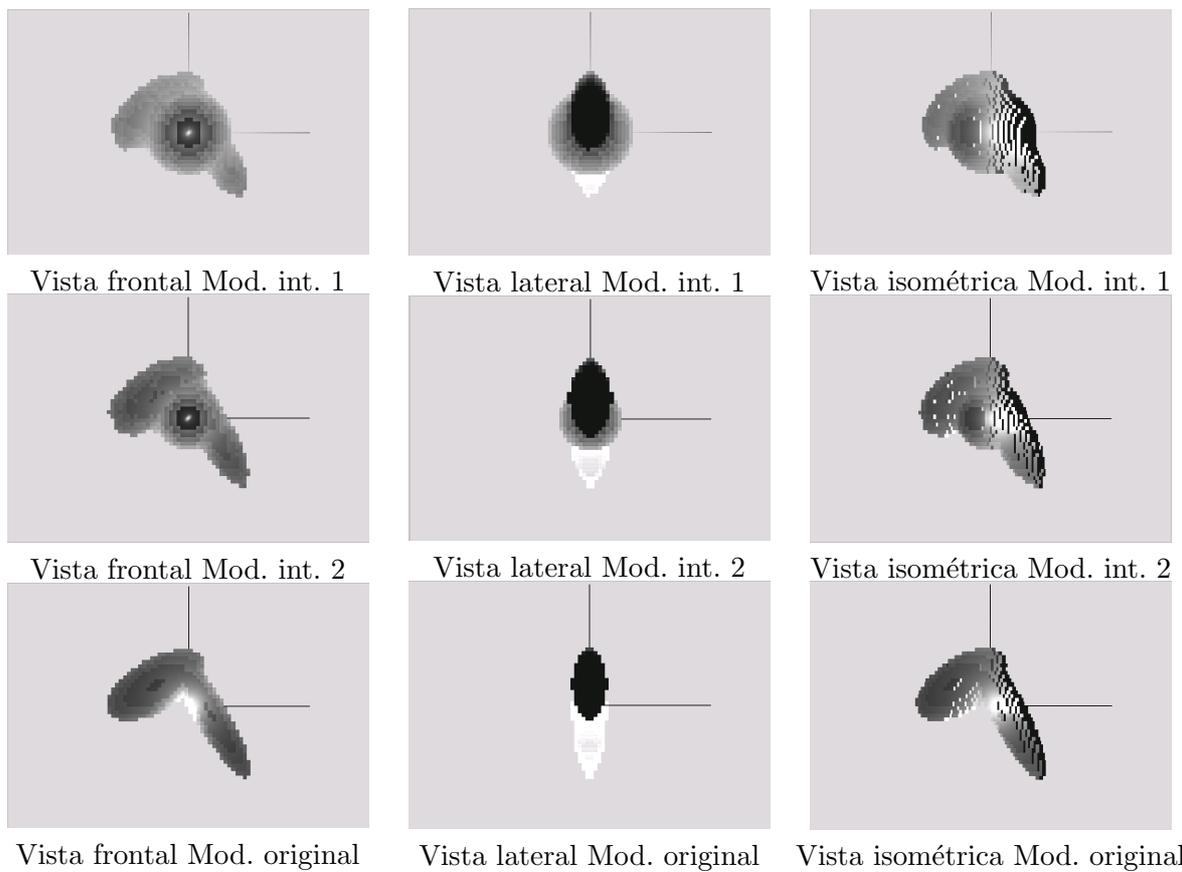


Figura 3.14: Modelos intermedios obtenidos de las iteraciones al aplicar el algoritmo de reconstrucción basado en mallas de simplejos.

Capítulo 4

Conclusiones

En este capítulo se dará las conclusiones obtenidas en el desarrollo de esta tesis, este capítulo lo dividiremos en secciones dentro de las cuales se darán nuestras conclusiones acerca de los siguientes temas:

- Mallas de simplejos.
- Pruebas al algoritmo iterativo de reconstrucción 3D.

4.1. Mallas de simplejos

Como todo método de aproximación las mallas de simplejos cuentan con ventajas y desventajas a continuación enumeraremos las más importantes.

Las mallas de simplejos son un modelo deformable discreto que cuenta con ciertas ventajas por encima de los modelo continuos. Las ventajas se centran principalmente en la cantidad de datos a procesar. En el caso de un modelo continuo se puede procesar una gran cantidad de datos, mientras que por el otro lado en lo modelos de mallas de simplejos se puede definir la resolución que maneja el modelo; entre mayor sea la resolución mejor será la aproximación entregada por el algoritmo de reconstrucción. Como se ha explicado en el capítulo 2, los modelos basados en mallas de simplejos están limitados por la posición de sus vértices vecinos, la posición del vértice atractor y la condición de continuidad utilizada que definirá que tan buena es la reconstrucción. En nuestro caso, por cuestiones de tiempo de ejecución y la baja resolución de los especímenes ($64 \times 64 \times 64$ voxels, que son los equivalentes que se obtendrían de una macromolécula biológica a baja resolución), se utilizó la *condición de continuidad orientada a superficies*, si bien es cierto que esta condición nos permite un procesamiento rápido de los datos por el número de cálculos realizados, no nos entrega la mejor reconstrucción posible debido a que en esta condición de continuidad no se contempla a los vértices asociados a los vecinos del vértice que se esta computando. Para obtener una deformación de mejor calidad es recomendable utilizar la *condición de continuidad del ángulo del simplejo*, la cual toma en cuenta la posición de los niveles de vecinos por medio del parámetro S_i . En la práctica, al estar utilizando la *condición de continuidad orientada a superficies*, sucede que en algunos vértices de la malla nos resultan algunos de los coeficientes de las coordenadas baricentricas con valor negativo, la interpretación de esta situación se traduce en que nuestro vértice se

encuentre fuera del triángulo formado por los tres vértices vecinos, cuando la intensión de la *condición de continuidad orientada a superficies* es que el vértice se mantenga siempre en el centro del triángulo asociado. Teniendo en cuenta lo mencionado anteriormente se podría pensar que este esquema esta bastante limitado en la gama de formas que puede representar, pero en realidad no sucede debido a que la deformación se basa en atractores para cada vértice, con lo que al cabo de unas cuantas iteraciones se puede lograr superficies no muy complejas (casi cualquier sólido que no tenga espirales dentro de si) con *género* igual a cero (sin hoyos que atraviesen al sólido) en menos de docientas iteraciones, según las pruebas realizadas con distintos modelos. Por otra parte, en las mallas de simplejos se depende de modelos iniciales para tomarlos como base para la modelación. La calidad de la modelación depende de que modelo inicial se utilice: si se utiliza un modelo inicial inadecuado para una modelización es muy posible que la caracterización del modelo por la malla de simplejos sea errónea. Para saber cual es el modelo inicial adecuado para cada caso, a través de la experiencia hemos recopilado las siguientes guías:

- El modelo inicial debe de ser similar al modelo final (por ejemplo si el modelo es de forma alargada utilizar un cilindro).
- Si no se tiene noción de que modelo utilizar, una buena opción es utilizar como modelo inicial la esfera.
- Los vértices del modelo inicial debe de estar lo más equiespaciados posibles.
- El modelo inicial debe de tener su centro en el mismo lugar que el centro de masa del modelo final.

En el caso del penúltimo punto mencionado, es necesario que las distancias entre los vecinos y la proyección F tengan un valor similar, debido a que la normal asociada al vértice dada por el plano formado por los vecinos a este apunte hacia el centro del modelo (como se puede ver en la figura 4.1), de lo contrario la normal puede chocar con una esquina del modelo o no chocar con él, causando un error, el cual puede propagarse hasta un número elevado de iteraciones o no desaparecer. Cuando este tipo de error sucede, la deformación progresiva del modelo inicial no es “suave”, siendo los modelos intermedios entre el inicial y el final inservibles para nuestro propósito

El último punto mencionado en nuestras recomendaciones esta encaminado a que todos los vértices del modelo discreto tengan asegurados que un punto del modelo final este en la dirección de su normal, para evitar el caso en el que el punto no tenga hacia donde ser atraído, dentro del trabajo realizado por [10] cuando un modelo es complejo se genera un modelo inicial un tanto complejo formado la unión de los modelos iniciales presentados en el capítulo 2, evitando con esto la situación mencionada, al tener el modelo inicial cierto parecido con el modelo final al menos en su topología.

4.2. Pruebas al algoritmo iterativo de reconstrucción 3D

Como se puede apreciar en el capítulo 3 sección 3.4, basta con que nuestro modelo inicial sea ligeramente parecido (aproximadamente en un 15% usando nuestra medida de similitud), para que el algoritmo converja. Entre más parecido es el modelo inicial

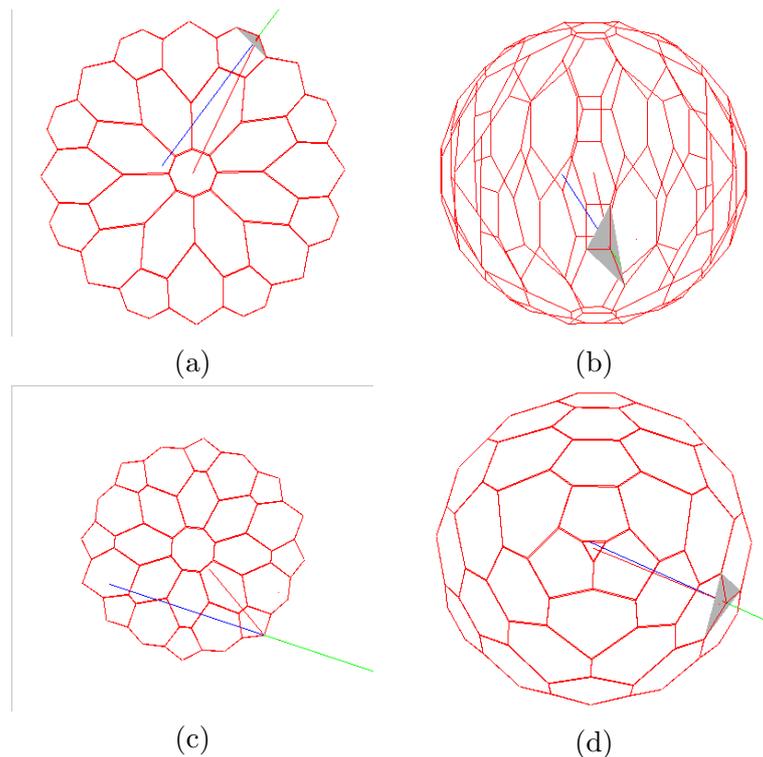


Figura 4.1: Las figuras a,b y c muestran como puede propagarse el error de una normal la cual es muy próxima al centro del modelo (a y b), al ser atraído este vértice a una parte del modelo la cual originalmente no debería provoca que la normal (en la figura c) se aleje más del centro, generando con esto en algunos casos que no tenga un punto de atracción en la dirección, positiva o negativa de su normal. En la figura d se muestra un punto con la dirección de su vector normal adecuada.

al modelo *destino*, menos iteraciones necesitará el algoritmo iterativo de reconstrucción para converger. Por la simetría de los dos primeros especímenes (con simetría 3 y 4), podría pensarse que la convergencia del algoritmo es producto de la simetría, ya que cada proyección es usada un número de veces igual a la simetría y podría “guiar” a la reconstrucción. Ahora bien, analizando los porcentajes de similitud de la primera iteración para cada modelo intermedio en cada uno de los especímenes, mostrados en la tabla 4.1, podemos ver que entre más alta es la simetría más alejado esta el resultado de la primera iteración al espécimen. Este fenómeno también está en función del número de proyecciones que existe para cada modelo, debido a que entre mayor es el número de proyecciones, su porcentaje de similitud en la primera iteración es mayor y esto queda en evidencia al revisar los porcentajes del modelo sin simetría en donde en la primera iteración del segundo modelo inicial, su porcentaje de similitud es bastante alto (77.33%). Basándonos en estos resultados se puede decir que el algoritmo de reconstrucción 3D depende de dos parámetros los cuales son el porcentaje de similitud del modelo inicial y el número de proyecciones con que se realice la reconstrucción.

Espécimen	Porcentaje de similitud					No. de Proy.
	Mod. Int. 1	Mod. Int. 2	Mod. Int. 3	Mod. Int. 4	Mod. Int. 5	
Simetría 4	7.75	23.35	26.95	53.29	74.25	334
Simetría 3	46.94	63.49	81.63	88.44	89.80	441
Sin simetría	46.03	77.33	–	–	–	1297

Cuadro 4.1: Tabla referente a la primera iteración del algoritmo de reconstrucción 3D, para cada modelo intermedio de cada espécimen.

Apéndice A

Demostraciones

En el desarrollo de este apéndice nos ocuparemos de desarrollar las demostraciones matemáticas de algunas de las formulas utilizadas en el capítulo 2, que son utilizadas para la deformación de las mallas de simplejos y que son también necesarias para su entendimiento y correcta implementación.

A.1. Ángulo del simplejo

Calcular el ángulo del simplejo es sencillo debido a la propiedad implícita de los mallas de simplejos al poder tratar su ángulo como un ángulo de geometría plana. El ángulo φ_i (como se aprecia en la figura A.1) se calcula por medio del trazo de líneas entre los puntos (P_i, P_{i-1}, P_{i+1}, C) de los cuales se obtienen dos triángulos al primero le denominaremos

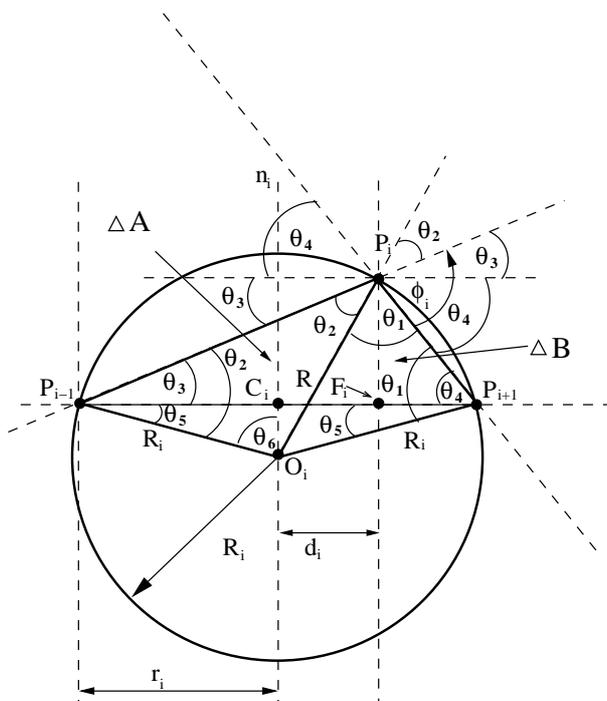


Figura A.1: El ángulo $\angle \varphi_i$ se calcula por medio de las relaciones entre los dos triángulos isósceles formados por P_i y sus vecinos.

A , formado por los vértices P_{i-1}, P_i, C , $A = \triangle(P_{i-1}, P_i, C)$; el segundo será B , definido por $B = \triangle(P_i, C, P_{i+1})$. Como se puede apreciar en la figura A.1 cada uno de los dos triángulos tiene dos aristas en donde la magnitud de estas es igual al radio R del círculo circunscrito a nuestros puntos $(P_i, P_{i-1}, P_{i+1},)$, debido a esta particularidad podemos asegurar que dos de los tres ángulos internos son iguales en ambos triángulos, con lo que partiendo de esto podemos asignar variables a los ángulos internos, como se ve en la figura A.1, y de esta forma establecer las siguientes relaciones:

$$\pi = \varphi_i + \theta_1 + \theta_2 \quad (\text{A.1})$$

$$\varphi_i = \theta_3 + \theta_4 \quad (\text{A.2})$$

$$\theta_2 = \theta_3 + \theta_5 \quad (\text{A.3})$$

$$\theta_1 = \theta_4 + \theta_5 \quad (\text{A.4})$$

$$\frac{\pi}{2} = \theta_5 + \theta_6 \quad (\text{A.5})$$

ó

$$\pi = 2\theta_5 + 2\theta_6 \quad (\text{A.6})$$

De las ecuaciones A.3 y A.4 podemos despejar θ_5 y sustituirlo en la ecuación A.6 para obtener:

$$\pi = 2\theta_6 + \theta_1 + \theta_2 - \theta_4 - \theta_3 \quad (\text{A.7})$$

al sustituir la ecuación A.2 dentro de la ecuación A.7 obtenemos:

$$\begin{aligned} \pi &= 2\theta_6 + \theta_1 + \theta_2 - \varphi_i \\ \varphi_i + \pi &= 2\theta_6 + \theta_1 + \theta_2 \\ 2\varphi_i + \pi &= 2\theta_6 + \theta_1 + \theta_2 + \varphi_i \end{aligned} \quad (\text{A.8})$$

en donde al sustituir la ecuación A.1 en A.8 obtenemos:

$$\theta_6 = \varphi_i \quad (\text{A.9})$$

y basándonos en esta igualdad podemos expresar al $\text{sen}(\varphi_i)$ y al $\text{cos}(\varphi_i)$ como:

$$\begin{aligned} \text{sen } \varphi_i &= \frac{r_i}{R_i} \text{signo}(P_i P_{N_1(i)} \cdot n_i) \\ \text{cos } \varphi_i &= \frac{\|O_i C_i\|}{R_i} \text{signo}(O_i C_i \cdot n_i) \end{aligned}$$

Donde se aprecia claramente que el ángulo φ_i depende de la posición de los vértices vecinos en relación al centro del círculo y no del lugar en el círculo en donde se encuentre P_i .

A.2. Función $L(r_i, d_i, \varphi_i)$.

Esta función es fundamental dentro de las mallas de simplejos, debido a que sin ella no se puede obtener la posición del punto P en relación de sus vecinos. Como puede verse en la figura A.2, esta función aprovecha la característica que tiene el ángulo φ_i del simplejo de poder ser interpretado como un ángulo de geometría plana para ser calculado de manera sencilla y práctica. De la figura A.2 podemos inferir las siguientes relaciones:

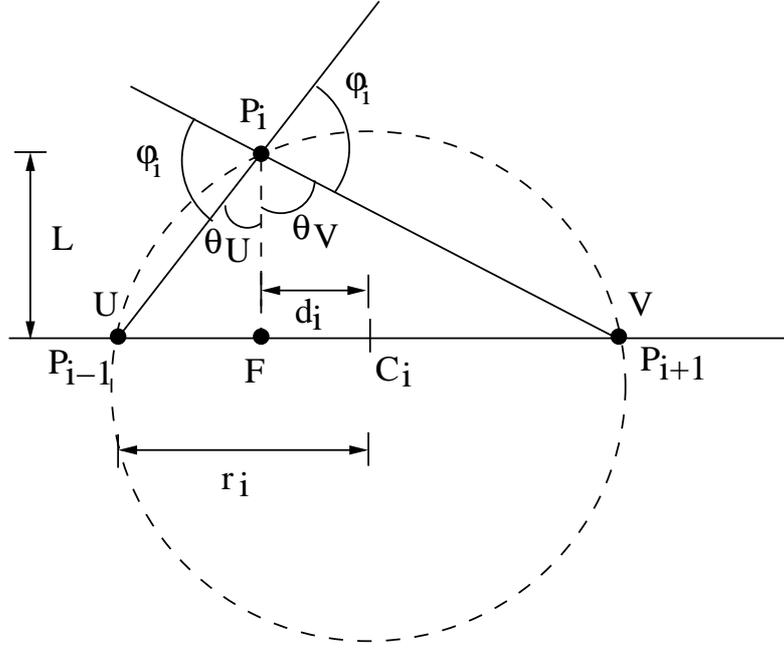


Figura A.2: La función $L(r_i, d_i, \varphi_i)$ nos devuelve la magnitud de la distancia entre la proyección del punto P_i , que es F , y el mismo punto P_i .

$$\varphi = \pi - \theta_U - \theta_V = \pi - (\theta_U + \theta_V) \quad (\text{A.10})$$

$$\tan \theta_V = \frac{r_i - d_i}{L} \quad (\text{A.11})$$

$$\tan \theta_U = \frac{r_i + d_i}{L} \quad (\text{A.12})$$

para obtener la tangente de φ , usamos la identidad para obtener la tangente de la suma de dos ángulos y consideramos que la tangente de π es cero, entonces obtenemos:

$$\begin{aligned} \tan \varphi &= \frac{\tan \pi - \tan(\theta_U + \theta_V)}{1 + \tan \pi \tan(\theta_U + \theta_V)} \\ &= -\tan(\theta_U + \theta_V) \end{aligned} \quad (\text{A.13})$$

Al sustituir las ecuaciones A.11 y A.12 en la Ec. (A.13) obtenemos:

$$\tan \varphi_i = \frac{-2 \cdot r_i \cdot L}{L^2 - (r_i^2 - d_i^2)} \quad (\text{A.14})$$

usando álgebra, la ecuación A.14 se puede expresar como una ecuación de segundo grado en función de la distancia L

$$L^2 \tan \varphi_i + 2L \cdot r_i - (r_i^2 - d_i^2) \tan \varphi_i = 0 \quad (\text{A.15})$$

al resolver esta ecuación por medio de la ecuación general de segundo grado, obtenemos la expresión para calcular la magnitud de la longitud L :

$$L = \frac{-r_i \pm \sqrt{r_i^2 + (r_i^2 - d_i^2) \tan^2 \varphi_i}}{\tan \varphi_i} \quad (\text{A.16})$$

La cual es correcta pero no adecuada para un cálculo computacional, debido a que al tener un valor de φ igual a $\frac{\pi}{2}$ se obtiene una cifra inexistente o cercana al infinito, por lo tanto esta expresión deberá transformarse en una que no contenga solamente a la tangente en el denominador. Esto se logra al multiplicar el conjugado del numerador de la ecuación A.16 por esta de como se presenta a continuación:

$$L = \frac{-r_i \pm \sqrt{r_i^2 + (r_i^2 - d_i^2) \tan^2 \varphi_i}}{\tan \varphi_i} \cdot \frac{r_i \pm \sqrt{r_i^2 + (r_i^2 - d_i^2) \tan^2 \varphi_i}}{r_i \pm \sqrt{r_i^2 + (r_i^2 - d_i^2) \tan^2 \varphi_i}} \quad (\text{A.17})$$

al desarrollar esta expresión finalmente llegaremos a la expresión utilizada para calcular la distancia L que es:

$$L = \frac{(r_i^2 - d_i^2) \tan \varphi}{r_i \pm \sqrt{r_i^2 + (r_i^2 - d_i^2) \tan^2 \varphi_i}} \quad (\text{A.18})$$

ó

$$L = \frac{(r_i^2 - d_i^2) \tan \varphi}{r_i + \epsilon \sqrt{r_i^2 + (r_i^2 - d_i^2) \tan^2 \varphi_i}} \quad (\text{A.19})$$

en donde :

$$\epsilon = 1 \quad \text{si} \quad \|\varphi_i\| < \frac{\pi}{2}$$

y

$$\epsilon = -1 \quad \text{si} \quad \|\varphi_i\| > \frac{\pi}{2}$$

A.3. Coordenadas baricéntricas

Las coordenadas baricentricas son una herramienta útil dentro de las mallas de simplejos debido a que nos permiten definir la posición de un punto respecto a otros. El primer caso a tratar es el utilizados por las mallas-1 de simplejos en donde un punto se tiene que definir a partir de sus dos vecinos.

A.3.1. Posición de un punto a partir de dos vecinos

Supongamos que se tienen tres puntos (a, b, c) los cuales están sobre el eje de x (véase la figura A.3), de la cual podemos establecer las siguientes relaciones:

$$c = b + d_2 \quad (\text{A.20})$$

$$d = d_1 + d_2 \quad \therefore \quad 1 = \frac{d_1 + d_2}{d} \quad (\text{A.21})$$

$$c - a = d_1 + d_2 \quad \therefore \quad c - a = d \quad (\text{A.22})$$

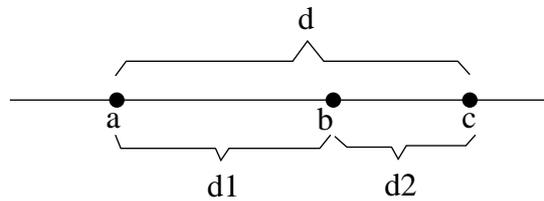


Figura A.3: La definición del punto b a partir de los puntos a y c .

De las cuales podemos obtener la siguiente ecuación al despejar c de la ecuación A.20:

$$1 = \frac{b + d_2}{c} \quad (\text{A.23})$$

en donde al igualar la ecuación A.23 con la ecuación A.21 obtenemos:

$$\frac{c \cdot d_1}{d} + \frac{c \cdot d_2}{d} = b + d_2 \quad (\text{A.24})$$

y si sustituimos la ecuación A.22 dentro de la ecuación A.24 tendremos:

$$\frac{c \cdot d_1}{d} + \frac{c \cdot d_2}{d} = b + c - d_1 - a \quad (\text{A.25})$$

reduciendo:

$$\frac{c \cdot d_1}{d} + \frac{c \cdot d_2}{d} = b + \frac{c \cdot d_1}{d} + \frac{c \cdot d_2}{d} - d_1 - \frac{a \cdot d_1}{d} - \frac{a \cdot d_2}{d} \quad (\text{A.26})$$

y finalmente:

$$\frac{a \cdot d_2}{d} + \frac{c \cdot d_1}{d} = b \quad (\text{A.27})$$

Si hacemos los siguientes cambios de variables

$$\frac{d_2}{d} = \epsilon_1 \quad y \quad \frac{d_1}{d} = \epsilon_2$$

obtendremos:

$$a\epsilon_1 + c\epsilon_2 = b \quad (\text{A.28})$$

con:

$$\epsilon_1 + \epsilon_2 = 1$$

A.3.2. Posición de un punto a partir de tres vecinos

El obtener la posición de un punto a partir de 3 puntos vecinos (o dentro de un triángulo) se apoya de la ecuación A.28 demostrada, en donde como se puede ver en la imagen A.4, se pueden trazar varias líneas rectas las cuales parten de uno de los puntos a, b, c pasan por el punto e y terminan en alguna parte de las rectas formadas por los vértices de nuestro triángulo. Debido a que estas rectas pueden ser expresadas en términos de la ecuación

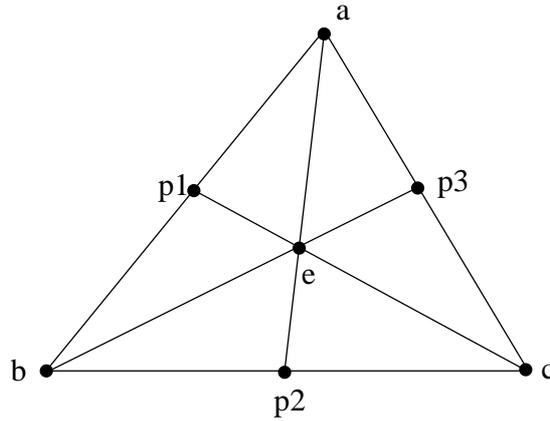


Figura A.4: La definición del punto e a partir de los puntos a, b y c .

A.28 escogeremos las rectas $\overline{cp_1}$ y \overline{ba} , para plantear la demostración a este problema, en la recta $\overline{cp_1}$ el punto que está entre p_1 y c es el punto cuya posición se desea encontrar e , expresando este en coordenadas baricéntricas se obtendría:

$$e = p_1\epsilon_{21} + c\epsilon_{22} \quad (\text{A.29})$$

con:

$$\epsilon_{21} + \epsilon_{22} = 1$$

En la recta \overline{ba} el punto p_1 se encuentra entre los puntos b y a , expresado en coordenadas baricéntricas como:

$$p_1 = a\epsilon_{11} + b\epsilon_{12} \quad (\text{A.30})$$

con:

$$\epsilon_{11} + \epsilon_{12} = 1$$

Si sustituimos la ecuación A.30 dentro de la ecuación A.29, la ecuación resultante será:

$$e = \epsilon_{21}(a\epsilon_{11} + b\epsilon_{12}) + c\epsilon_{22} = a\epsilon_{11}\epsilon_{21} + b\epsilon_{12}\epsilon_{21} + c\epsilon_{22} \quad (\text{A.31})$$

en donde podemos hacer:

$$\epsilon_{11}\epsilon_{21} + \epsilon_{12}\epsilon_{21} + \epsilon_{22} = \Gamma$$

ó

$$\epsilon_{21}(\epsilon_{11} + \epsilon_{12}) + \epsilon_{22} = \Gamma$$

pero como:

$$\epsilon_{21} + \epsilon_{22} = 1$$

tendremos

$$\begin{aligned}\epsilon_{21}(\epsilon_{11} + \epsilon_{12}) + \epsilon_{22} &= \Gamma = \epsilon_{21} + \epsilon_{22} = 1 \\ \therefore \Gamma &= 1\end{aligned}$$

y si hacemos los siguientes cambios de variables:

$$\begin{aligned}\epsilon_{11}\epsilon_{21} &= \Psi_1 \\ \epsilon_{12}\epsilon_{21} &= \Psi_2 \\ \epsilon_{22} &= \Psi_3\end{aligned}$$

con:

$$\Psi_1 + \Psi_2 + \Psi_3 = 1$$

la ecuación A.31 tomara la siguiente forma:

$$e = a\Psi_1 + b\Psi_2 + c\Psi_3 \tag{A.32}$$

A.4. Determinar el círculo que pasa a través de tres puntos en el espacio

El problema del círculo que pasa a través de tres puntos en geometría analítica es bien conocido, este se resuelve construyendo las ecuaciones de las mediatrices que pasan a través de dos lados del triángulo, el cual es formado a su vez por los tres puntos. Las mediatrices son rectas perpendiculares las cuales pasan por el punto medio del segmento de recta del cual se obtienen.

El centro del círculo es el punto en donde las mediatrices se intersectan, y el radio se obtiene fácilmente de la distancia del centro a uno de los puntos. En el caso de estar en el espacio, se puede ver que un círculo es la intersección de una esfera con un plano, el cual esta definido a su vez por los tres puntos los cuales también forman parte de el círculo, para formar las mediatrices basta con crear dos planos perpendiculares al plano que forman los tres puntos, estos planos tiene que pasar por punto medio de los segmentos de recta que son formados por los puntos, en el punto en el cual se intersecten estos planos perpendiculares y el plano, sera el centro, el radio del círculo se obtiene de manera análoga al procedimiento en geometría plana, el cual es calculando la distancia del centro a cualquiera de los puntos.

Escribiéndolo de una manera más formal, basándonos en la Fig. A.5 tenemos tres puntos en el espacio p_1, p_2, p_3 . Para obtener la ecuación del plano P de la forma

$$A_P x + B_P y + C_P z + D_P = 0 \tag{A.33}$$

que pasa por los puntos p_1, p_2, p_3 , necesitamos saber cual es el valor de los coeficientes A_P, B_P, C_P y D_P .

Los coeficientes A_P, B_P, C_P y D_P , se pueden obtener calculando los números directores de una recta perpendicular al plano, por medio del artificio de los números directores (véase [15] Art. 113), aplicado a los números directores de dos rectas las cuales sean coplanares. El coeficiente D_P se obtiene al sustituir alguno de los puntos que esta dentro del plano.

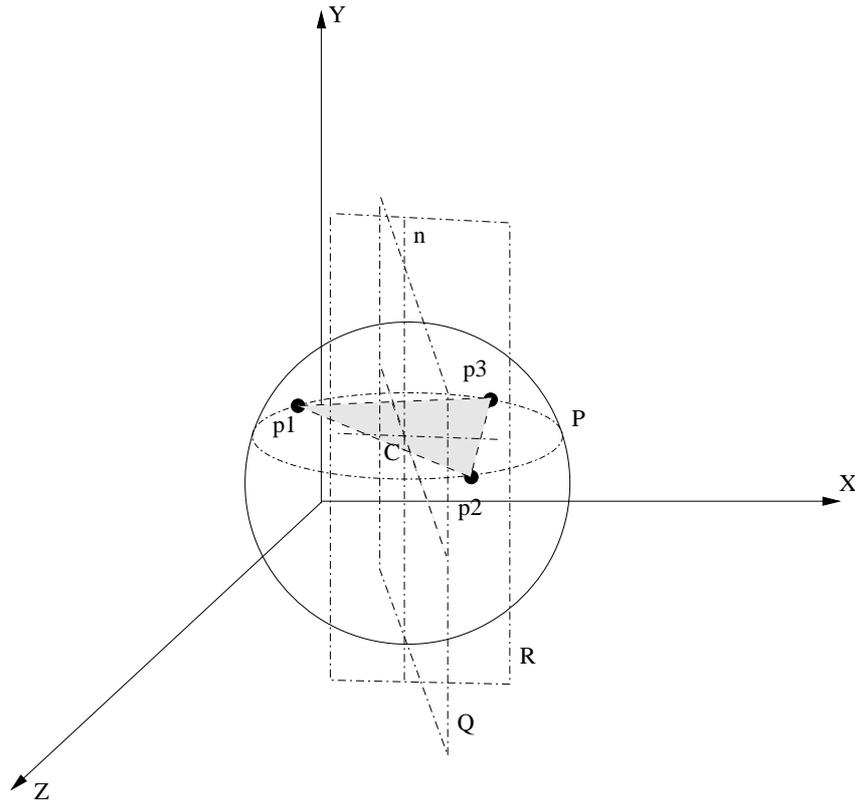


Figura A.5: Los planos formados por los tres puntos p_1 , p_2 y p_3 y su círculo circunscrito con centro C

Para obtener los planos Q y R los cuales son la mediatrices del triángulo formado por p_1, p_2 y p_3 se pueden utilizar las rectas formadas por los segmentos $\overline{p_1p_3}$ y $\overline{p_3p_2}$, tomándolas como rectas perpendiculares a los planos Q y R , respectivamente. Con lo cual la ecuación del plano Q será:

$$A_Q x + B_Q y + C_Q z + D_Q = 0 \quad (\text{A.34})$$

donde por el teorema de los números directores tendremos:

$$A_Q = p_3 x - p_1 x$$

$$B_Q = p_3 y - p_1 y$$

$$C_Q = p_3 z - p_1 z$$

El coeficiente D_Q se puede obtener al sustituir un punto conocido que este dentro del plano Q , un punto que cumple con estas características, es el punto por donde pasa el plano Q , que esta en medio del segmento de recta $\overline{p_1p_3}$, por lo cual el coeficiente D_Q sera definido por la siguiente ecuación:

$$D_Q = -\frac{A_Q(p_3 x - p_1 x) + B_Q(p_3 y - p_1 y) + C_Q(p_3 z - p_1 z)}{2} \quad (\text{A.35})$$

de manera análoga la ecuación del plano R sera de la siguiente forma:

$$A_R x + B_R y + C_R z + D_R = 0 \quad (\text{A.36})$$

donde por el teorema de los números directores tendremos:

$$A_R = p_2x - p_3x$$

$$B_R = p_2y - p_3y$$

$$C_R = p_2z - p_3z$$

con D_R definido por:

$$D_R = -\frac{A_R(p_2x - p_3x) + B_R(p_2y - p_3y) + C_R(p_2z - p_3z)}{2} \quad (\text{A.37})$$

Con todos estos datos podemos hacer el sistema de 3 ecuaciones:

$$A_Px + B_Py + C_Pz + D_P = 0$$

$$A_Qx + B_Qy + C_Qz + D_Q = 0$$

$$A_Rx + B_Ry + C_Rz + D_R = 0$$

El cual al resolverse nos arrojará por resultado el punto en donde se intersectan todos estos planos, este será nuestro centro C de el círculo, una vez que ya se tiene calculado el centro C , podemos obtener el radio r del círculo por medio de la siguiente ecuación:

$$r = \sqrt{(C_x - p_1x)^2 + (C_y - p_1y)^2 + (C_z - p_1z)^2} \quad (\text{A.38})$$

A.5. Determinar la proyección de un punto dentro de un plano.

Este problema se puede resolver de manera similar al problema anterior, como se puede apreciar en la Fig. A.6, la intersección de los planos R y Q forman la recta n , la cual es perpendicular al plano P , la ecuación de esta recta es:

$$A_Qx + B_Qy + C_Qz + D_Q + k(A_Rx + B_Ry + C_Rz + D_R) = 0 \quad (\text{A.39})$$

siendo este el resultado de resolver un sistema de ecuaciones simultáneas de las ecuaciones del plano Q y R , en donde todos los puntos que satisfagan la ecuación A.39 serán los puntos de esta recta. Esta recta puede tener un número infinito de rectas paralelas, de manera tal que existirá una recta n' la cual contenga el punto pi , los planos que se cortan para formar esta recta tienen los mismos números directores que los planos, que pasan por las mediatrices, del triángulo formado por los vértices p_1 , p_2 y p_3 . Estos dos planos se intersectan en el punto pi , de manera, tal que podemos definir las ecuaciones de los planos de la siguiente manera: Para el plano Q tendremos:

$$A_Qx + B_Qy + C_Qz + D_Q = 0 \quad (\text{A.40})$$

donde por el teorema de los números directores tendremos:

$$A_Q = p_3x - p_1x$$

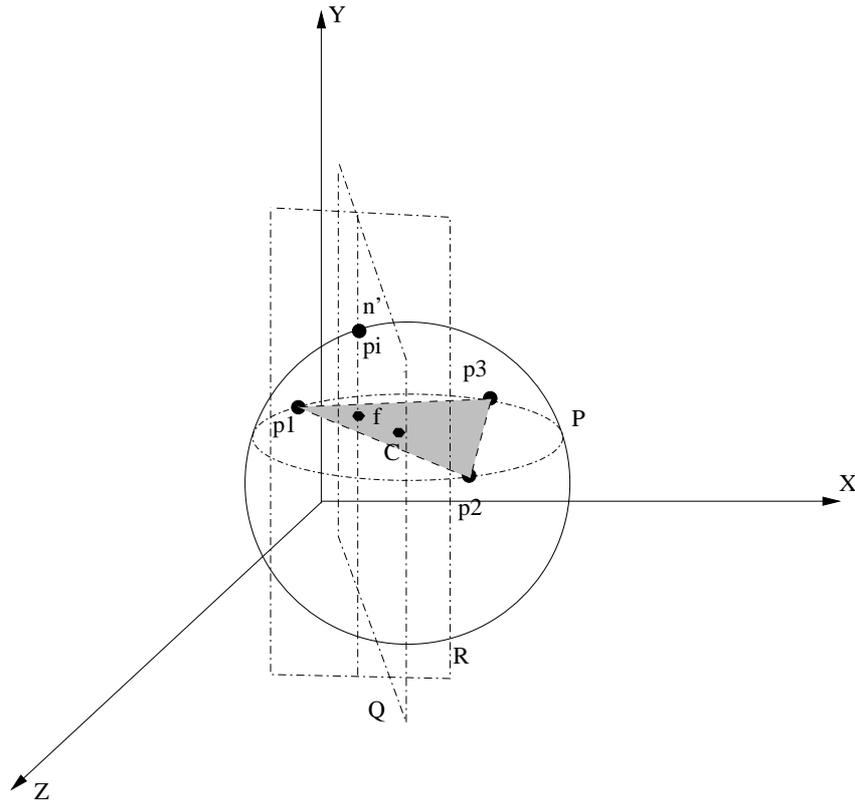


Figura A.6: La proyección f del punto p_i en el plano formados por los 3 puntos p_1, p_2, p_3 .

$$B_Q = p_3y - p_1y$$

$$C_Q = p_3z - p_1z$$

con el coeficiente D_Q definido por la siguiente ecuación:

$$D_Q = -(A_Q p_i x + B_Q p_i y + C_Q p_i z) \quad (\text{A.41})$$

de manera análoga la ecuación del plano R sera de la siguiente forma:

$$A_R x + B_R y + C_R z + D_R = 0 \quad (\text{A.42})$$

donde por el teorema de los números directores tendremos:

$$A_R = p_2x - p_3x$$

$$B_R = p_2y - p_3y$$

$$C_R = p_2z - p_3z$$

con D_R definido por:

$$D_R = -(A_R p_i x + B_R p_i y + C_R p_i z) \quad (\text{A.43})$$

Una vez que se ha definido las ecuaciones de las recta n' , la proyección f de esta recta con el plano P , sera el punto en donde la recta n' y el plano P , se intersecten, este punto F sera definido por el sistema de ecuaciones:

$$A_P x + B_P y + C_P z + D_P = 0$$

$$A_Qx + B_Qy + C_Qz + D_Q = 0$$

$$A_Rx + B_Ry + C_Rz + D_R = 0$$

En donde la diferencia entre el este problema y el anterior, radica principalmente en que los planos P y Q en este caso están trasladados, de manera tal que se intersecten en el punto pi , formando con esto una recta, la cual al cortar el plano P , nos dará su proyección, que es el punto f

A.6. Inversión

La inversión¹ es el proceso de transformar un conjunto de puntos P a un correspondiente conjunto de puntos P' conocidos como sus puntos inversos. Dos puntos P y P' se dice que son inversos con respecto a un a un círculo de inversión con un centro de inversión $O = (x_0, y_0)$ y a un radio de inversión k si P' es el pie de la altitud del $\triangle OQP$, donde Q es un punto en el círculo tal que $OQ \perp PQ$. Si P y P' son puntos inversos (como puede

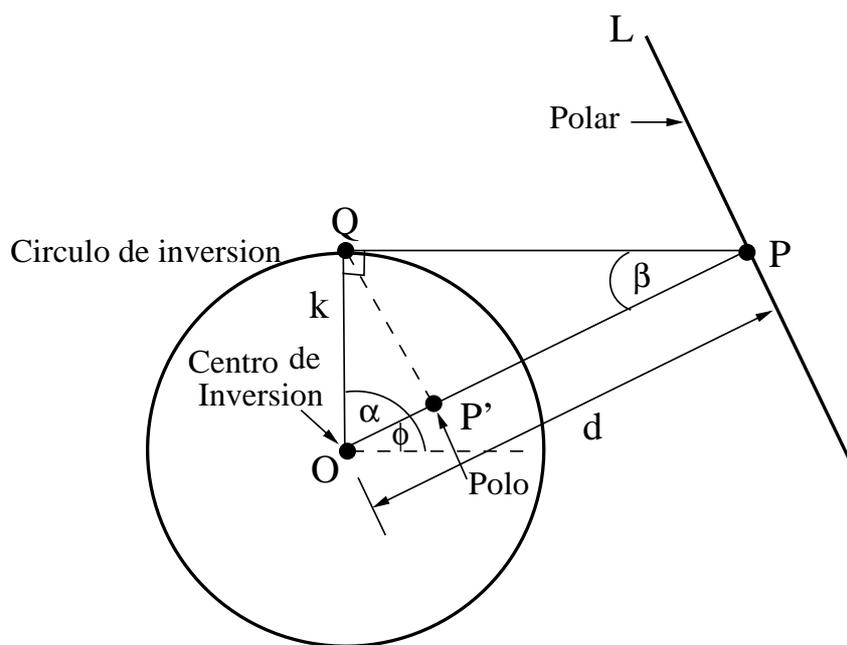


Figura A.7: La coordenada P , su círculo de inversión y su inversa P'

apreciarse en la figura A.7), entonces la línea L que pasa a través de P y perpendicular a OP es a veces llamado *polar* con respecto al punto P , conocido como el polo de inversión. Además la curva a la que es transformada bajo la inversión es llamada su *curva inversa* (o de manera más simple inversa). Este tipo de inversión fue estudiada por Jakob Steiner. Por medio de triángulos similares, se puede apreciar que los puntos inversos P' a P están dados por la siguiente relación:

$$\frac{\overline{OP}}{k} = \frac{k}{\overline{OP'}} \tag{A.44}$$

¹★:Información tomada parcialmente de:<http://mathworld.wolfram.com/Inversion.html>

ó

$$k^2 = \overline{OP} \cdot \overline{OP'} \quad (\text{A.45})$$

y si

$$\overline{OP} = \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (\text{A.46})$$

entonces:

$$\overline{OP'} = \sqrt{(x' - x_0)^2 + (y' - y_0)^2} \quad (\text{A.47})$$

también podemos escribir a OP' como:

$$\overline{OP'} = \frac{k^2}{\sqrt{(x - x_0)^2 + (y - y_0)^2}} \quad (\text{A.48})$$

Si los ángulos formados por el triángulo rectángulo $\triangle OPQ$ son: $\angle\alpha$ y $\angle\beta$, el ángulo resultante de restar $\pi - \alpha$ sera $\angle\phi$ (como se muestra en la figura A.7), de esta relación podemos expresar las coordenadas P' y P con respecto al ángulo $\angle\phi$ de la siguiente forma:

$$(x - x_0) = \cos(\beta)\overline{OP} \quad (\text{A.49})$$

y

$$(x' - x_0) = \cos(\beta)\overline{OP'} \quad (\text{A.50})$$

donde al sustituir la ecuación A.48 con la ecuación A.50 obtendremos:

$$(x' - x_0) = \frac{k^2 \cdot \cos(\beta)}{\sqrt{(x - x_0)^2 + (y - y_0)^2}} \quad (\text{A.51})$$

si despejamos $\cos(\beta)$ y sustituimos la ecuación A.46 dentro de la ecuación A.49 obtendremos la siguiente ecuación:

$$\cos(\beta) = \frac{(x - x_0)}{\sqrt{(x - x_0)^2 + (y - y_0)^2}} \quad (\text{A.52})$$

donde al sustituir la ecuación A.52 dentro de la ecuación A.51 obtendremos

$$(x' - x_0) = \frac{\frac{k^2 \cdot (x - x_0)}{\sqrt{(x - x_0)^2 + (y - y_0)^2}}}{\sqrt{(x - x_0)^2 + (y - y_0)^2}} \quad (\text{A.53})$$

En donde despejando la coordenada x_0 y con un poco de álgebra obtendremos la forma final de la coordenada x' del punto P' que es:

$$x' = x_0 + \frac{k^2(x - x_0)}{(x - x_0)^2 + (y - y_0)^2} \quad (\text{A.54})$$

Aplicando el mismo proceso a la coordenada y' en relación del ángulo $\sin(\phi)$ la coordenada y' para el punto P' sera:

$$y' = y_0 + \frac{k^2(y - y_0)}{(x - x_0)^2 + (y - y_0)^2} \quad (\text{A.55})$$

A.7. Inversión en el espacio

De manera análoga el concepto de inversión puede ser llevado al espacio tridimensional por medio de una esfera de inversión. El estar dentro del espacio tridimensional no modifica (como se ve en la figura A.8) en mucho la manera de resolver el problema, las relaciones dadas por la ecuaciones A.44 y A.45 siguen siendo válidas. Debido a que se está en el espacio tridimensional la forma de calcular los segmentos \overline{OP} y $\overline{OP'}$ será:

$$\overline{OP} = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \quad (\text{A.56})$$

$$\overline{OP'} = \sqrt{(x' - x_0)^2 + (y' - y_0)^2 + (z' - z_0)^2} \quad (\text{A.57})$$

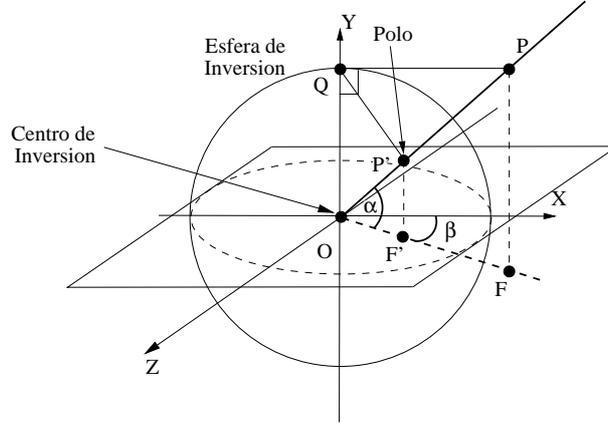


Figura A.8: La coordenada P , su esfera de inversión y su inversa P'

Por lo tanto podemos escribir $\overline{OP'}$ como:

$$\overline{OP'} = \frac{k^2}{\sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}} \quad (\text{A.58})$$

y como se puede apreciar en la figura A.8 podemos definir las coordenadas del punto P' por medio de dos ángulos α y β , en donde podemos apreciar que el $\text{sen}(\alpha)$ está definido de la siguiente manera:

$$\text{sen}(\alpha) = \frac{(y - y_0)}{\sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}} \quad (\text{A.59})$$

ó

$$\text{sen}(\alpha) = \frac{(y' - y_0)}{\overline{OP'}} = \frac{(y' - y_0)\sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}}{k^2} \quad (\text{A.60})$$

y llegamos a la siguiente expresión:

$$y' = y_0 + \frac{k^2(y - y_0)}{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \quad (\text{A.61})$$

la cual nos da la coordenada en el eje y del punto P' , las coordenadas x y z se obtienen en función del ángulo $\angle\beta$, como se muestra en la figura A.8, se obtienen mediante el uso de la proyección de los puntos P y P' , que son F y F' respectivamente de donde se pueden obtener las siguientes relaciones:

$$\cos(\beta) = \frac{z' - z_0}{\overline{OF'}} \quad (\text{A.62})$$

ó

$$\cos(\beta) = \frac{z - z_0}{\overline{OF}} \quad (\text{A.63})$$

y

$$\text{sen}(\beta) = \frac{x' - x_0}{\overline{OF'}} \quad (\text{A.64})$$

ó

$$\text{sen}(\beta) = \frac{x - x_0}{\overline{OF}} \quad (\text{A.65})$$

al igualar la ecuación A.62 con la ecuación A.63 y la ecuación A.64 con la ecuación A.65 obtendremos las siguientes dos ecuaciones:

$$\frac{z' - z_0}{\overline{OF'}} = \frac{z - z_0}{\overline{OF}} \quad (\text{A.66})$$

$$\frac{x' - x_0}{\overline{OF'}} = \frac{x - x_0}{\overline{OF}} \quad (\text{A.67})$$

las proyecciones $\overline{OF'}$ y \overline{OF} pueden ser calculadas en base al ángulo $\angle\alpha$ dándonos como resultado las siguientes relaciones:

$$\overline{OF'} = \cos(\alpha)\overline{OP'} \quad (\text{A.68})$$

y

$$\cos(\alpha) = \frac{\overline{OF}}{\overline{OP}} \quad (\text{A.69})$$

Despejando $(z' - z_0)$ y $(x' - x_0)$ de las ecuaciones A.66 y A.67 y sustituyendo el valor de $\overline{OF'}$ de la ecuación A.68 obtenemos:

$$z' - z_0 = \frac{(z - z_0) \cos(\alpha) \overline{OP'}}{\overline{OF}} \quad (\text{A.70})$$

$$x' - x_0 = \frac{(x - x_0) \cos(\alpha) \overline{OP'}}{\overline{OF}} \quad (\text{A.71})$$

de la ecuación A.57 sabemos el valor de $\overline{OP'}$ donde al sustituirlo obtenemos:

$$z' - z_0 = \frac{(z - z_0) \cos(\alpha) k^2}{\overline{OF} \cdot \overline{OP}} \quad (\text{A.72})$$

A.8. ÁNGULO DE LA RECTA RESULTANTE AL CÍRCULO INVERTIDO (DIRECCIÓN DEL VECTOR)

$$x' - x_0 = \frac{(x - x_0) \cos(\alpha) k^2}{\overline{OF} \cdot \overline{OP}} \quad (\text{A.73})$$

al sustituir la ecuación A.69 dentro de las ecuaciones A.72 y A.73 las ecuaciones resultantes son:

$$z' - z_0 = \frac{\frac{(z - z_0) k^2 \cdot \overline{OF}}{\overline{OP}}}{\overline{OF} \cdot \overline{OP}} = \frac{(z - z_0) k^2}{\overline{OP}^2} \quad (\text{A.74})$$

$$x' - x_0 = \frac{\frac{(x - x_0) k^2 \cdot \overline{OF}}{\overline{OP}}}{\overline{OF} \cdot \overline{OP}} = \frac{(x - x_0) k^2}{\overline{OP}^2} \quad (\text{A.75})$$

como el valor del segmento \overline{OP} es bien conocido de la ecuación A.56 al sustituirlo obtendremos el valor de las coordenadas x' y z' que son:

$$x' = x_0 + \frac{(x - x_0) k^2}{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \quad (\text{A.76})$$

$$z' = z_0 + \frac{(z - z_0) k^2}{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \quad (\text{A.77})$$

A.8. Ángulo de la recta resultante al círculo invertido (dirección del vector perpendicular u_i a la recta resultante)

Para obtener la dirección del vector unitario \vec{u}_i se tiene el siguiente problema (el cual podemos visualizar en la Fig.A.9): Necesitamos saber cual es el ángulo de la pendiente de la recta que Q que da por resultado de invertir el círculo P con centro en $(0, 0)$ y radio R , con respecto del círculo de inversión C con centro en p_i y radio unitario.

De la Fig. A.9, podemos deducir lo siguiente: Las ecuaciones paramétricas del punto p_i que está sobre el círculo P son:

$$\begin{aligned} x &= R \cos \phi \\ y &= R \sin \phi \end{aligned} \quad (\text{A.78})$$

Para el punto p_{i-1} , se puede decir que el valor de ϕ es 180, lo cual hace que sus ecuaciones paramétricas sean:

$$\begin{aligned} x &= -R \\ y &= 0. \end{aligned} \quad (\text{A.79})$$

De manera análoga para el punto p_{i+1} podemos decir que el valor de ϕ es 0, lo cual hace que sus ecuaciones paramétricas sean:

$$\begin{aligned} x &= R \\ y &= 0. \end{aligned} \quad (\text{A.80})$$

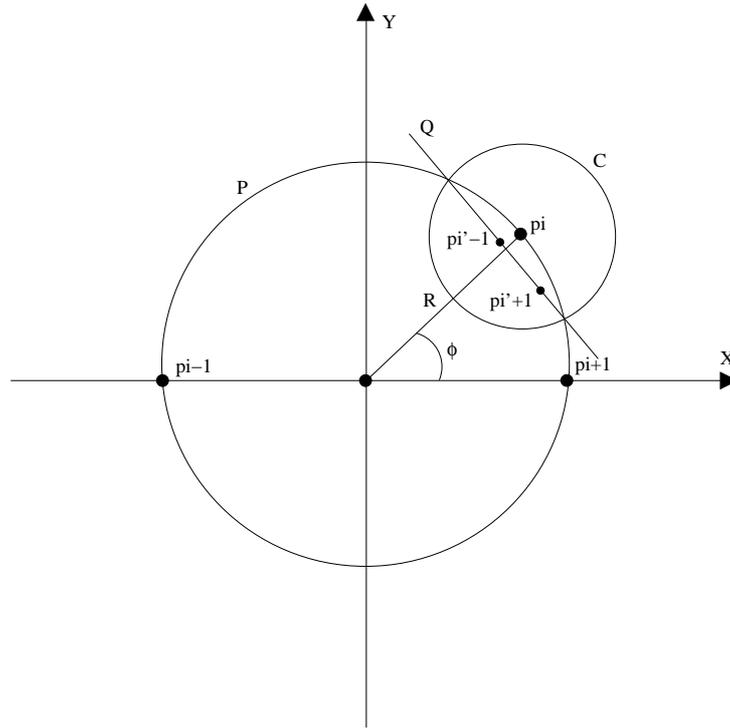


Figura A.9: Definición de nuestro problema: Queremos saber cual es el ángulo de la pendiente m de la recta Q .

Al aplicar la inversión a los puntos p_{i-1} , p_{i+1} , obtendremos los puntos p'_{i-1} y p'_{i+1} , los cuales son parte de la recta Q (que en este caso, es el resultado de la inversión del círculo P con respecto al círculo C).

Las ecuaciones del punto p'_{i-1} son:

$$p'_{i-1}x = R \cos \phi + \frac{-(1 + \cos \phi)}{2R(1 + \cos \phi)} \quad (\text{A.81})$$

$$p'_{i-1}y = R \sin \phi + \frac{-\sin \phi}{2R(1 + \cos \phi)}.$$

De manera análoga, las ecuaciones del punto p'_{i+1} son:

$$p'_{i+1}x = R \cos \phi + \frac{1 + \cos \phi}{2R(1 - \cos \phi)} \quad (\text{A.82})$$

$$p'_{i+1}y = R \sin \phi + \frac{-\sin \phi}{2R(1 - \cos \phi)}.$$

La fórmula de la recta que pasa por dos puntos nos dice que la pendiente m se calcula de la siguiente forma:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Si sustituimos las ecuaciones A.82 y A.81, tomando a p'_{i-1} como el punto 1 y a p'_{i+1} como el punto 2, entonces obtendremos:

$$\begin{aligned}
m &= \frac{\frac{-\operatorname{sen} \phi}{2R(1-\cos \phi)} - \frac{-\operatorname{sen} \phi}{2R(1+\cos \phi)}}{\frac{1+\cos \phi}{2R(1-\cos \phi)} - \frac{-\operatorname{sen} \phi}{2R(1+\cos \phi)}} \\
&= \frac{-\operatorname{sen} \phi(1 - \cos \phi) + \operatorname{sen} \phi(1 + \cos \phi)}{-2(1 + \cos \phi)(1 - \cos \phi)} \\
&= \frac{\cos \phi}{-\operatorname{sen} \phi} \\
&= \frac{\operatorname{sen}(\phi + \frac{\pi}{2})}{\cos(\phi + \frac{\pi}{2})} = \tan(\phi + \frac{\pi}{2}).
\end{aligned} \tag{A.83}$$

Con lo que podemos ver que el ángulo de la pendiente m de la recta Q es $\phi + \frac{\pi}{2}$, de lo anterior podemos deducir que si existiese un vector unitario \vec{u}_i perpendicular a la recta Q tendría la dirección del ángulo ϕ .

De lo siguiente podemos desprender también las siguientes ecuaciones:

$$\begin{aligned}
p_i &= p_0 + R\vec{u}_i \\
o \\
p_0 &= p_i - R\vec{u}_i.
\end{aligned} \tag{A.84}$$

A.9. Distancia de un punto en el círculo a la curva invertida

Supongamos que se tiene un círculo C a invertir de radio R , un círculo de inversión c_i de con radio igual a uno y centro en $(0, R)$ y una recta L que es el resultado de aplicar la inversión al círculo C con el círculo de inversión c_i (como se ve en la figura A.10). Las ecuaciones de los puntos invertidos con el radio del círculo de inversión igual a uno son:

$$\begin{aligned}
x' &= x_0 + \frac{R \cos \theta - x_0}{R^2 + x_0^2 + y_0^2 - 2R(x_0 \cos \theta + y_0 \operatorname{sen} \theta)} \\
y' &= y_0 + \frac{R \operatorname{sen} \theta - y_0}{R^2 + x_0^2 + y_0^2 - 2R(x_0 \cos \theta + y_0 \operatorname{sen} \theta)}
\end{aligned} \tag{A.85}$$

al sustituir las coordenadas $(0, R)$ de nuestro círculo de inversión c_i en las ecuaciones A.85 obtenemos:

$$\begin{aligned}
x' &= x_0 + \frac{R \cos \theta}{R^2 + R^2 - 2R(R \operatorname{sen} \theta)} \\
x' &= x_0 + \frac{R \cos \theta}{2R^2(1 - \operatorname{sen} \theta)}
\end{aligned} \tag{A.86}$$

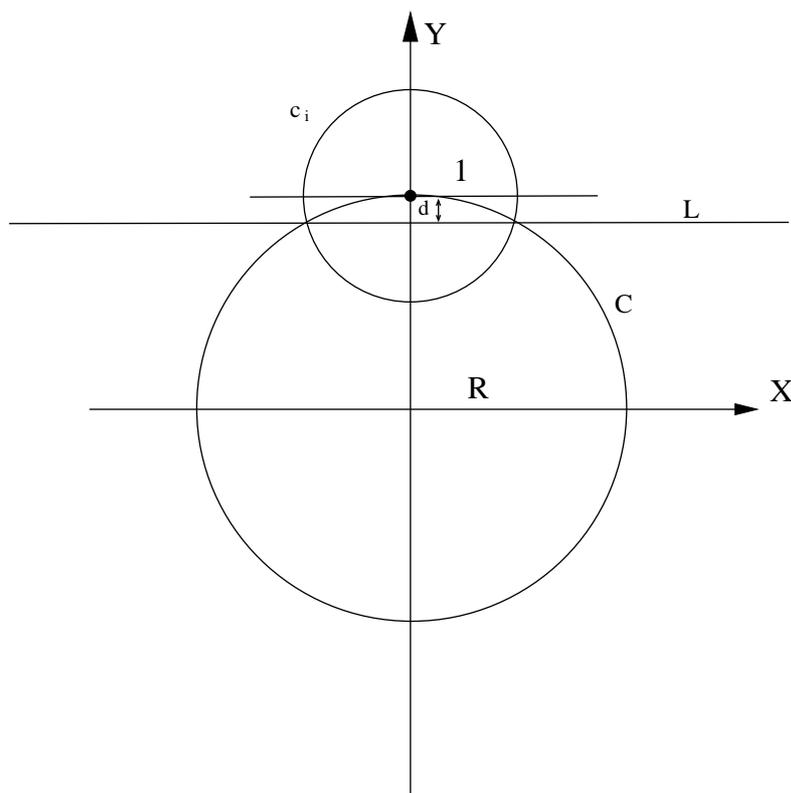


Figura A.10: El círculo de inversión c_i , el círculo a invertir C y la curva invertida L

$$\begin{aligned}
 y' &= R + \frac{R \operatorname{sen} \theta - R}{2R^2(1 - \operatorname{sen} \theta)} \\
 y' &= R + \frac{-R(1 - \operatorname{sen} \theta)}{2R(1 - \operatorname{sen} \theta)} \\
 y' &= R - \frac{1}{2R}
 \end{aligned}
 \tag{A.87}$$

de la ecuación A.87 podemos concluir que la distancia d entre la curva a invertir y la curva invertida sera:

$$d = \frac{1}{2R} \tag{A.88}$$

si y solo si el radio de c_i es igual a uno y el centro de c_i se encuentra sobre la curva C .

Apéndice B

Lista de Programas Realizados.

1. Generación de modelos iniciales.
 - Estructuras de datos.
 - Distribución de vértices y aristas para la esfera, cilindro y toroide.
 - Cálculo de diversos valores (normales, ángulos, etc).
 - Inversión de coordenadas.
 - Coordenadas baricéntricas.
2. Deformación
 - Fuerza interna
 - Normal
 - Tangente
 - Fuerza externa.
 - Trazo de rectas en el espacio tridimensional.
 - Algoritmo iterativo de relleno por inundación en tercera dimensión.
3. Modelaje tridimensional
 - Generación del modelo de una macromolécula biológica
 - Generación de modelos sucesivos desde el modelo inicial al modelo de la macromolécula
 - Interfase gráfica para la generación de modelos iniciales, intermedios y finales.

Apéndice C

Algoritmo para el Trazo de una Linea en Tres Dimensiones

A continuación vamos a dar una explicación de nuestra implementación del algoritmo de Bresenham (comúnmente conocido como algoritmo del punto medio), la cual fue expandida a tres dimensiones.

Partiendo de la explicación dada en [16](pags. 74-78) podemos expresar una línea recta por medio de la función implícita con coeficientes a , b y c . En el espacio tridimensional las ecuaciones tomadas de [15] nos indican que una línea recta en el espacio se puede expresar utilizando la forma simétrica como:

$$\frac{x - x_1}{a} = \frac{y - y_1}{b} = \frac{z - z_1}{c} \quad (\text{C.1})$$

Siendo $[a, b, c]$ los números directores de esta recta en el espacio. Si se tienen dos puntos $P_1(x_1, y_1, z_1)$ y $P_2(x_2, y_2, z_2)$ sobre los cuales pasa la recta, sus números directores están dados por $[x_2 - x_1, y_2 - y_1, z_2 - z_1]$. Por lo tanto la ecuación C.1 tomara la siguiente forma:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} \quad (\text{C.2})$$

De la ecuación C.2 se pueden deducir las siguientes ecuaciones:

$$\frac{x - x_1}{x_2 - x_1} = \frac{z - z_1}{z_2 - z_1} \quad (\text{C.3})$$

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} \quad (\text{C.4})$$

Al desarrollar la ecuación C.4 obtenemos:

$$x = \frac{z dx}{dz} - \frac{z_1 dx}{dz} + x_1 \quad (\text{C.5})$$

Donde:

$$dx = x_2 - x_1$$

y

$$dz = z_2 - z_1$$

la ecuación C.5 se puede expresar de la siguiente forma:

$$x = \frac{zdx}{dz} + B_{zx} \quad (\text{C.6})$$

Con:

$$B_{zx} = x_1 - \frac{z_1 dx}{dz}$$

con lo cual podemos expresar construir una función la siguiente función implícita:

$$F(z, x) = dx \cdot z - dz \cdot x + B_{zx} dz = 0 \quad (\text{C.7})$$

De manera análoga a la ecuación C.4 podemos transformarla a la forma implícita:

$$F(x, y) = dy \cdot x - dx \cdot y + B_{xy} dx = 0 \quad (\text{C.8})$$

La cual es idéntica a la ecuación dada en [16](pag. 76), teniendo como resultado la descomposición de una recta en tres dimensiones a dos en dos dimensiones, si tomamos como punto de partida la ecuación C.7 el algoritmo descrito en [16] se puede aplicar al plano $z - x$, como ya se ha mencionado la ecuación C.8 es idéntica a la manejada en [16] esta ecuación en específico maneja el plano $x - y$, por lo tanto para hacer el trazo de una línea en el espacio tridimensional se debe de aplicar el algoritmo de Bresenham dos veces (una por el plano $z - x$ y otra por el plano $x - y$). La forma en que se aplicara para cada plano el algoritmo no debe ser arbitraria, por ser un algoritmo incremental el algoritmo del punto medio tiene la siguiente estrategia:

- Siempre se incrementa el valor de una de las coordenadas (en el plano $x - y$, si dx es mayor que dy el incremento sera en x).
- Dependiendo del valor de la diferencia incremental d (si es mayor al valor de la pendiente) se incrementara el valor de la otra coordenada.
- Cuando la diferencia en dx sea igual a cero el algoritmo termina.

Para tres dimensiones la manera de proceder es similar, la diferencia radica en en las coordenadas de las cuales dependen los planos, para hacer la explicación mas clara tomemos el siguiente caso:

Se quiere trazar un línea en tres dimensiones en donde los valores definidos resultantes de obtener la diferencia entre cada una de las coordenadas nos da la siguiente relación (mostrada en la figura C.1): $dz > dx > dy$. En donde para trazarla se utilizaran las siguientes reglas:

- Siempre se incrementa el valor de la coordenada en el eje z .
- Se utilizaran dos variables para la diferencia incremental d_1 y d_2 (una para cada plano).
- El plano $z - x$ utilizara la variable d_1 .
- El plano $x - y$ utilizara la variable d_2 .

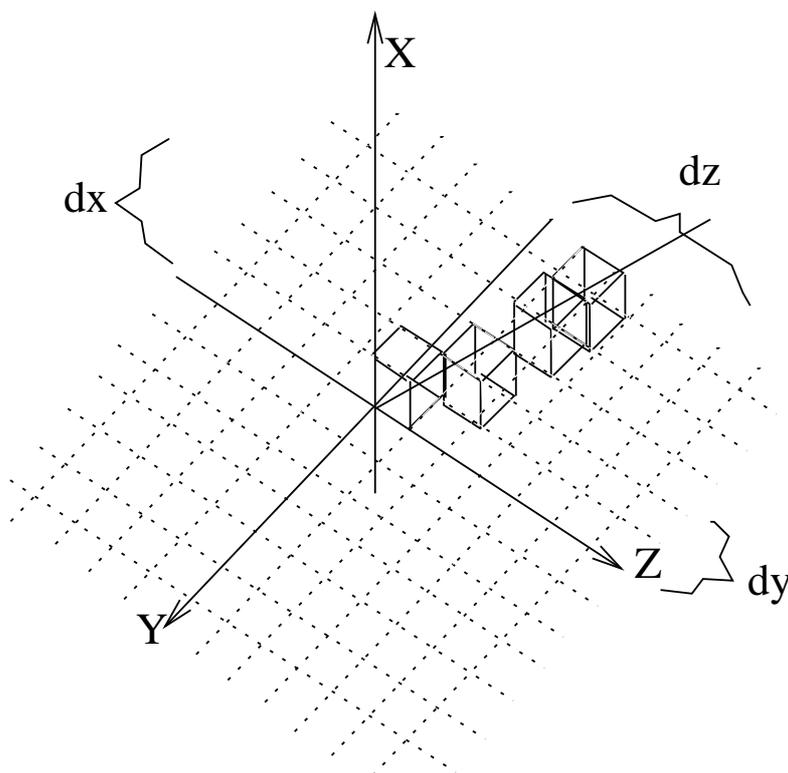


Figura C.1: La diferencias de una recta trazada en el espacio tridimensional.

- Dependiendo del valor de la diferencia incremental d_1 , se incrementara el valor en x .
- Si se ha incrementado el valor en la coordenada x , y la diferencia incremental d_2 , es mayor al valor de la normal de la recta formada por la ecuación C.8, se incrementara el valor en y .

Como se puede apreciar la expansión del algoritmo del punto medio se basa en detectar en cual de las coordenadas se tiene una diferencia mayor, una vez que se ha detectado debido la igualdad mostrada en la ecuación C.2, se podrá trazar una recta en cualquier parte del espacio haciendo una generalización de las reglas anteriores en donde básicamente se descompone a la recta por planos dependiendo de cual es el valor de las diferencias entre el punto inicial y el final, la coordenada que tenga la máxima diferencia y la que tenga el valor intermedio de las diferencias formara un plano. Una vez que se tiene el primer plano el segundo se escoge con el valor mínimo de las diferencias y el valor intermedio, con la particularidad de que la recta que se encuentra en el segundo plano se trazara dependiendo de si existió incremento en la coordenada a la que le corresponde el valor intermedio.

Apéndice D

Poliedros

Un poliedro es un sólido el cual está unido por medio de un conjunto de polígonos, en estos polígonos cada una de sus aristas es miembro de un número par de polígonos y satisfacer algunas restricciones adicionales. Un *poliedro simple* es aquel que puede ser deformado dentro de una esfera; esto significa que este poliedro, a diferencia del toroide, no tiene agujeros. La representación de borde de un poliedro simple satisface la formula de Euler, la cual expresa una relación invariante entre el número de vértices, aristas, y caras de un poliedro simple:

$$V - E + F = 2 \tag{D.1}$$

donde:

- V es el número de vértices.
- E es el número de aristas.
- F es el número de caras.

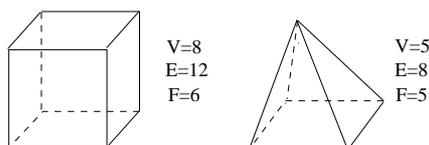


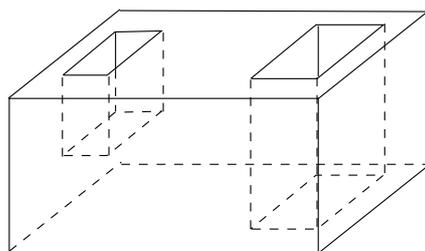
Figura D.1: Algunos poliedros simples con los valores de V , E y F .

Nótese que la formula de Euler es aún válida para cuando existen aristas curvadas y caras que su superficie no sea plana. Una generalización de la formula de Euler aplica a poliedros los cuales tienen hoyos en sus caras:

$$V - E + F - H = 2(C - G) \tag{D.2}$$

- H es el número de hoyos en las caras.
- G es el número de hoyos que pasan a través del objeto.
- C es el número de componentes separados (partes) del objeto.

Si un objeto tiene un solo componente, la variable G es conocida como *género*; si este tiene múltiples componentes, entonces G es la suma de los género de sus componentes.



$$\begin{array}{r} V - E + F - H = 2(C - G) \\ 24 - 36 + 15 - 3 = 2(1 - 1) \end{array}$$

Figura D.2: Un poliedro con hoyos donde se puede ver las características de la generalización de la formula de Euler.

Bibliografía

- [1] L.G. de la Fraga. *Refinamiento de la Estructura Tridimensional de Macromoléculas Biológicas Aisladas: Principios y Realización en Multiprocesadores*. PhD thesis, Centro Nacional de Biotecnología., Campus de Cantoblanco. 28049, Madrid, España., 1998.
- [2] Timothy S. Baker and R. Holland Cheng. A model-based approach for determining orientations of biological . macromolecules imaged by cryoelectron microscopy. *Journal of Structural Biology*, 116, 1996.
- [3] P. A. Penczek, R. A. Grassuni, and J. Frank. The ribosome at improved resolution: new techniques for for merging and orientation refinement in 3D cryo-electron microscopy of biological particles. *Ultramicroscopy*, 53:251–270, 1994.
- [4] M. van Heel. In A. Csanády, P. Röhlich, and D. Szabó, editors, *Proc. 8th Eur. Congr. on Electron Microscopy, Budapest*, volume 2, pages 1347–1348, 1984.
- [5] M. Radermacher. Three-dimensional reconstruction from random projections: Orientational alignment via Radon transforms. *Ultramicroscopy*, 53:121–136, 1994.
- [6] J. Montagnat, H. Delingette, N. Scapel, and N. Ayache. Representation, shape, topology and evolution of deformable surfaces. application to 3d medical image segmentation. Technical Report RR-3954, INRIA, 2000.
- [7] H. Delingette. Simplex meshes: a general representation for 3d shape reconstruction. Technical Report RR-2214, INRIA, 1994.
- [8] L.G. de la Fraga. Visualizing coefficients of matching functions over the three-dimensional space. In *In Proc. IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2002)*, pages 715 – 719, 2002.
- [9] R. Johnsonbaugh. *Discrete Mathematics*. Prentice Hall, 4th edition, 1997.
- [10] H. Delingette. General object reconstruction based on simplex meshes. Technical Report RR-3111, INRIA, 1997.
- [11] H. Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision.*, 32(2):111–146, September 1999.
- [12] H. Delingette. *Modélisation, Déformation et Reconnaissance d’objets tridimensionnels a l’aide de maillages simples*. PhD thesis, Ecole Centrale de Paris, 1994.

- [13] R. Marabini, I.M. Masegosa, M.C.San Martín, S. Marco, J.J.Fernández, L.G. de la Fraga, C. Vaquerizo, and J.M. Carazo. Xmipp:an image processing package for electron microscopy. *Journal of Structural Biology*, 116, 1996.
- [14] Luis Gerardo de la Fraga and Feliú Sagols Troncoso. Scubes: A program to visualize vox-solids. In *VII Conferencia de Ingeniería Eléctrica*. CINVESTAV, Septiembre 2001.
- [15] Charles H. Lehmann. *Geometria Analitica*. Limusa, 1947.
- [16] J.D. Foley, J.F. Hughes, Anries van Dam, and S.K. Feiner. *Computer Graphics:Principles and Practice Second Edition in C*. Addison Wesley., 1997.