

UNIVERSIDAD SAN PABLO – CEU

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA SUPERIOR DE TELECOMUNICACIONES



PROYECTO FINAL DE CARRERA

Estimación de los ángulos de proyección en procesos de
reconstrucción de volúmenes a partir de tomografías

Autor: Ignacio Fernández Moreno

Director: Carlos Óscar Sánchez Sorzano

Junio 2008

Indice

1	Introducción	
1.1	Resumen.....	1
1.2	Abstract.....	2
2	Estado de la técnica. Reconstrucción de volúmenes a partir de tomografías	
2.1	Visión general.....	5
2.2	Preparación de las muestras.....	6
2.2.1	Muestras completas.....	7
2.2.2	Seccionado ultra fino.....	7
2.2.3	Criofijación.....	8
2.3	Adquisición.....	8
2.3.1	Microscopía electrónica.....	8
2.3.2	Una muestra rotada.....	9
2.3.3	Muchas muestras rotadas aleatoriamente.....	9
2.3.4	Problemas.....	10
2.4	Corrección de la CTF.....	11
2.5	Procesamiento 2D de las proyecciones.....	11
2.5.1	Recorte y centrado.....	12
2.5.2	Alineamiento de la rotación.....	12
2.5.3	Clasificación.....	13
2.6	Cálculo del ángulo de proyección.....	14
2.7	Reconstrucción.....	16
3	Herramientas	
3.1	Sistema Operativo.....	19
3.2	Librerías Xmipp.....	19
3.2.1	Descripción y entorno.....	19
3.2.2	Arquitectura.....	21
3.2.3	Funcionalidades de reconstrucción.....	24
3.2.4	Elementos utilizados en el proyecto.....	30

3.3 Herramientas matemáticas.....	31
3.3.1 Ángulos de Euler.....	31
3.3.2 Teorema de la Sección Central.....	35
3.3.3 Teorema de la Proyección Común.....	37
3.3.4 Técnicas de minimización.....	38
3.3.5 Transformada de Radon.....	47
4 Desarrollo	
4.1 Problemática.....	50
4.2 Algoritmo.....	53
4.2.1 Esquema general.....	53
4.2.2 Idea inicial.....	55
4.2.3 Uso de la proyección común.....	57
4.2.4 Introducción de la minimización por partes.....	58
4.2.5 Desarrollo final.....	60
5 Resultados	
5.1 Resumen.....	63
5.2 Volúmenes a estudio.....	64
5.3 Obtención de la proyección común.....	66
5.4 Búsqueda de la proyección común.....	69
5.5 Obtención de ángulos de Euler de varios grupos de proyecciones.....	75
6 Conclusiones	

1 Introducción

1.1 Resumen

Existen numerosas aplicaciones para la reconstrucción de volúmenes a partir de proyecciones. En general, en todas ellas se hace necesario transformar la información que está repartida en múltiples imágenes en otro tipo de visualización a la que estamos más acostumbrados y que comprendemos mejor como son los objetos tridimensionales. Esto es especialmente importante y delicado en aplicaciones de bioingeniería, donde la tecnología debe actuar para que los biólogos puedan centrarse en el estudio de las diversas partículas y sus interacciones. Dicha reconstrucción se puede realizar de varias formas pero en este proyecto que se presenta nos centraremos en el caso en el que no conocemos los ángulos desde los que se tomaron las proyecciones.

El propósito final de este proyecto es la asignación automática de los ángulos de proyección desconocidos de una reconstrucción tridimensional a partir de la información de sus proyecciones basándonos en el conocido Teorema de la Proyección Común y sus derivados. Éste es un punto clave del proceso de reconstrucción ya que supone poder realizar una primera reconstrucción de referencia disponiendo únicamente de un grupo de imágenes previamente procesadas. Este primer resultado puede mejorarse entonces con técnicas de reconstrucción más precisas pero que necesitan una primera asignación de ángulos y, por tanto, un volumen de referencia.

Con la ayuda teórica del Teorema de la Proyección Común, tal y como se verá en el cuerpo de este proyecto, el problema se reduce a la búsqueda de un mínimo dentro de una función multivariable, cuya dimensión depende del número de proyecciones involucradas. Esta función tiene la particularidad de que, pese a ser derivable, es difícil el cálculo de su derivada y posee muchos mínimos locales, por tanto, no nos podemos basar en técnicas de minimización que se basen en el gradiente. Se han de utilizar, por tanto, técnicas de cálculo directo global como la Evolución Diferencial. Con el uso de

un algoritmo evolutivo el número de cálculos que se tienen que realizar para alcanzar el mínimo deseado disminuye drásticamente respecto a la técnica de fuerza bruta. En el caso que nos ocupa, en el que el número de dimensiones depende de la cantidad de datos que poseamos, este ahorro de cálculos se hace imprescindible.

Tras el estudio del problema y su desarrollo práctico se ha obtenido un algoritmo capaz de encontrar los ángulos de *Euler* desde los que fue tomada cada proyección introducida en relación a las demás proyecciones, de tal forma que todas las normales de los planos de proyección mantienen las mismas distancias que los originales formando así el mismo sistema de vectores, pese a que este sistema pueda tener una orientación en el espacio diferente como conjunto lo cual no afectará a la reconstrucción final, como se verá en la teoría. La eficacia del algoritmo estará relacionada con tres factores que el usuario puede modificar, el número de proyecciones introducidas, el número de generaciones del algoritmo de Evolución Diferencial y el número de individuos por generación.

Con un mayor desarrollo de este algoritmo y su extensión a otros casos como la utilización de ciertos tipos de simetrías para acelerar el cómputo se podría llegar a utilizar en ámbitos de estudio de partículas reales mejorando a otros algoritmos existentes en ciertos aspectos. Sería un buen complemento a las técnicas de reconstrucción actuales que necesitan un volumen inicial de referencia para comenzar su proceso iterativo.

1.2 Abstract

There are a number of applications to volume reconstruction techniques from projections. All of them are used to transform the information from a number of images into another kind of visualization that we understand better such as three-dimensional objects. This is particularly important and sensitive in bioengineering applications, where the technology must act so that biologists can focus on studying the various

particles and their interactions. This reconstruction can be done in several ways but in this project we present, we will center in the case where we do not know the angles from which the projections were taken.

The ultimate purpose of this project is the automatic assignment of unknown projection angles of a three-dimensional reconstruction process based on information from their projections based on the well-known Common Projection Theorem and its derivatives. This is a key point of the reconstruction process as it allows for an initial reference reconstruction having just a group of preprocessed images. This first result can then be improved with more accurate reconstruction techniques which need a first angle assignment and, therefore, a reference volume.

With the theoretical support of the Common Projection Theorem, as we will see in the body of this project, the problem boils down to finding a minimum of a multivariable function, whose dimension depends of the number of projections involved. This function has the peculiarity that, although it is derivable, it is difficult to calculate and has a lot of local minimum and, therefore, we can not use minimization techniques that are based on gradient. We must use direct estimation techniques such as Differential Evolution. Using an evolutionary algorithm the number of calculations that must be done to achieve the desired minimal decreases drastically with regard to the brute force technique. In the case of study, in which the number of dimensions depends on the amount of data we hold, this calculation savings are essential.

After the problem study and its practical development we have obtained an algorithm capable of finding the Euler angles of each projection relative to the other projections, so that all the projection plane normals hold the same distances as the originals forming the same vector system, despite the fact that this system may have a different space orientation as a whole which does not affect the final reconstruction, as will be seen in theory. The effectiveness of the algorithm is related to three factors that the user can modify, the number of projections, the number of generations for the Differential Evolution algorithm and the population per generation.

With further development of this algorithm and its extension to other fields such as the use of certain types of symmetries to accelerate the computation it could be used in real particle studies improving existing algorithms in some aspects. It would be a good complement to current reconstruction techniques that require an initial reference volume to begin its iterative process.

2 Estado de la técnica. Reconstrucción de volúmenes a partir de tomografías

El objetivo de este apartado es explicar las técnicas que se utilizan en los procedimientos de reconstrucción de volúmenes actuales a partir de tomografías. Para entender el objetivo final del proyecto es necesario tener un conocimiento básico del proceso general, para luego entrar en detalle en la parte de este proceso que nos interesa, que es la determinación de los ángulos de proyección. Para ello se comentarán las ideas que cada uno de los pasos del proceso lleva a la práctica sin entrar demasiado en el apartado matemático que los sostiene. Se explicarán las características más interesantes de todos ellos y qué se obtiene tras su ejecución. Entre las técnicas comentadas se encuentran la preparación de las muestras, la adquisición, el cálculo y posterior corrección de la *CTF* (*Contrast Transfer Function*), la aplicación de un formato común, la determinación de la rotación, y la aplicación del teorema de la sección central al problema de la reconstrucción tridimensional.

2.1 Visión general

En la actualidad, la microscopía electrónica en general y la reconstrucción de volúmenes en particular han sido de gran ayuda a la hora de ampliar el conocimiento de la biología moderna sobre las estructuras que forman parte de muchos procesos biológicos. Con ello se ha conseguido una visión mucho mejor sobre la morfología de las moléculas y sus uniones, teniendo como resultado la creación de mejores medicamentos que fomentan o anulan dichas uniones y sus efectos. Debido a esto en la última década se han realizado grandes avances en el tratamiento de las imágenes que se obtienen de los microscopios electrónicos (tomografías) aprovechando hasta la mínima información que poseen. En este breve resumen se da una idea general de todo el proceso, desde que se recogen las imágenes hasta que se genera el volumen final.

Existen diversas técnicas para la obtención de la información requerida en cada fase del proyecto, esto puede deberse, por ejemplo, a las diferentes características de los microscopios de adquisición o a la existencia o no de conocimientos *a priori* sobre la estructura molecular. Aquí se describirán las más importantes, pero solo se entrará en detalle en las que forman parte del proceso en el cual a partir de proyecciones de la molécula en ángulos aleatorios, y sin información *a priori*, se obtiene el mejor modelo tridimensional posible.

2.2 Preparación de las muestras

Para los microscopios electrónicos de transmisión (*TEM*) lo más común es que las muestras sean delgadas, estén secas y contengan moléculas que difracten los electrones. Los espécimenes biológicos, en cambio, son grandes, están formados en su mayor parte por agua y no difractan los electrones por lo que son difíciles de visualizar con un *TEM*. Se hace imprescindible una preparación previa de las muestras para el *TEM* que conserve la estructura del material, lo cual es complicado. Por fortuna, los materiales biológicos han sido estudiados durante bastante tiempo y existen diversos protocolos que nos permiten visualizar las muestras biológicas de diferentes maneras. Sólo mostraremos aquí una corta descripción de los más importantes puesto que no forman parte del objetivo principal de este proyecto.

2.2.1 Muestras completas

Los objetos pequeños o muy delgados pueden ser examinados montándolos directamente sobre una placa de soporte que se introduce directamente bajo el haz de electrones. El contraste se consigue con la precipitación de metales pesados en una de estas tres maneras.

Tinción positiva: Se tiñe la muestra con una solución de sal metálica que se ve negra sobre fondo blanco.

Tinción negativa: No se tiñe el objeto sino que se introduce en un soporte de sales metálicas pesadas. Es la más utilizada en el campo de la microscopía electrónica.

Sombreado: Una capa delgada de átomos de metales pesados se deposita en el espécimen mediante evaporación en una cámara de vacío.

2.2.2 Seccionado ultra fino

La técnica más popular para examinar especímenes biológicos es introducir el material bajo estudio en plástico y cortar secciones ultra finas que pueden ser examinadas en un *TEM*. El proceso comienza por fijar químicamente el material, luego se obtiene contraste con soluciones de sales de metales pesados, se deshidrata en etanol o acetona y se recubre de plástico. Una vez tenemos la muestra procesada se utilizan cuchillas de cristal o diamante para cortar secciones ultra finas (60nm) mediante un ultramicrotomo, entonces se introducen en agua y se colocan en el soporte para ser examinados al microscopio.

2.2.3 Criofijación

Es posible congelar las muestras lo suficientemente rápido para vitrificar el agua del interior de las células. Ésto ocurre cuando el proceso de congelación se ha realizado tan rápido que los cristales de hielo no han tenido tiempo para formarse. Los espécimenes así tratados pueden ser seccionados a bajas temperaturas. Estas secciones se pueden observar en *TEMs* que estén correctamente equipados para mantener las muestras frías.

2.3 Adquisición

2.3.1 Microscopía electrónica

Los microscopios electrónicos actuales siguen unos principios bastante parecidos a los ópticos ya que en ambos se irradia un espécimen con una fuente de energía y cuando el rayo incide en él rescata la información de su estructura y la plasma en algún dispositivo receptor. En el caso de la microscopía electrónica la fuente de energía es un cátodo que irradia electrones a lo largo de un tubo de aceleración compuesto por ánodos y que son enfocados hacia el espécimen mediante lentes magnéticas. Algunos electrones se desvían al traspasarlo, pero la mayoría mantienen su trayectoria sufriendo cambios de fase que se utilizan para formar una imagen. Estas imágenes, debido a la naturaleza de la adquisición, no son fotografías ordinarias, sino proyecciones bidimensionales del volumen estudiado. Esto es, cuanto más blanco es un píxel, mayor es el volumen que el electrón ha tenido que traspasar. Esta característica es la que da pie a los algoritmos de reconstrucción tridimensional actuales.

Para la reconstrucción es necesario que existan proyecciones del objeto desde varios ángulos diferentes, para conseguirlo se utilizan dos técnicas diferentes que tienen ciertas ventajas y ciertos inconvenientes que se explicarán en las secciones siguientes.

2.3.2 Una muestra rotada

La primera técnica de obtención de proyecciones de la molécula desde distintos ángulos es la adquisición de proyecciones de una sola molécula aislada desde varios ángulos mediante la rotación del microscopio. En la actualidad existen unidades capaces de hacer esto automáticamente. La ventaja principal es que los ángulos desde los que se han tomado las proyecciones son conocidos, por lo que no hay que calcularlos y podemos saltarnos ese paso con el consiguiente ahorro de tiempo y potencia de cálculo. Además siempre va a ser mejor el conocimiento que tenemos sobre la rotación de la molécula si sabemos desde dónde hicimos la tomografía que el que pueda darnos cualquier algoritmo de reconocimiento de rotaciones dentro de una muestra aleatoria. El mayor inconveniente es que los microscopios electrónicos, debido a limitaciones técnicas por ahora insalvables sólo son capaces de rotar 60 grados en ambos sentidos, es decir, abarcan un ángulo de 120°, lo que suele resultar insuficiente para moléculas donde no existe ningún tipo de simetría.

2.3.3 Muchas muestras rotadas aleatoriamente

La segunda técnica se basa en moléculas con rotación aleatoria. Se prepara una muestra con miles de especímenes del mismo tipo en la que cada uno ha tomado una rotación cualquiera y se adquiere una micrografía con todos ellos. Así nos aseguramos de haber cogido todos los ángulos posibles. La desventaja de este método es, precisamente, la ventaja del anterior, aunque los ángulos de rotación abarcan todos los posibles, no son conocidos por lo que tienen que ser estimados. La estimación de estos

ángulos es, precisamente, el objetivo de este proyecto y para ello se introducirá una técnica novedosa que mejora las prestaciones de las que hay en la actualidad.

Además de determinar las rotaciones hay que hacer un reconocimiento de cada proyección por si aparece alguna molécula externa no deseada que pueda estropear todo el estudio.

La técnica de rotaciones aleatorias requiere un mayor número de cálculos, sin embargo es la mejor para crear volúmenes de moléculas sin simetría. Además, las técnicas de tratamiento de imágenes han avanzado mucho y es posible reconocer ángulos y figuras de manera óptima sin que suponga demasiada carga añadida.

2.3.4 Problemas

Para finalizar esta sección deben comentarse dos de los mayores inconvenientes derivados del uso de los microscopios electrónicos en cualquiera de las técnicas descritas, estos son el nivel extremadamente bajo de la relación señal a ruido de las micrografías y la función de transferencia de contraste que el propio microscopio añade al sistema, más conocida como *CTF*. La aparición de ruido es inevitable trabajando en porciones tan pequeñas de espacio y con las técnicas actuales de adquisición. Podrían aplicarse técnicas de filtrado de ruido, pero como el objetivo es capturar la mayor cantidad de información posible esto no tiene mucho sentido. Sin embargo la *CTF* puede calcularse y corregirse, siendo un proceso imprescindible para la obtención de una mínima calidad en las imágenes. Por su importancia se describe en la siguiente sección.

2.4 Corrección de la CTF

Este paso se ha convertido en una necesidad en los procedimientos de reconstrucción de volúmenes debido a los grandes efectos negativos de la Función de Transferencia Constante en la precisión de las imágenes obtenidas.

La corrección de la *CTF* se suele realizar mediante un proceso de dos pasos, el primero consiste en calcular la densidad espectral de potencia o *PSD* (*Power Spectral Density*) de la imagen y después se filtra con un modelo teórico de la *CTF* que la estima en la medida de lo posible. Este modelo se suele computar mediante la técnica del promedio de periodogramas [1], aunque también se utilizan estimaciones paramétricas con modelos *ARMA* como en el caso del paquete de procesamiento de imágenes *XMIPP*.

En este punto del proceso tan sólo se corrigen los errores de signo causados por la *CTF* porque cuanto antes se realice esto mejores resultados se obtienen en los siguientes pasos. Más adelante, durante la reconstrucción tridimensional, se corregirán los errores de amplitud.

2.5 Procesamiento 2D de las proyecciones

Este apartado tiene que ver directamente con el que dedicamos a la adquisición. Como se ha obtenido de la fase anterior una micrografía donde se encuentran las moléculas en posiciones y ángulos aleatorios hay que seguir tres pasos importantes antes de continuar con el proceso. En orden de aplicación éstos son el recorte centrado de cada molécula simple, la alineación de sus rotaciones y la clasificación por características comunes. De esta fase obtenemos una población homogénea de proyecciones ordenada y clasificada.

2.5.1 Recorte y centrado

En este momento contamos con una gran imagen en la que muchas moléculas de la misma clase están colocadas en diferentes posiciones con diferente rotación. Ahora deseamos aislar cada una de las moléculas en una imagen simple y para ello hemos de recortar la imagen general. Esto se puede hacer mediante procedimientos automáticos o permitiendo que el usuario seleccione manualmente las zonas en las que se encuentra cada una de las proyecciones.

En el caso del procedimiento automático se han de obtener todas aquellas zonas de la imagen donde exista cierta cantidad de potencia por encima de un umbral determinado. Hay que escoger un umbral suficiente para que la potencia del ruido de la imagen no provoque la selección de zonas sin proyección. Una vez determinadas estas zonas se recortan al tamaño estándar y se centra la imagen colocándola en su centro mediante diversas técnicas, por ejemplo se puede utilizar el centro de masas de la imagen.

En el caso de la selección manual se le presenta al usuario la imagen general y se le permite seleccionar los puntos donde él estima que se encuentran algunas de las partículas, después mediante un algoritmo se vuelven a colocar los puntos seleccionados para centrarlos sobre la partícula en base a su centro de masas, de forma similar a como se hace en el caso de selección automática. La ventaja de éste método es que contamos con una información *a priori* de la posición de las partículas. También se puede realizar un posicionamiento previo a la selección del usuario para que éste cuente con una orientación sobre las posiciones de las partículas y, tras su selección efectuar el proceso de recorte.

2.5.2 Alineamiento de la rotación

Dado que ahora tenemos una serie de matrices de información de la misma proporción que contienen todas las proyecciones a estudiar, lo que debemos hacer a

continuación es alinearlas para poder discriminar mejor a que grupo de proyecciones pertenecen y, facilitar así su posterior asignación. Para conseguir el alineamiento se selecciona una primera partícula que será la representativa de su grupo de similitud, se va estimando la correlación cruzada entre ella y las demás en sucesivas rotaciones y se trata de escoger la orientación que minimiza dicha correlación. Dadas estas correlaciones las que pasen de un umbral serán seleccionadas como similares a la partícula de referencia y orientadas según el ángulo para el que se dio dicha correlación. De entre las partículas no seleccionadas se escoge una nueva referencia y se repite el proceso hasta que todas las partículas estén orientadas. Tras esto se obtiene una gama de proyecciones orientadas por grupos de similitud.

2.5.3 Clasificación

Para evitar el uso de la información de una proyección de un espécimen diferente al del estudio se realiza una comprobación adicional. Además de este paso se obtiene una agrupación de las tomografías por características comunes que más adelante ayudarán en los procesos de reconstrucción.

Con este propósito se utilizan técnicas clásicas de reconocimiento de patrones y de grupos para obtener un vector de características de cada proyección. Existen una gran variedad de características que se pueden tener en cuenta desde las más simples, como el rango de valores de los píxeles hasta las más complejas como el espectro rotacional.

La inclusión de unos u otros en el análisis depende de los objetivos de la reconstrucción. Los vectores se introducen en un algoritmo de agrupación que permite al usuario controlar la creación de sub-poblaciones con similares características. Las dos técnicas más populares en este campo son la Clasificación Jerárquica Ascendente [2] y los mapas auto-organizados [3].

Debido a que una buena agrupación depende de la correcta alineación de sus componentes y ésta se obtiene cuando se aplica a grupos de similares características no

es suficiente con realizar los pasos 2.5.1 Recorte y centrado, 2.5.2 Alineamiento de la rotación y 2.5.3 Clasificación una sola vez. Por ello se aplican procesos iterativos que se detienen cuando, tras sucesivas traslaciones, rotaciones y agrupaciones los grupos cambian poco o nada.

Tras la agrupación el proceso concluye cuando las imágenes del mismo grupo, que consideramos proyecciones de la partícula desde los mismos ángulos tridimensionales, se combinan extrayendo su información para crear una sola proyección con una relación señal a ruido mejorada y unas características más definidas. Esto se hace así porque la señal a ruido que se obtiene en las micrografías del microscopio electrónico es baja y contamos con muchas proyecciones repetidas que contienen distinta información de la proyección real. Debido a esto el recorte, centrado y la agrupación de micrografías son pasos esenciales y delicados, ya que si no se resuelven bien podemos estar combinando proyecciones que no tienen nada que ver, con lo que se obtendrían imágenes aún más distorsionadas que las que teníamos en un principio.

2.6 Cálculo del ángulo de proyección

Llegados a este punto del proceso poseemos una población de proyecciones separadas por grupos de características y posicionadas de un modo homogéneo. Además tenemos cierta certeza de que todas pertenecen a la estructura bajo estudio. Debido a que en nuestro caso desconocemos la orientación en el espacio de las proyecciones antes de comenzar la reconstrucción tridimensional es necesario calcularla para realizar un análisis correcto. Sobre todo es importante que conozcamos la orientación relativa entre sí, puesto que la posición absoluta se puede fijar al conocer la orientación de la primera proyección ya que no tiene relevancia en el estudio. Recordemos que lo único que nos interesa es la morfología de la partícula bajo estudio.

Una técnica para el cálculo del ángulo de proyección se realiza basándose en el Teorema de la Sección Central (Ver punto 3.3.1) y en el Teorema de las Proyecciones

Comunes (Ver 3.3.3), que son la base, además, de las técnicas de reconstrucción tridimensional.

Dados los dos teoremas explicados se pueden calcular los ángulos relativos entre tres proyecciones 2D basándose en sus proyecciones 1D compartidas dos a dos. Comenzamos calculando las proyecciones 1D de una de las tres proyecciones 2D del volumen. Esto se suele hacer a lo largo de un ángulo de 180° (pues las proyecciones son iguales desde ambos lados salvo por una operación especular) con intervalos de 1° a 3° dependiendo de la potencia de cálculo disponible y de la calidad deseada. Estas proyecciones suelen colocarse como filas de una imagen llamada sinograma, donde el grado de gris da el valor normalizado de cada muestra, el eje y son los grados a los que tomó cada proyección 1D y el x cada una de esas proyecciones. El siguiente paso es calcular la Función de Correlación entre Sinogramas (*SCF*) para cada pareja de éstos. Se procede calculando los coeficientes de la correlación cruzada entre la primera fila del primer sinograma y cada una de las filas del segundo sinograma. Estos coeficientes se colocan en la primera fila de la *SCF*. La segunda línea se rellena con la correlación cruzada entre la segunda fila del primer sinograma y cada una de las filas del segundo, luego la tercera, la cuarta y así se obtiene la *SCF* entre el primer sinograma y el segundo. Existen varias formas de encontrar el máximo de la *SCF*, que se correspondería al punto en el que la correlación de dos proyecciones 1D es máxima. Se puede hacer en el espacio de Fourier como en [4] y [5], sin embargo, ya que los resultados son idénticos, una posibilidad es hacerlo en el espacio real mediante un procedimiento de correspondencia de los mínimos cuadrados con una función paraboloidal en un entorno de los mayores máximos para afinar la búsqueda. Obtenido este máximo podemos saber en que ángulos se corresponden las proyecciones 1D de cada pareja y por tanto los diferentes ángulos entre las tres líneas comunes.

Así, hemos obtenido los ángulos relativos entre las tres proyecciones 2D. Para posicionarlas en un entorno absoluto basta con fijar una de ellas y las demás vendrán impuestas por los ángulos de inclinación relativos. Sabemos que los ángulos de inclinación obtenidos son perpendiculares a las direcciones de las dos proyecciones de

las que se obtuvieron. Así, si sabemos que el primer ángulo de inclinación tiene la dirección del eje z podemos colocar dos proyecciones si mantienen sus normales a 90° del ángulo de inclinación, pudiendo rotar cada una 360° como se ve en la Ilustración 1 en la imagen de la izquierda. Si incluimos entonces una tercera proyección que tiene un ángulo de inclinación con cada una de ellas se pierde la rotación al aparecer tres líneas comunes y se establecen los ángulos relativos entre las proyecciones como se muestra en la Ilustración 1 a la derecha.

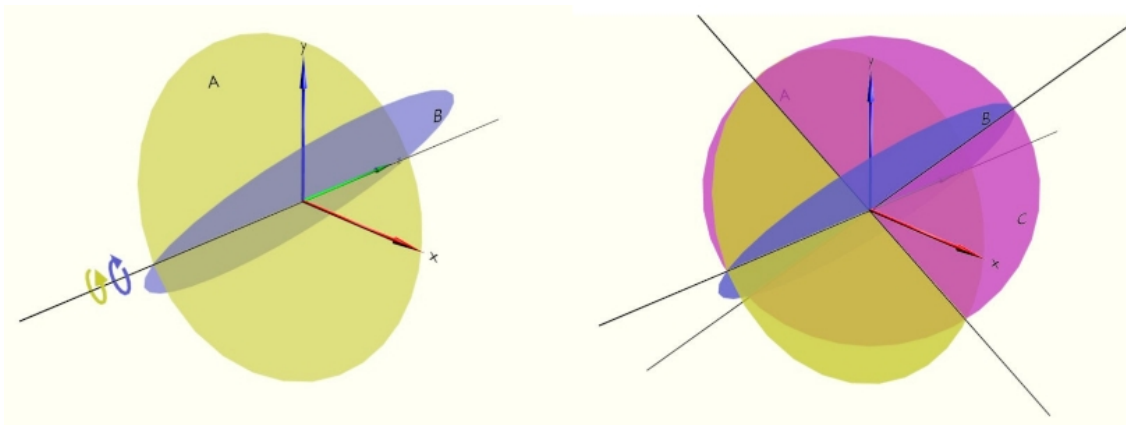


Ilustración 1: Intersección de planos en tres dimensiones

El proceso puede ser repetido entre proyecciones 2D ya colocadas en el espacio absoluto y las que aun no estén colocadas para poder proceder a la reconstrucción.

2.7 Reconstrucción

Ahora que por fin se han obtenido unas proyecciones del objeto razonablemente libres de ruido, con un formato homogéneo y orientadas en el espacio solo nos queda crear un volumen a partir de ellas. La idea general de todos los algoritmos utilizados en este proceso consiste en minimizar el error entre las proyecciones de muestra y unas nuevas proyecciones obtenidas del objeto tridimensional que se genere desde los mismos ángulos que las originales. Es lógico pensar que si este error es mínimo tenemos la mejor reconstrucción posible con las proyecciones de que disponemos, esto

no quiere decir que sea una reconstrucción exacta, sino solo la mejor que podemos obtener.

Los algoritmos utilizados en Microscopía Electrónica para partículas simples son *Weighted Back-Projection* (o *Filtered Back-Projection*) y algoritmos de Expansión en Series [6]. La Retro-proyección Ponderada (*Weighted Back-Projection*) se basa en el Teorema de la Sección Central utilizándolo de forma directa para obtener la transformada de Fourier del volumen final. Sin embargo, como suele ocurrir en todas las aplicaciones basadas en el mundo real, el Teorema de la Sección Central tiene unas premisas que no se cumplen exactamente. Una de las más importantes es que los planos en el espacio de Fourier no son planos matemáticos infinitamente finos, sino que tienen un grosor aproximado de $1/D$, donde D es la distancia de la dimensión lineal más grande. En la Ilustración 2 se puede apreciar muy bien como los planos en los que tenemos información se solapan en una zona tridimensional alrededor de la línea común, de modo que en la zona más cercana al centro (las frecuencias bajas) existe mucha más información que a distancias más lejanas (las frecuencias altas). Por ello al transformar del espacio de Fourier 3D al espacio real obtenemos un volumen emborronado. Para solucionar esto se introdujeron en este campo filtros paso alto (de ahí el nombre *Filtered Back-Projection*) obtenidos primero de forma analítica [7] y después mediante algoritmos de Filtro Exacto [7] y [8].

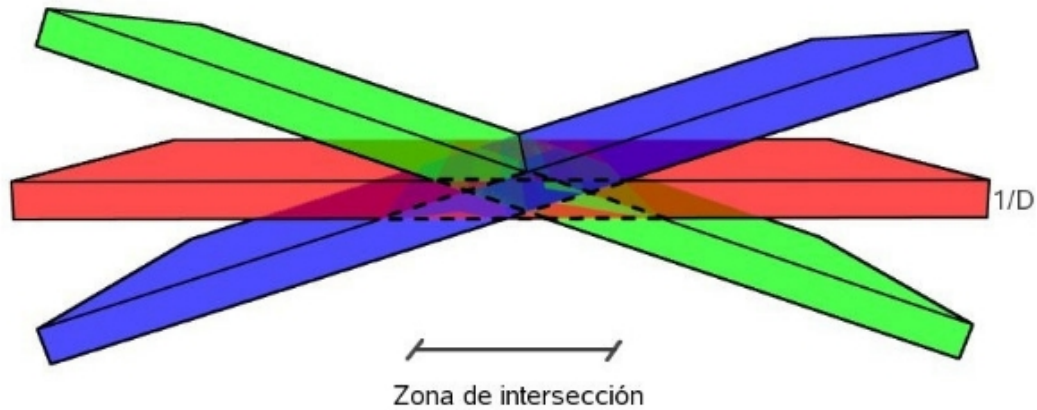


Ilustración 2: Zona de intersección de planos con grosor $1/D$

Por otro lado los algoritmos de Expansión en Series toman el volumen como una suma de funciones simples multiplicadas por ciertos coeficientes y trasladadas en el espacio. El cálculo del volumen se convierte entonces en la resolución de un sistema de ecuaciones lineales cuyas incógnitas son los coeficientes antes mencionados. Las funciones base que se suelen escoger van desde los *voxels* (constantes en el volumen) hasta los *blobs* (funciones ventana de *Kaiser-Bessel* en el volumen), siendo estos últimos más apropiados como se describe en [9]. Otra decisión importante es la elección de la malla en la que se colocarán los centros de las funciones base. La malla cúbica centrada en el cuerpo (*BBC grid*) ha demostrado ser la mejor solución dando mejores resultados que la malla cúbica simple (*SC grid*). Técnicas basadas en algoritmos de Expansión en Series como *ART* (*Algebraic Reconstruction Technique*) se utilizan en la actualidad para la reconstrucción de volúmenes en paquetes informáticos como *XMIPP* (Ver 3.2).

3 Herramientas

3.1 Sistema Operativo

La plataforma elegida es GNU/Linux [10]. Las principales características de este sistema operativo es que es libre y dispone de buenas herramientas de desarrollo y programación, así como de documentación y realización de diagramas.

La elección de esta herramienta está basada en la existencia de un conjunto de librerías de código abierto que se presentarán en profundidad en la sección 3.2.

Linux se puede instalar compilando el núcleo e instalando las herramientas necesarias, pero lo más común y lo más fácil es instalar una distribución que ya contiene todo esto listo para utilizarse. Una de las que más apoyo está recibiendo en la actualidad, y la utilizada para este proyecto, es Ubuntu, con el gestor de escritorio y ventanas *Gnome*. Esta distribución es sencilla en su instalación y mantenimiento y permite la instalación de aplicaciones de desarrollo esenciales como el compilador de `c++ gcc` [11] y la herramienta de compilación e instalación de proyectos *make* [12], así como las librerías asociadas.

3.2 Librerías Xmipp

3.2.1 Descripción y entorno

La reconstrucción de volúmenes requiere de técnicas computacionales avanzadas, cálculo de transformadas de Fourier, algoritmos de minimización, rotación de imágenes, así como de una base de herramientas de desarrollo de conceptos matemáticos sencillos como vectores, matrices, volúmenes y las diferentes operaciones entre ellos. La aplicación elegida, *Xmipp* [9], es una *suite* de código abierto que contiene

librerías y programas que dan acceso a las funcionalidades descritas arriba y a muchas más, además de ser un elemento indispensable para la realización de pruebas y experimentos relacionados con este campo.

En este punto se describirán muchas de las funcionalidades de esta aplicación incluyendo todas las que se hayan utilizado tanto para el desarrollo como para las pruebas posteriores.

Xmipp es el acrónimo de *X-windows based microscopy image processing package*, es decir, es un paquete de procesamiento de imágenes microscópicas basadas en el interfaz de ventanas para *GNU/Linux X-windows*. Está desarrollado en el lenguaje de programación *c++* y aprovecha toda la capacidad de la programación orientada a objetos, presentando elementos como imágenes o proyecciones. Para los componentes gráficos del programa se escogió la librería *qt*. Esto lo convierte en una aplicación portable a cualquier sistema que sea compatible con estas librerías y *c++*, lo que incluye cualquier sistema basado en UNIX, así como los sistemas *Windows* mediante la emulación proporcionada por *Cywin*.

La idea inicial de *Xmipp*, presentado por primera vez en 1996, estaba enfocada a la reconstrucción de partículas simples. Además, se basó, en cuanto a formatos y convenciones matemáticas (ángulos de Euler, orientaciones de los ejes, etc) en el paquete de reconstrucción de volúmenes *Spider*, uno de los referentes en este campo. La compatibilidad con otras aplicaciones se consigue mediante conversores de formato *em2em* [13] o *bimg* [14], que permiten exportar e importar los datos desde los principales paquetes de tratamiento de imágenes (*EMAN*, *IMAGIC*, *MRC*, etc) en cualquier punto del proceso de reconstrucción.

Dado que *Xmipp* ha sido desarrollado bajo la filosofía del código abierto tanto las fuentes como los programas están disponibles en los repositorios de dominio público *SourceForge* (<http://www.sourceforge.net>) para ser descargados por cualquier persona interesada en el uso del paquete y en su mejora. Además la descarga incorpora una

extensa documentación en formato *html*, donde se detallan los distintos programas y el Api de c++. Esta documentación se encuentra disponible en formato *Wiki* editable en <http://xmipp.cnb.csic.es>.

3.2.2 Arquitectura

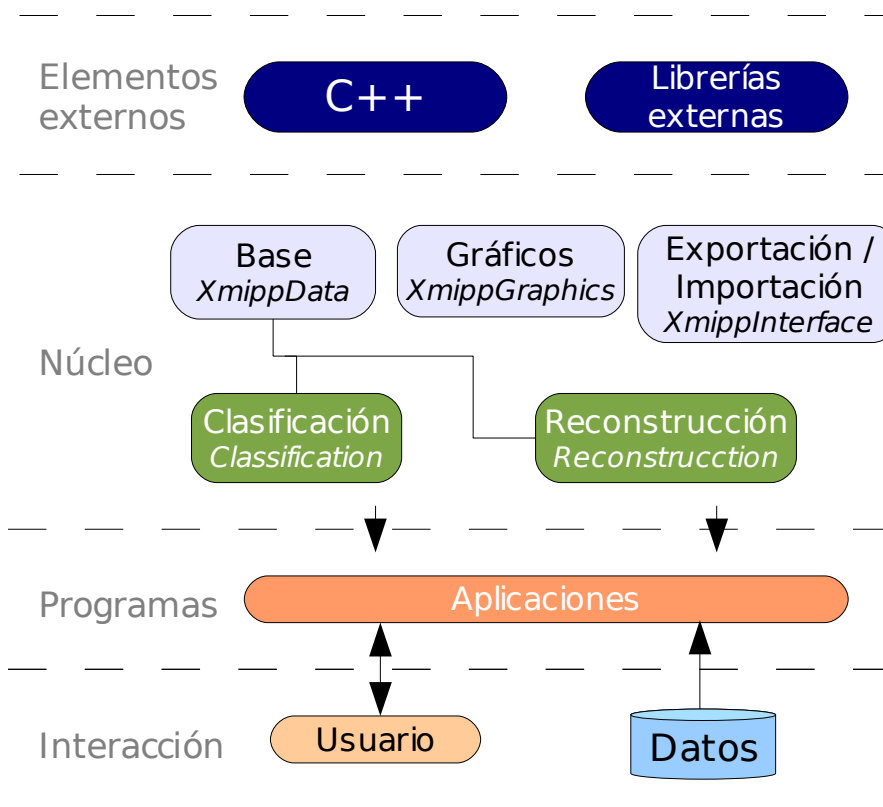


Ilustración 3: Arquitectura de Xmipp

- **Elementos externos:** Como ya se ha dicho *Xmipp* está programado en *C++*. Además utiliza las librerías *Qt* para la visualización.
- **Núcleo:** Grupo de funcionalidades de soporte que incluyen estructuras y algoritmos comúnmente utilizadas en los programas de reconstrucción 3-D. Se puede dividir en:

- Estructuras y funciones matemáticas base (*XmippData*): Contiene todas aquellas herramientas matemáticas desarrolladas para *Xmipp* a un nivel no orientado a la reconstrucción 3D. Algunos ejemplos de objetos contenidos son vectores, matrices, imágenes, todos ellos con funciones para interactuar entre ellos (multiplicación de matrices, vectores, producto vectorial, etc) y sobre sí mismos (rotaciones, traslaciones, Matriz inversa, etc), además de algunas funcionalidades matemáticas genéricas pero más avanzadas, como transformadas de Fourier multidimensionales o una implementación de *Differential Evolution* extraída de [15] y adaptada para usos propios. Contando únicamente con este bloque *Xmipp* se puede ver como un paquete matemático avanzado y puede ser utilizado en la resolución de problemas de cierta complejidad.
- Funciones de visionado de proyecciones, imágenes y volúmenes (*XmippGraphics*): Estas librerías proporcionan un Api para la correcta visualización de estructuras complejas como imágenes o volúmenes en el entorno en el que corran las librerías *qt*.
- Algoritmos, estructuras y funciones propias del proceso de reconstrucción (*Classification, Reconstruction*): Incorpora herramientas avanzadas enfocadas en la resolución de los diferentes pasos del proceso de reconstrucción, tanto para la clasificación (punto 2.5.3) como para la reconstrucción (punto 2.7). Estas utilidades son de mayor nivel que las que aparecen en *XmippData* y, aunque algunas pueden ser utilizadas como base para experimentos en otras áreas, han sido diseñadas para cumplir con los objetivos de este tipo de procesos específicos. Contiene utilidades como la corrección de la *CTF* o las estructuras tipo *blob*.
- Funciones para la exportación y la importación de datos (*XmippInterface*): Este paquete contiene clases y métodos para la conversión del formato de datos de *Xmipp* a otros paquetes de tratamiento de señales como *Spider*, de visualización de datos como *OpenDX* o a otros formatos como *JDL (Job Description Lenguaje)*.

- **Programas:** Aplicaciones que utilizan funcionalidades del núcleo para la implementación de rutinas ejecutables que utilizan datos reales almacenados en el sistema. El usuario no programador interactuará con estos programas introduciendo sus propios datos bajo estudio para la obtención de un volumen reconstruido o para la realización de uno de los pasos del proceso. Recordemos que gracias a que *Xmipp* está diseñado de forma modular y existen las funcionalidades de exportación e importación a otros formatos el investigador puede utilizar las herramientas de reconstrucción de *Xmipp* que prefiera y después utilizar otras ajenas que se ajusten a sus necesidades.

La lista de aplicaciones contenidas en *Xmipp* es extensa, con más de 100 programas y está, como ya se ha dicho, bien documentada. Estos programas son tan sólo una interfaz que gestiona los datos introducidos por el usuario mediante la línea de comandos UNIX y ejecuta las funcionalidades contenidas en el núcleo. Un usuario experto puede realizar *Scripts* de *Shell* para automatizar partes del proceso, o incluso el proceso completo. Además, algunos de estos programas cuentan con una interfaz gráfica, como en el caso de la selección manual de partículas.

Algunos ejemplos de estos programas son *FSOM* [16], para el cómputo de Mapas borrosos auto-organizables (*Fuzzy Self-Organizing Map*), *AssignCTF* para la búsqueda de la función de transferencia constante y *project* para realizar proyecciones planas de volúmenes según una serie de rotaciones dadas.

- **Interacción:** El usuario puede interactuar con las funcionalidades de la aplicación introduciendo datos en cualquier punto del proceso.

3.2.3 Funcionalidades de reconstrucción

En esta sección se comentarán las soluciones implementadas por *Xmipp* para cada uno de los pasos de la reconstrucción vistos en el capítulo 2. La idea es tener una visión general de las capacidades de esta suite de programas a la vez que se muestran las rutinas más avanzadas para la resolución de cada paso, ya que *Xmipp* es una herramienta que se mantiene al día de las últimas técnicas experimentales desarrolladas para cada problema concreto de la reconstrucción.

- **Algoritmos de clasificación**

1. *Self-Organizing Maps (SOM)*

Xmipp se basa en las redes neuronales como base para los distintos algoritmos de clasificación. En concreto se centra en los *SOM* [17] que poseen buenas características para la discriminación de datos y la clasificación. La idea detrás de los Mapas Auto Organizables o *SOMs*, es que dado un conjunto multidimensional de características del sistema relacionadas es capaz de reducir el número de dimensiones a dos (una malla) combinando los distintos valores multidimensionales en proyecciones bidimensionales mejoradas en cuanto a su precisión. Esto es, obtener un mapa de características resumidas a partir de un complejo grupo de datos inicial. Es un proceso iterativo en el que en cada iteración se van escogiendo uno a uno los vectores representados y calculando el efecto que tienen sobre los demás. Si están lo suficientemente cerca son atraídos por el valor del vector actual. Este proceso se denomina *Kohonen SOM*, para diferenciarlo de los desarrollos posteriores.

El primero que introdujo este algoritmo al proceso de clasificación fue [18], y su idea fue extraer características de las proyecciones obtenidas del microscopio electrónico y utilizarlas como vectores que las representasen, estos vectores se denominan *Codevectors*. Estas características pueden ser de muchos tipos como por

ejemplo la media, la relación señal a ruido, la resolución, etc. Estas características están en un espacio multidimensional por que se puede escoger cualquier número de características de las imágenes para clasificarlas, además sus valores son muy cambiantes debido a que las proyecciones se extraen del microscopio electrónico en múltiples perspectivas de la molécula bajo estudio, en posiciones aleatorias y con altos niveles de ruido, sin olvidar que puede que en la muestra se presenten otras partículas o esa misma partícula en un estado no deseado y, por tanto, los vectores de características que se obtienen de ellas se muevan por todo el espacio de características multidimensional. Aplicando *SOM* al conjunto de proyecciones y uniéndolas según la similitud de sus vectores de características es posible obtener grupos de proyecciones con vectores similares. Combinando estas proyecciones por grupos se puede obtener una proyección con características mejoradas (por ejemplo su resolución) por grupo. Además, si uno de esos grupos de características similares está relativamente lejos del resto, y tiene pocos vectores a su alrededor probablemente se tratará de una partícula diferente a la del estudio o en un estado no deseado.

En el ámbito de *Xmipp* el desarrollo más reciente es *KenderSOM* (*Kernel Density Estimator Self-Organizing Map*). Sobre la base de los *SOM*, utiliza una función de coste para la selección de los vectores de características a combinar que trata de mantener la densidad de probabilidad entre los vectores multidimensionales y los bidimensionales [3]. La ventaja respecto a los *SOM* clásicos es que combina la visión intuitiva de los *SOM* con la precisión matemática de la estimación de probabilidad. Esta técnica se ha mostrado muy efectiva a la hora de discriminar elementos ajenos en el conjunto de proyecciones, que es uno de los pasos más difíciles en el proceso de clasificación.

Hay otros dos algoritmos basados en *SOMs* implementados en *Xmipp*, que aunque no tienen tanta relevancia como el anterior son útiles en ciertos casos. El primero es *BatchSOM*. La diferencia entre el *BatchSOM* y el *Kohonen SOM* es que los vectores de características no se presentan uno a uno al mapa auto-organizable, sino como un todo. De tal forma que ahora no se evalúa el efecto que cada vector tiene sobre

los demás uno a uno, sino todos a la vez. Esta técnica tiene una convergencia más rápida que el *SOM* clásico.

El otro algoritmo se denomina *FuzzySOM* (*SOM* borroso). Se basa en la aplicación de una función de coste derivada del algoritmo de *clustering fuzzy c-means* [16]. Mientras que en los *SOM* tradicionales los vectores eran afectados por otros a una distancia fija, y si se superaba esa distancia no eran afectados, en este caso la función distancia termina de forma suave, de tal manera que un vector muy cercano será afectado de forma completa, mientras que uno relativamente alejado, pero no demasiado será atraído sólo parcialmente.

2. Otros algoritmos de mapeo y agrupación

Xmipp también posee algoritmos de reconocimiento de patrones clásicos y métodos estadísticos. La búsqueda de grupos, más conocida por el vocablo inglés *clustering* tiene como finalidad la separación en diferentes conjuntos de los vectores de características para así clasificar las proyecciones que representan. Para ello se suele recurrir a técnicas de *clustering* que reducen la dimensión del problema eliminando la información menos útil y manteniendo ciertas propiedades del espacio. *Xmipp* implementa métodos lineales, como *Principal Component Analisis* (PCA) [19], y métodos no lineales, como *Sammon mapping* [20].

Otros métodos clásicos implementados son *Fuzzy c-Means* [21], *Kernel c-Means* [22] y [23], and *Fuzzy Kohonen Clustering Network* [24].

- **Estimación y corrección de la CTF (*Contrast Transfer Function*)**

Como ya se ha explicado en 2.4 Corrección de la CTF, la corrección de la fase distorsionada por la CTF es un paso esencial en el proceso para la mejora de la calidad de las proyecciones y, por tanto, de la reconstrucción final. Disponer de herramientas eficaces que calculen la CTF y la anulen es imprescindible en una aplicación como

Xmipp.

El primer paso es estimar la Función de Transferencia de Contraste que se aplicará a cada proyección, para ello *Xmipp* posee un proceso automatizado para asignar a cada tomografía la *CTF* que, teóricamente le correspondería. Normalmente para este procedimiento se suelen implementar soluciones basadas en el promediado de periodogramas, sin embargo este método no es útil en algunos casos especialmente difíciles, donde aparecen anillos de *Thon* débiles y con mucho ruido. Para paliar los malos resultados del promedio de periodogramas algunos paquetes de rutinas realizan una simetrización radial de la *PSD*, pero esta técnica puede ocultar problemas leves de astigmatismo y no es útil si el astigmatismo aparece en mayor medida. *Xmipp*, en cambio, implementa una solución basada en modelos Auto regresivos (*ARMA*) para la estimación paramétrica de la *PSD*. Con este método se obtienen anillos de *Thon* claros y definidos además de que necesita menos muestras para una estimación de la *PSD* útil. Esto permite la implementación de estimaciones locales, lo que es interesante cuando se observan grandes cambios en la *CTF* en diferentes puntos de la micrografía. Basándose en modelos *ARMA*, tanto unidimensionales como bidimensionales, lo que permite tratar imágenes sin astigmatismo o con él, *Xmipp* obtiene una estimación 2D de la *CTF* cuya fase se corrige mediante el sencillo método de *phase flipping* [25] y [26].

Es conveniente realizar la corrección de fase en momentos tempranos del proceso, para evitar errores en los cálculos, sin embargo la corrección de módulo, si ésta es necesaria se puede hacer más adelante. *Xmipp* incorpora dos métodos para la corrección de módulo, *Iterative Data Refinement (IDR)* [26] y la incorporación directa de la *point spread function (PSF)* al algoritmo de reconstrucción. *IDR* no es más que un proceso iterativo en el que se van cambiando las proyecciones para que el módulo de la *CTF* afecte menos cada vez. Por otra parte la incorporación de la *PSF* al algoritmo de reconstrucción presenta problemas ya que en este tipo de reconstrucciones las ecuaciones de la *PSF* son bastante densas y perjudica el rendimiento del programa. Esto se evitó añadiendo al algoritmo de reconstrucción una versión de la *PSF* normalizada (y

no la propia *PSF*) mediante el algoritmo de [27].

- Asignación de los ángulos de proyección

Este paso sólo se necesita si no conocemos el ángulo desde el que fueron proyectadas las tomografías. Suele deberse al uso de una técnica específica comentada en la sección 2.3.3 para obtener rotaciones desde el máximo número posible de ángulos. Con esta técnica se saca una sola proyección general de una preparación con muchas repeticiones de la misma molécula. Con un número muy grande de moléculas es probable que éstas se coloquen fortuitamente en la preparación en ángulos muy distintos consiguiendo el resultado buscado. La única desventaja es que no conocemos esos ángulos, y por lo tanto tenemos que estimarlos. Este será el apartado de *Xmipp* en el que introducirá la nueva funcionalidad desarrollada para este proyecto, explicada en la sección 4.

Pero *Xmipp* ya posee una técnica para realizar la asignación de ángulos. La diferencia entre la técnica implementada y la que se ha desarrollado para este Proyecto Fin de Carrera es que la implementada en *Xmipp* se basa en la comparación de las proyecciones experimentales con otras que han sido obtenidas proyectando un volumen ideal de referencia. Por tanto, con la tecnología de *Xmipp* anterior a este proyecto era necesario disponer de una librería de proyecciones para la comparación. La mejora respecto a otros algoritmos es que el que *Xmipp* posee introduce técnicas de búsqueda mediante multiresolución y una selección angular novedosa.

La idea principal es la de calcular la correlación entre la proyección cuyos ángulos de proyección son desconocidos y el resto de proyecciones de la librería. Esta comparación se realiza de menor a mayor resolución utilizando la transformada *wavelet* discreta. Primero se compara la proyección principal con las demás a baja resolución y se selecciona un grupo de proyecciones de la librería para pasar a la siguiente fase, después se va subiendo su resolución en un procedimiento iterativo hasta alcanzar la calidad real de la tomografía bajo estudio. Este procedimiento es más rápido que la

búsqueda general y, al reducir la resolución, también es menos sensible a ruido. Además, como ya se ha indicado, la asignación de ángulos no se basa simplemente en el máximo de la correlación, como en otros algoritmos, sino que se utiliza la correlación de varias de las proyecciones de referencia que mejores resultado hayan dado.

Es evidente que la capacidad de este algoritmo depende del número de proyecciones de referencia. Dado que al aumentar el número de referencias se reduce la velocidad hay que llegar a un compromiso entre velocidad y precisión. Para poder apostar por la precisión sin comprometer excesivamente la velocidad *Xmipp* implementa un algoritmo que optimiza una función objetivo basada en los parámetros angulares y posicionales de la proyección experimental. Las imágenes de referencia se escogen observando la similitud entre un modelo a base de *Splines* de la proyección experimental y las de la librería, descartando las menos parecidas y dejando solo aquellas que entrarán en el algoritmo iterativo de multiresolución explicado antes.

Una vez asignado los ángulos se puede realizar una reconstrucción inicial del volumen, y después se pueden mejorar la estimación de los ángulos de proyección iterando el proceso asignación / reconstrucción hasta obtener la calidad deseada.

- Reconstrucción

Xmipp utiliza el método de *expansión en series* para la reconstrucción. Se trata de una técnica en la que se crea el volumen final ajustando una serie de pesos en un proceso iterativo, cada uno de los cuales controla el valor de una función de referencia desplazada un número indicado de veces en el espacio de Fourier. Las funciones de referencia utilizadas por *Xmipp* son las *Blobs*, nombre común para la ventana de *Kaiser-Bessel*. Fue propuesta como sustitución de los *Voxels* en [28] y [29], y se ha convertido en una de las más utilizadas en este campo debido a los buenos resultados que obtiene. Una de sus características más atractivas, debido a los altos niveles de ruido del proceso es que es un función que varía suavemente tanto en el espacio real como en Fourier.

Para la realización del proceso de reconstrucción iterativo *Xmipp* implementa varias soluciones: la *técnica de Reconstrucción algebraica* [30], la *técnica de reconstrucción iterativa simultánea* [31], la *técnica de Reconstrucción algebraica simultánea* [32], *promedio de componentes* [33], *promedio de componentes iterativo por bloques* y [34], *promedio de cadenas* [35].

ART es la mejor técnica para la mayoría de los casos como muestran algunos estudios como [36] y [37].

3.2.4 Elementos utilizados en el proyecto

En este punto se explicarán los elementos de las librerías de *C++* de *Xmipp* que se han utilizado en el proyecto. La amplia variedad de elementos, tanto de matemática general, como de procedimientos específicos de la reconstrucción de volúmenes permite la implementación de algoritmos con mayor eficacia ya que permite un acercamiento matemático a la programación en *C++*.

Una buena muestra de ello son los elementos como *Matrix1D* (vectores) y *Matrix2D* (matrices), que nos permiten ejecutar fácilmente operaciones como la multiplicación de matrices, la multiplicación de una matriz por un vector, el producto vectorial y el escalar o la rotación matricial.

Como elementos matemáticos imprescindibles podemos destacar el tratamiento de los ángulos de Euler y *Differential Evolution*. *Xmipp* posee una función para recuperar la matriz de rotación a partir de unos ángulos de Euler, del modo en que se verá en la sección 3.3.1 Ángulos de Euler. Por otra parte *Xmipp* contiene una implementación de *Differential Evolution* en forma de clase de *C++* que se encarga de gestionar los individuos de cada generación y utilizarlos en la función objetivo que se le indique, programada específicamente para el problema a resolver.

El último grupo de elementos utilizados son los que facilitan la lectura, escritura y visionado de datos. Por una parte Xmipp posee clases para gestionar los ficheros *sel* (ficheros propios de agrupación de imágenes) y las imágenes, con la clase *ImageXmipp*, a las que se les pueden asignar los ángulos de Euler y otros parámetros. Para el visionado de datos Xmipp posee elementos gráficos con los que se pueden mostrar tanto vectores, con la ayuda del programa libre *gnuplot*, como tomografías, con un visor propio basado en la librería *qt3*.

3.3 Herramientas matemáticas

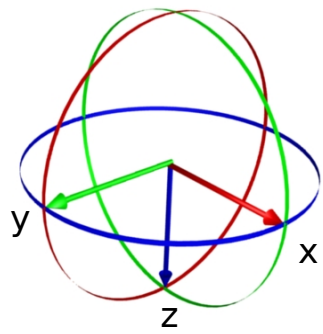
3.3.1 Ángulos de Euler

Existen varias maneras de describir la orientación de un cuerpo en el espacio. Uno de ellos son las matrices homogéneas de transformación 4x4, que se usan, por ejemplo en robótica y tienen la ventaja de representar rotaciones y traslaciones en una sola matriz. Otro de los más utilizados, presentes en la computación de gráficos por ordenador son los cuaterniones, que son una extensión de los números reales, similar a los números complejos, pero en vez de incluir la unidad imaginaria, tal que $i^2 = -1$, incluyen tres unidades tal que $i^2 = j^2 = k^2 = ijk = -1$, y como todo conjunto numérico poseen operaciones aritméticas adecuadas para su manipulación. Sin embargo, pese a que estas dos representaciones de la rotación sean de las más utilizadas debido a su versatilidad no utilizan el menor número parámetros ya que, como se ha demostrado en numerosas ocasiones, [38] y [39], cualquier rotación en el espacio de un cuerpo puede ser descrita mediante sólo tres ángulos.

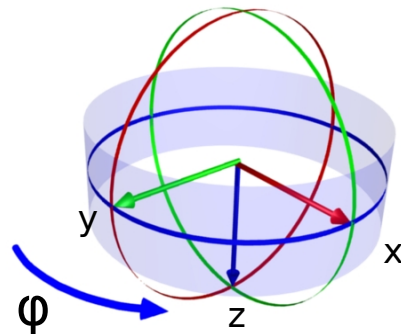
Cuando se utilizan tres ángulos como representación de la orientación de un sólido en el espacio es conveniente abstraerse de la forma del cuerpo y tener en cuenta que la posición inicial puede describirse con un sistema de coordenadas A de tres ejes que se mantendrán estáticos y que la rotación no es más que otro conjunto de ejes B rotado con respecto al original y con el mismo centro. Siendo esto así, cada uno de los

tres ángulos de Euler que generan esta rotación supone un giro sobre alguno de los ejes descritos, tanto del sistema A como del B. Esto implica que las rotaciones también pueden realizarse sobre los ejes ya girados. Tanto A como B poseen tres ejes XA, YA, ZA, XB, YB, ZB sobre los que es posible realizar las rotaciones y, dado que la única restricción es que el segundo eje no sea igual al primero ni al tercero, se nos presentan 24 posibilidades, doce para las rotaciones sobre los ejes estáticos de A (XYZ, XYX, YZX, YZY, ZXY, ZXZ, XZY, XZX, YXZ, YXY, ZYX, ZYZ) y otras doce sobre los ejes móviles de B. Las tres más usadas son XYZ, ZXZ y ZYZ, siendo esta última, sobre ejes móviles, la que utiliza Xmipp. Se debe tener en cuenta que, si definimos los tres ángulos de Euler como (α , β , φ) las rotaciones sobre ejes estáticos utilizan los ángulos en orden, pero si lo hacemos sobre ejes móviles el orden se invierte y las rotaciones se dan primero por φ , luego por β y por último por α .

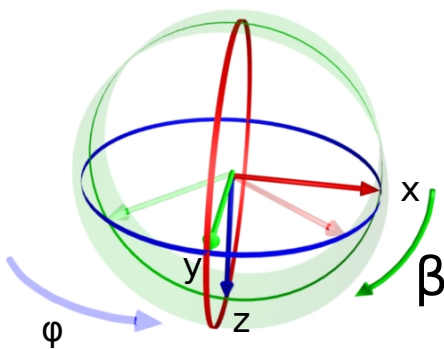
Como apunte de la notación se debe aclarar que en aeronáutica se utilizan los ángulos Cardanos, denominados Yaw, Pitch y Roll. A veces estos ángulos son denominados erróneamente de Euler, sin embargo no son más que una concreción de la problemática expuesta en este punto, en la que las rotaciones se realizan sobre los ejes estáticos sin que ninguno se repita, es decir, las rotaciones pueden ser sobre XYZ, ZYX, YZX, etc.



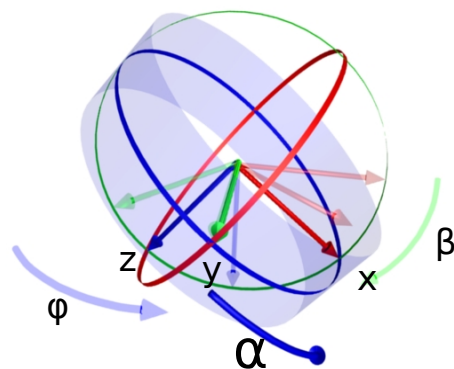
Paso 1: Posición inicial



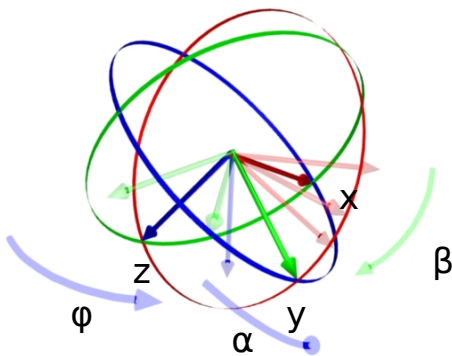
Paso 2: Previo a primera rotación en Z



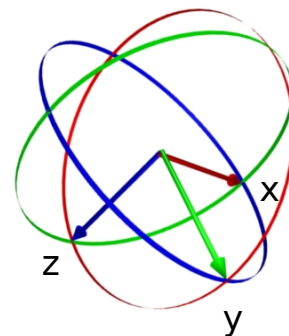
Paso 3: Primera rotación en Z previo a rotación en nuevo Y



Paso 4: Primera rotación en Y previo a segunda rotación en nuevo Z



Paso 5: Segunda rotación en nuevo Z



Paso 6: Resultado de la rotación

Ilustración 4: Proceso gráfico de la rotación por ángulos de Euler

Ahora nos centraremos en el caso de las rotaciones en $Xmipp$, que se definen mediante tres ángulos (α , β , φ) sobre ejes móviles en el orden ZYZ. Para ilustrar este

tipo de rotación nos apoyaremos en la Ilustración 4. Partimos del primer paso en el que, siguiendo la notación de *Xmipp*, el sistema de coordenadas se orienta con el eje Z mirando hacia abajo. Entonces, en el paso 2, se efectúa una rotación de ángulo φ sobre el eje Z (nótese que éste ángulo no es el primero de los tres, sino el último, debido al uso de rotaciones sobre ejes móviles, como ya se ha explicado).

Tras la primera rotación seleccionamos el eje Y ya girado y rotamos el cuerpo respecto a él β grados en el paso 3. El último giro se efectúa sobre el eje Z ya girado, en el paso 4 giramos el cuerpo respecto a ese eje α grados. En el paso 5 se pueden ver todas las rotaciones en un sólo gráfico. Para interpretarlo debemos seguir primero el movimiento del eje Y en verde en el giro de φ grados, después el movimiento del eje X en rojo en el giro de β grados y por último, en el giro de α grados seguir el movimiento del eje Y. En el paso 6 se pueden ver todos los ejes tras la rotación de Euler.

Aritméticamente esta rotación se puede ver como una multiplicación de cada uno de los puntos del cuerpo por una matriz que se puede obtener a través de los ángulos de Euler definidos y el tipo de rotación establecido. Sabiendo el orden de las rotaciones sobre cada eje cada posición de la matriz se puede obtener en función de los ángulos de Euler mediante la multiplicación de las tres matrices que representan cada rotación individual, es decir $M = R_\alpha \cdot R_\beta \cdot R_\varphi$, donde:

$$R_\alpha = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad R_\beta = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix}$$

$$R_\varphi = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Si definimos:

$$c_\alpha = \cos(\alpha); \quad c_\beta = \cos(\beta); \quad c_\varphi = \cos(\varphi)$$

$$s_\alpha = \sin(\alpha); \quad s_\beta = \sin(\beta); \quad s_\varphi = \sin(\varphi)$$

$$CC = C_\beta \cdot C_\alpha; \quad CS = C_\beta \cdot S_\alpha; \quad SC = S_\beta \cdot C_\alpha; \quad SS = S_\beta \cdot S_\alpha$$

Entonces la matriz de rotación será:

$$M = \begin{pmatrix} CC \cdot C_\varphi - S_\varphi \cdot S_\alpha & -CC \cdot S_\varphi - S_\alpha \cdot C_\varphi & SC \\ CS \cdot C_\varphi + C_\alpha \cdot S_\varphi & -CS \cdot S_\varphi + C_\alpha \cdot C_\varphi & SS \\ -S_\beta \cdot C_\varphi & S_\beta \cdot S_\varphi & C_\beta \end{pmatrix}$$

Para rotar cualquier punto del objeto bastará poner su valor , que antes estaba en las coordenadas A(x,y,z) en las coordenadas resultado de multiplicar el vector A por la matriz M, es decir:

$$(x', y', z') = M \cdot (x, y, z)$$

3.3.2 Teorema de la Sección Central

Este teorema asegura que la transformada de Fourier 2D de la proyección de un volumen es un corte o sección que pasa por el centro de la transformada de Fourier 3D del objeto tridimensional. Además esta sección se orienta de tal manera que la dirección de la proyección es la normal del plano que la contiene. Debido a que las diferentes secciones de la transformada de Fourier obtenidas pasan por el centro y en su orientación solo se diferencian por el vector normal es evidente que cualquier pareja de secciones centrales de un mismo objeto tendrán una línea en común, puesto que es imposible colocar dos planos en el espacio de diferente orientación y que no sean paralelos sin que éstos compartan una línea. Dado que las normales de estos planos comparten un punto del espacio, la línea común de los dos planos contendrá también ese punto, que pasa por el centro de ambos planos.

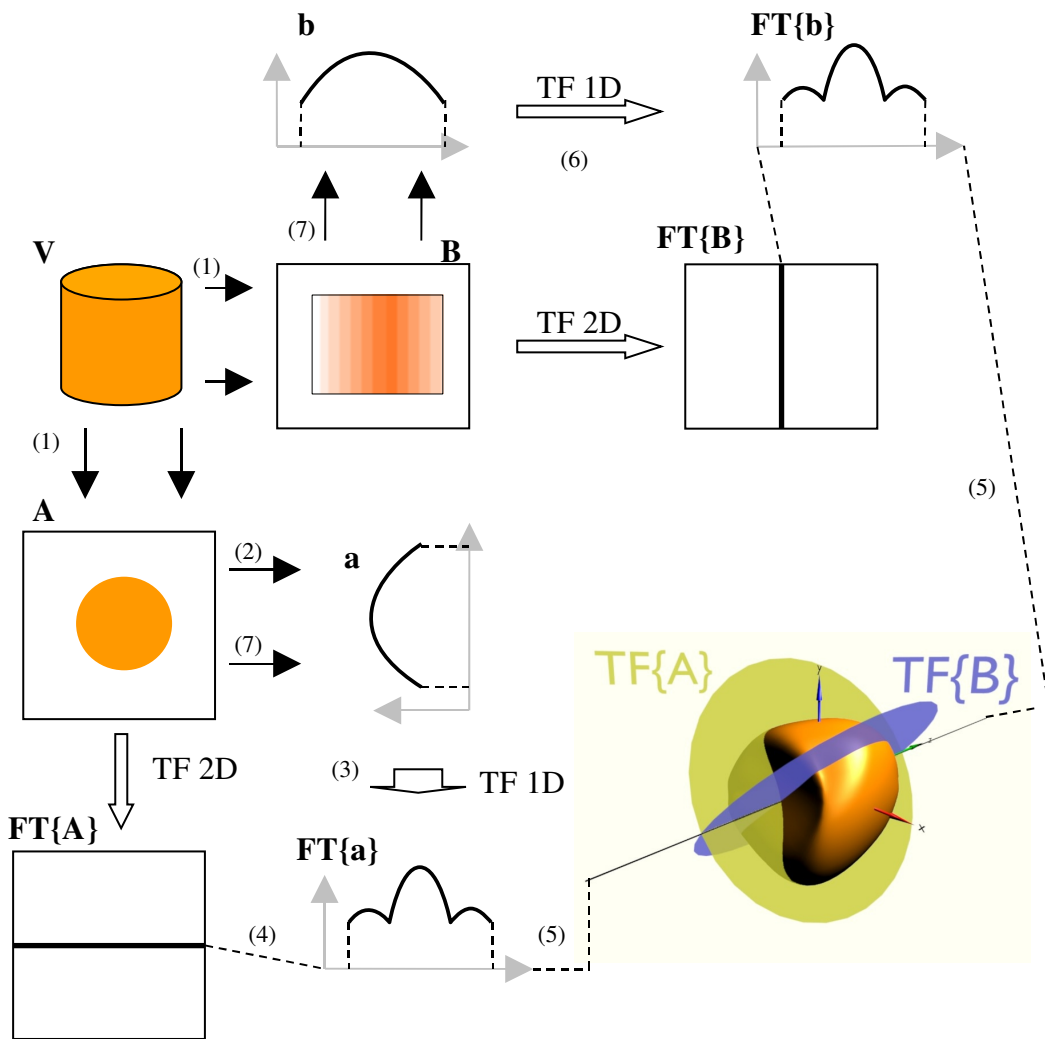


Ilustración 5: Líneas Comunes

Si sabemos que dos secciones tridimensionales centrales compartan forzosamente una línea y además tenemos en cuenta el “Teorema de la Línea Central” podemos concluir que dos proyecciones bidimensionales de un mismo volumen siempre tendrán una de sus proyecciones 1D en común. Dado que este teorema es esencial para el proyecto que nos ocupa, lo estudiaremos detenidamente, basándonos en la Ilustración 5. El proceso es el siguiente: Dadas dos proyecciones bidimensionales A y B obtenidas de un mismo volumen V (1), obtenemos una proyección 1D de A llamada a (2) y su transformada de Fourier $TF\{a\}$ (3). Por el “Teorema de la Línea Central” sabemos que $TF\{a\}$ es una línea de la Transformada de Fourier 2D de A (4) que, en este caso y con

fines didácticos, es justo la línea compartida por la transformada de Fourier 2D de \mathbf{B} que impone el Teorema de la Sección Central (5). Pues bien, esta línea compartida por ambas Transformadas, $\mathbf{TF}\{\mathbf{a}\}$ es además $\mathbf{TF}\{\mathbf{b}\}$, cuya transformada inversa es \mathbf{b} (6). Por tanto, \mathbf{b} (o \mathbf{a} , porque son iguales) es la proyección 1D que comparten las dos proyecciones 2D del objeto tridimensional (7).

3.3.3 Teorema de la Proyección Común

Para el caso del cálculo de ángulos que nos ocupa el Teorema de la Sección Central se puede extender sin que deje de ser válido formulándolo con una dimensión menos, con lo que el “Teorema de la Línea Central” impondría que la transformada de Fourier 1D de la proyección de una imagen en gama de grises (que se puede ver como una imagen de densidad) es una línea de la transformada de Fourier 2D de ese plano. Además, el teorema de la sección central se puede expresar también en el espacio real, con lo que obtendríamos el “Teorema de la Proyección Común”. Este teorema indica que dos proyecciones 2D de un mismo objeto comparten una proyección 1D.

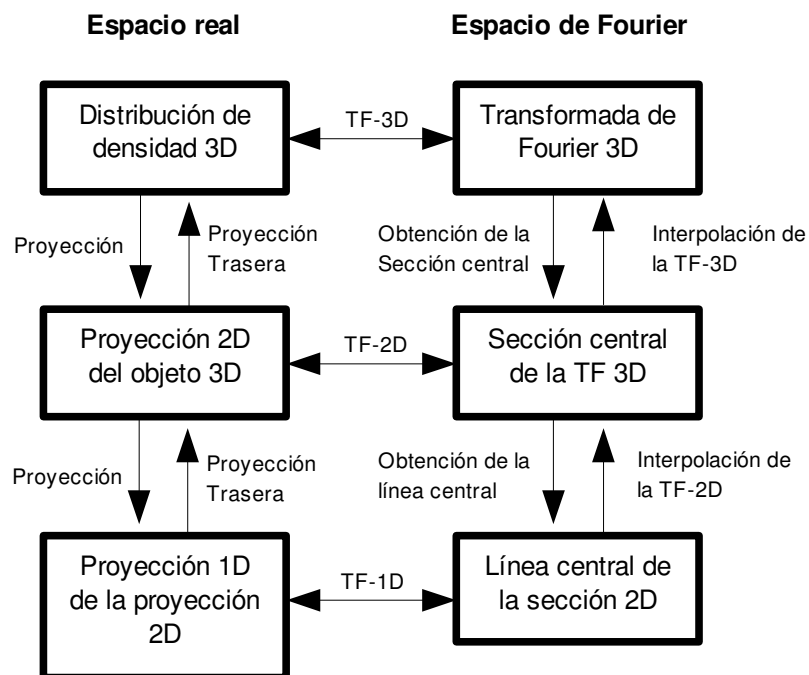


Ilustración 6: Teorema sección central: Correspondencia de espacios

En la Ilustración 6 se puede observar la correspondencia entre el espacio real y el de Fourier. Como se ha explicado una proyección bidimensional del volumen en el espacio real se corresponde con una sección que pasa por el centro de la transformada de Fourier del volumen. Así mismo se puede disminuir en una dimensión el enunciado del teorema sin que éste pierda validez, por tanto, la proyección 1D de una proyección bidimensional tiene una transformada de Fourier que es una línea de la transformada de Fourier 2D de la proyección bidimensional.

3.3.4 Técnicas de minimización

Para la resolución del algoritmo que se presentará en el capítulo 4 es necesario disponer de un algoritmo de minimización con la capacidad de obtener el mínimo valor de una función multidimensional cuya derivada no está disponible como las que resultarán del proceso de reconstrucción angular. Muchos de los problemas encontrados en entornos científicos, como el que nos ocupa, se reducen a sistemas cuya entrada es n -

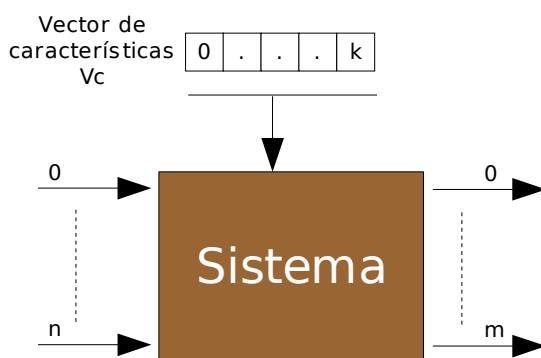


Ilustración 7: Sistema General

dimensional y su salida m -dimensional, siendo n y m enteros sin relación. Estos sistemas tratan, pues, de obtener la salida correcta para cualquier entrada ajustando una serie de características internas del sistema. Esas características suelen representarse, por conveniencia, en forma de vector (Ilustración 7). Para obtener los valores de ese vector de características se suele recurrir a funciones que los seleccionan minimizando una función objetivo, normalmente la que calcula el error entre la salida obtenida y la esperada. La función objetivo, por tanto, es una función continua con las características del sistema como entrada y un vector de valores reales como salida, que después se reducirá a un solo valor real mediante el criterio de suma de errores o de error máximo. Este valor único dará una idea de la bondad del vector de

características elegido (Ilustración 8). A menor valor de salida mejor es la elección de los parámetros de entrada. Por tanto, los algoritmos de minimización deben ser capaces de realizar la optimización global de una función continua multivariable.

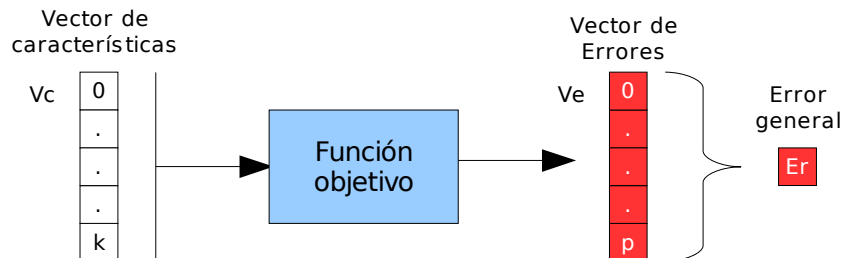


Ilustración 8: Función objetivo

En el caso concreto en el que la función objetivo es no lineal y sus derivadas no sean accesibles, es decir, que dado un valor no tenemos información sobre valores colindantes, no podemos basarnos en técnicas de gradientes para mejorar el algoritmo de minimización y por tanto debemos usar técnicas de búsqueda directa. Obviaremos la técnica de fuerza bruta puesto que nuestro problema tendrá múltiples variables y por tanto la computación del mínimo sería demasiado costosa en términos de capacidad de procesamiento.

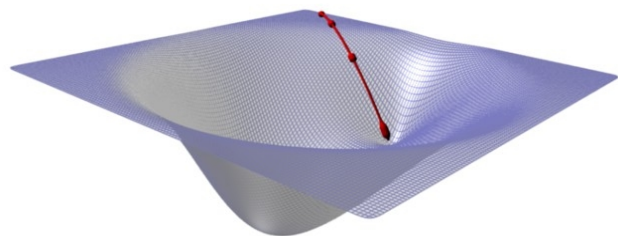


Ilustración 9: Greedy Criterion problema de los mínimos locales

Todos los algoritmos de minimización por búsqueda directa siguen el mismo esquema general. En principio crean posibles candidatos para el vector de características mediante diferentes criterios de selección. Entonces generan un nuevo vector de características a partir del anterior y lo seleccionan o no como nuevo candidato basándose en el error obtenido de la función objetivo. Los algoritmos que se explicarán aquí y que son los que suelen converger en el mínimo más rápido son aquellos que sólo

aceptan un nuevo candidato si éste reduce el valor de la función objetivo. Es el llamado *Greedy Criterion*, cuya traducción se puede aproximar a *criterio voraz*. El criterio voraz tiene un problema intrínseco y es que como la línea de candidatos siempre decrece corre el riesgo de entrar en un mínimo local, estropeando todo el proceso de minimización (Ilustración 9).

Para solucionar el problema de los mínimos locales surgieron los algoritmos genéticos, que generan varias líneas de candidatos, y después los algoritmos evolutivos como versión continua de los anteriores. Como se puede ver en la Ilustración 10 al usar varias líneas de candidatos, pese a que algunas caigan en mínimos locales (líneas en rojo) si se usa un número suficiente de candidatos al menos uno alcanzará el mínimo global (línea verde). Otro método que es capaz de salvar los mínimos locales es *Simulated Annealing* (Ilustración 11). Lo consigue permitiendo pequeños movimientos “cuesta arriba” de las líneas de candidatos, cuantas más iteraciones se han ejecutado menos subidas se permiten llegando al punto de prohibirlas completamente como en el criterio voraz. Ambos criterios se pueden utilizar para todos los algoritmos de minimización, aunque en los evolutivos suele utilizarse el criterio voraz, puesto que el problema de los mínimos locales ya ha sido solucionado.

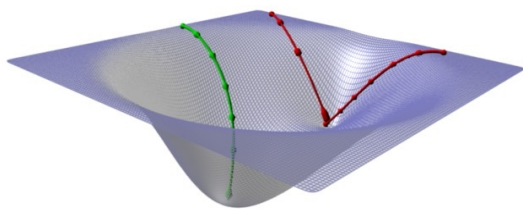


Ilustración 10: Algoritmo evolutivo

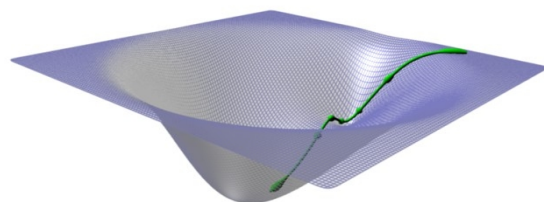


Ilustración 11: Simulated Annealing

A continuación se muestra una ilustración que resume las técnicas más conocidas para la minimización de funciones. En él se pueden ver desde las técnicas de minimización locales como el Método de Powell, hasta las técnicas más modernas basadas en poblaciones, como la Evolución Diferencial. Los algoritmos mencionados se desarrollarán más adelante ya que formarán parte del proceso desarrollado en este

proyecto. Tras la ilustración se indican referencias al resto de algoritmos por si el lector desea ampliar su conocimiento sobre alguno de ellos.

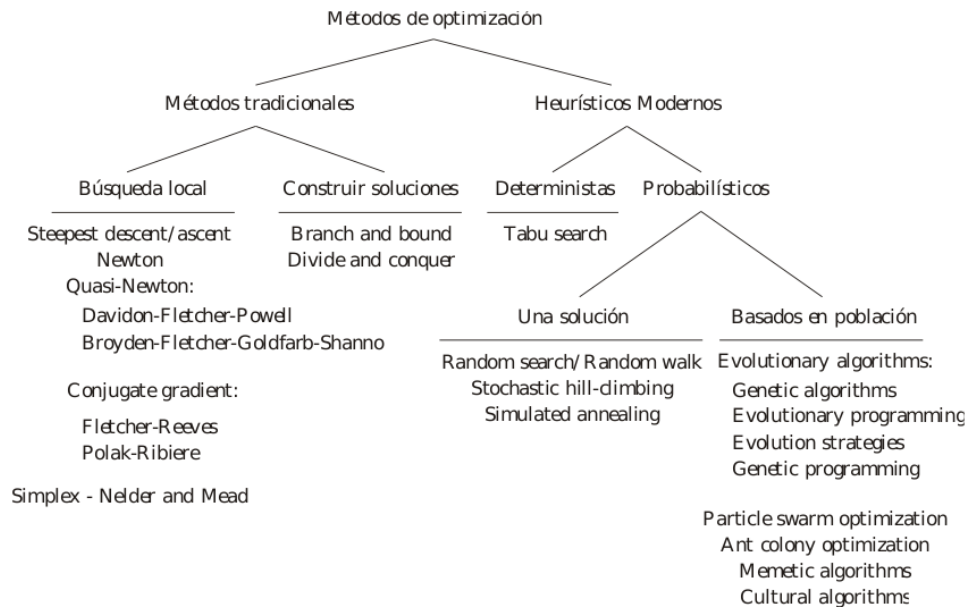


Ilustración 12: Clasificación de los métodos de optimización más relevantes.

Extraído de [40]

- Métodos de búsqueda local [41]
- Simplex – Nelder and Mead [42]
- Métodos basados en construir soluciones [43]
- Búsqueda Tabú [44]
- Simulated annealing [45]
- Algoritmos Genéticos [46]
- Particle swarm optimization [47]

- Método de Powell

El método de Powell es un método de minimización multidimensional de los denominados *single-shot*, que se basa en una serie de minimizaciones en una dimensión a lo largo de un grupo de direcciones representadas por vectores unitarios que

comienzan en un punto predefinido. El algoritmo se mueve a lo largo de la primera dirección establecida utilizando técnicas de minimización para una dimensión, después minimiza a lo largo de la segunda dirección y continúa con las demás. Después repite el proceso de forma cíclica, tantas veces como sea necesario hasta que la función objetivo deje de disminuir.

En los algoritmos *single-shot* la selección de las direcciones del algoritmo se basa en una exploración rápida de las cercanías del punto elegido para dirigir la minimización a la zona que parezca reducir más la función objetivo. Esto es un arma de doble filo ya que los resultados dependen mucho del valor inicial en el que se sitúe el algoritmo y por tanto debe contarse con una idea más o menos correcta de la zona en la que se sitúa el mínimo global.

El método de Powell también establece una serie de reglas para actualizar el grupo de direcciones para evitar que entremos en un bucle infinito de minimización en las direcciones iniciales, puesto que la minimización en una dirección puede estropear las anteriores y además se consigue aumentar la variedad de direcciones por la que se mueve el algoritmo a lo largo del tiempo. La regla principal es que mientras nos movemos por la dirección actual el gradiente de la función objetivo debe permanecer perpendicular respecto a la dirección anterior. Si esta condición se cumple para cada pareja del grupo de direcciones se dice que es un grupo conjugado. No es necesario, por tanto, rehacer ninguna de esas direcciones en sucesivas minimizaciones lineales de la función objetivo, porque la minimización en una dirección no estropea las anteriores.

Una de las desventajas de los algoritmos *single-shot*, entre ellos el de Powell es que pese a su eficacia a la hora de disminuir la función objetivo es muy dado a quedar atrapado en mínimos locales si el mencionado valor inicial está más cerca de uno de ellos que del mínimo global y sobre todo si estos mínimos locales son pronunciados, ya que al seleccionar las direcciones de minimización el algoritmo tenderá al mínimo local más cercano. Sucesivamente las siguientes direcciones volverán a dirigirse al centro del mínimo en vez de salir de él para encontrar el mínimo global.

Estas características hacen al método de Powell un complemento ideal para nuestro proceso, ya que, como se verá la Evolución Diferencial es capaz de encontrar la zona del mínimo local y acercarse a éste lo suficiente en casi todos los casos, sin embargo, en ciertas ocasiones con consigue centrarse del todo en el mínimo absoluto dentro de la zona del mínimo global, por lo que aplicando el algoritmo de Powell tras el de la Evolución Diferencial nos aseguramos que estemos en el mínimo que estemos nos dirigiremos a su mínimo absoluto.

- Evolución Diferencial (*Differential Evolution*)

Cuando uno se enfrenta a un problema tan complejo como los que se han descrito en la introducción, con funciones objetivo multivariable no lineales cuyas derivadas no están disponibles y millones de posibles candidatos para el vector de características lo que se espera de un algoritmo de minimización, por orden de prioridad, es que tenga como mínimo las siguientes características:

- Que sea capaz de encontrar el mínimo global, independientemente de los parámetros utilizados en el algoritmo. Característica indispensable puesto que es el objetivo final. Si se queda atrapado en mínimos locales nuestro sistema final no será eficiente y si esto depende de los parámetros de control del algoritmo, que se introducen según un criterio personal tenemos la dificultad añadida de no saber con certeza como se comportará el algoritmo.
- Que la convergencia sea rápida. El tiempo es una de las variables más importantes en cualquier experimento. Cuanto más rápido, mejor.
- Que tenga pocos parámetros de control. Demasiados parámetros de control con funciones poco documentadas o poco comprensibles implican complejidad añadida.

Con estas necesidades en mente, se eligió la Evolución Diferencial porque las cumple con creces. Este algoritmo ha sido probado en múltiples experimentos

obteniendo resultados siempre favorables y su funcionamiento es sencillo y comprensible, por lo que además de ser fácilmente implementable, cuando se utiliza siempre se tiene sensación de control sobre el proceso de minimización.

El problema de minimización general que se plantea en el algoritmo se puede expresar en términos matemáticos de la siguiente manera. Tendremos dos vectores de valores reales como elementos del problema. El primero contendrá los valores de las propiedades del sistema que habrá que minimizar manteniéndose dentro de unos límites (V_{min}), el segundo estará formado por aquellas características que no hace falta minimizar, pero que tendrán que mantenerse dentro de unos límites (V_{cons}). Ambos vectores se pueden combinar en uno solo para dar como resultado el vector de características (V_c).

$$\begin{aligned} & \vec{V}_{min}_i \quad i=0,1,2,\dots,k-1 \\ & \vec{V}_{cons}_j \quad j=0,1,2,\dots,l-1 \\ \vec{V}_c = & [\vec{V}_{min} \quad \vec{V}_{cons}] \quad p=0,1,2,\dots,k+l-1 \end{aligned}$$

Donde V_{cons} puede no contener ningún valor dependiendo de la nomenclatura del problema. Estas características estarán relacionadas con un tercer vector de parámetros pertenecientes al conjunto de los números reales.

$$\vec{x}_m \quad m=0,1,2,\dots,n-1 \quad \Rightarrow \quad \vec{x}_m \in [x_{min} \quad x_{max}] \quad (2)$$

En general, el proceso de optimización consistirá en variar el vector \vec{x}_m hasta que las propiedades V_{min} estén optimizadas y las características no optimizables estén dentro de los límites establecidos. Esta optimización se puede ver como una función cuya salida debe asemejarse a unos ciertos valores de V_c óptimos, por lo que siempre se puede reescribir como una minimización:

$$\min f_p(\vec{x}) \quad (3)$$

Donde $f_p(\vec{X})$ es un función en cuyo interior se calculan todos los valores p de Vc a partir de los parámetros x y se devuelve un valor a minimizar (normalmente la diferencia entre el valor óptimo de Vcp y el calculado). Las p funciones $f_p(\vec{X})$ se pueden combinar en una sola ($H(\vec{X})$) siguiendo dos posibles criterios:

- La suma ponderada:

$$H(\vec{X}) = \sum_{m=0}^{k+l-1} w_p \cdot f_p(\vec{X}) \quad (4)$$

- Criterio minimax (minimización del máximo):

$$H(\vec{X}) = \max(w_p \cdot f_p(\vec{X})) \quad (5)$$

Donde $w_p > 0$ y representa el vector de pesos que sirve para dar más importancia a unas características y menos a otras. Por tanto la minimización puede reescribirse como:

$$\min H(\vec{X}) \quad (6)$$

El criterio minimax garantiza que todos los mínimos locales, incluyendo mínimos globales múltiples pueden ser encontrados, al menos teóricamente, [48] y [49]. Para la función objetivo (4) esto solo es verdad si la región en la existe \vec{X} es convexa, [50], lo que no se aplica en la mayoría de los problemas técnicos.

El algoritmo de Evolución Diferencial (DE) utiliza NP vectores de parámetros como población de cada generación G.

$$\vec{x}_{i,G} \quad i=0,1,\dots, NP-1$$

Cada vector de parámetros \vec{X} es un candidato a mínimo de la función. NP no variará durante todo el proceso. La población de la primera generación depende de los

conocimientos *a priori* que se tengan sobre los parámetros, si no tenemos ninguna información lo normal es escogerlos de forma aleatoria con una función de probabilidad uniforme. Si lo que se desea es refinar un resultado previo se pueden escoger los valores aleatorios con una función de probabilidad normal, centrada en el valor predefinido y con una desviación estándar que depende de la fiabilidad del resultado previo.

Dado que DE es un algoritmo paralelo de minimización global directa la clave de su desarrollo está en la elección de los candidatos para cada generación y el criterio que verifica si el nuevo candidato es mejor que el anterior. La elección de nuevos vectores candidatos es sencilla y elegante, DE genera el nuevo vector de parámetros añadiendo la diferencia ponderada de dos vectores a un tercero. La decisión entre este nuevo candidato y el anterior, que puede pero no tiene por qué ser uno de los tres que forman parte del proceso anterior, es aun más sencilla. De entre los dos el que continuará en la siguiente generación será el que menor valor genere en la función objetivo. De esta manera tenemos un criterio de selección sencillo y un proceso de generación de candidatos que han evolucionado a partir de los existentes. Por último, se evalúa el mejor candidato de cada generación para tener una idea de la evolución del proceso de selección.

La idea de extraer información sobre la dirección y distancia de la población para generar mutaciones aleatorias da como resultado un proceso adaptativo con una gran capacidad de convergencia. Hay varias variantes del esquema de DE, aquí se explicarán dos de las que mejores resultados han obtenido, siendo las diferencias entre ambos bastante pequeñas:

- Esquema DE1

Con este esquema para cada vector $\vec{x}_{i,G}$ $i=0,1,\dots, NP-1$, se genera un vector de prueba de la forma:

$$\vec{V} = \vec{x}_{r_1,G} + F \cdot (\vec{x}_{r_2,G} - \vec{x}_{r_3,G})$$

donde $r_1, r_2, r_3 \in [0, NP-1]$ y son enteros diferentes entre si y diferentes al índice actual (elemento con el que se comparará el vector. Serán elegidos aleatoriamente en ese intervalo. Además $F > 0$ y es un valor real y constante que controla la importancia del factor de diferencia.

- Esquema DE2

En este caso el vector de prueba que se genera corresponde a la siguiente fórmula:

$$\vec{V} = \vec{x}_{r_1, G} + \lambda \cdot (\vec{x}_{best, G} - \vec{x}_{i, G}) + F \cdot (\vec{x}_{r_2, G} - \vec{x}_{r_3, G})$$

donde se introduce el factor λ , que servirá para controlar el nuevo valor incorporado, $\vec{x}_{best, G}$, que es el vector que mejor resultado ha dado en la minimización hasta ahora. Esta función puede ser útil acelerando la búsqueda en aquellas funciones objetivo donde el mínimo global es fácil de encontrar.

3.3.5 Transformada de Radon

La transformada de Radon 2D discreta (RT) consiste en un conjunto de proyecciones 1D de una función $f(x,y)$ realizadas desde unos ángulos determinados. El punto central de la proyección realizada desde el ángulo Θ se calcula como la integral de $f(x,y)$ a través de la línea A que pasa por el punto (0,0) y tiene ángulo $\Theta+90^\circ$. El cálculo del resto de puntos en sentido positivo y negativo consiste en ir integrando $f(x,y)$ a través de las líneas paralelas a la inicial desplazadas distancias discretas directamente proporcionales a la posición de los puntos que representan.

Una forma más intuitiva de entender este cálculo sería verlo con una rotación de los ejes x e y en los nuevos ejes x' e y' que resultan de rotar los primeros un ángulo Θ . Entonces cada valor p de la proyección en Θ se corresponderían con la integral de $f(x,y)$ a lo largo del eje y' en la posición $x' = p$.

Los valores de las integrales resultantes de este proceso se colocan en línea según el punto desde el que fueron realizadas dando lugar a las proyecciones desde distintos ángulos. Para obtener la transformada de Radon se colocan estas proyecciones en paralelo desde el grado 0 hasta el 180 en intervalos discretos (normalmente de 1°).

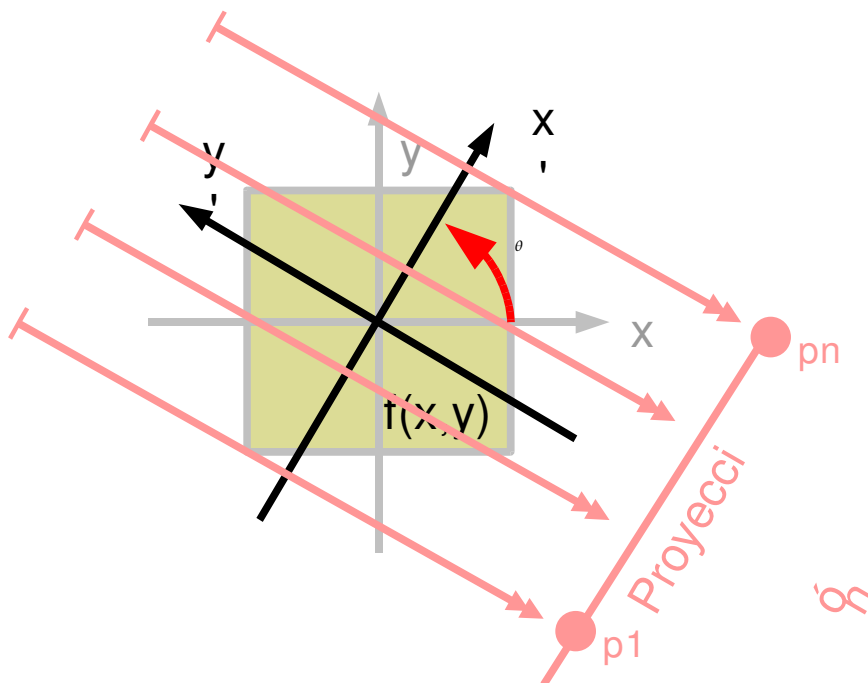


Ilustración 13: Transformada Radon por rotación

De forma que matemáticamente podemos definir la proyección relacionada con la transformada Radon para un ángulo Θ como:

$$R_{\theta}(x') = \int_{-\infty}^{\infty} f(x' \cos \theta - y' \sin \theta, x' \sin \theta + y' \cos \theta) dy'$$

Dada esta definición se nos presenta un método sencillo para calcular la transformada Radon mediante programación. Este consiste en rotar la imagen $-\Theta$ (utilizando una matriz de rotación sencilla de calcular) y luego sumar las columnas de la matriz que representa la imagen. Si repetimos este paso para $\theta \in [0, 179]$ (obsérvese que es un rango discreto) y colocamos las proyecciones como columnas de una nueva matriz habremos obtenido la transformada Radon de la imagen en intervalos de un grado. El rango de una transformada Radon solo llega a los 179° porque la proyección de 180° grados y sucesivos es equivalente a la proyección de 0° y sucesivos respectivamente (o en todo caso simétrica respecto del centro).

4 Desarrollo

4.1 Problemática

Nuestro problema se centra en la asignación de ángulos tridimensionales (ver punto 2.6) para el proceso de reconstrucción tridimensional a partir de proyecciones. El mayor problema que se presenta es la gran cantidad de grados de libertad que posee la solución, debido a que se han de calcular los tres ángulos de Euler para cada proyección introducida, en un rango de 360° para dos de ellos y 180° para el otro en un espacio continuo. Cuantas más proyecciones se utilicen en la reconstrucción mayor número de grados de libertad habrá, pero si no se usan las suficientes la calidad de la reconstrucción posterior disminuirá. Este compromiso será decisivo a la hora de diseñar el algoritmo de asignación.

Hay que tener en cuenta, además, que se nos presenta como entrada un grupo de imágenes, que no son más que proyecciones planas de un volumen. Si este volumen posee algún tipo de simetría las proyecciones realizadas desde varios ángulos serán idénticas, imposibilitando su discriminación. Esto, que en principio se presenta como un grave problema, se puede solventar si conocemos *a priori* el tipo de simetría del cuerpo tridimensional. Se suele tener conocimiento de este dato antes de entrar en el proceso de reconstrucción con lo que, pese a ser una información adicional, no requiere que el usuario pierda tiempo obteniéndolo.

Un problema de simetría adicional es el que se produce al girar el ángulo de proyección 180° en el eje y en el sistema de referencia local. Esto hace que la normal de la proyección cambie de sentido, lo que produce una proyección reflejada respecto de la inicial. Por ello, si buscamos la proyección 1D común de ésta proyección respecto a otra cualquiera, puede que el algoritmo estime que la proyección que más se parece es la de la tomografía reflejada, asignando unos ángulos que, en teoría son correctos, pero que no se corresponden con los iniciales. Incluso existe la posibilidad de que dos de las

tomografías se coloquen en las posiciones reflejadas, impidiendo que la tercera se posicione correctamente.

Dada la naturaleza del problema que se nos presenta solo nos es posible resolver la relación entre los ángulos de las proyecciones y no los ángulos reales desde los que fueron tomadas. Para que la reconstrucción tenga éxito es necesario disponer de una referencia en el espacio tridimensional, que una vez establecida servirá para la colocación del resto de tomografías según la relación entre sus respectivos ángulos. El problema es que para establecer una referencia ésta ha de tener sus ángulos fijados, pero dado que ese es precisamente el propósito de este algoritmo, no disponemos de esa información. *A priori* no se puede contar con los ángulos reales, sin embargo es posible utilizar una de las proyecciones de entrada como referencia si se le asignan unos ángulos cualquiera y se mantienen fijos durante la ejecución del algoritmo. Por simplicidad los tres ángulos que forman el ángulo de Euler de la tomografía de referencia se establecerán con un valor igual a cero.

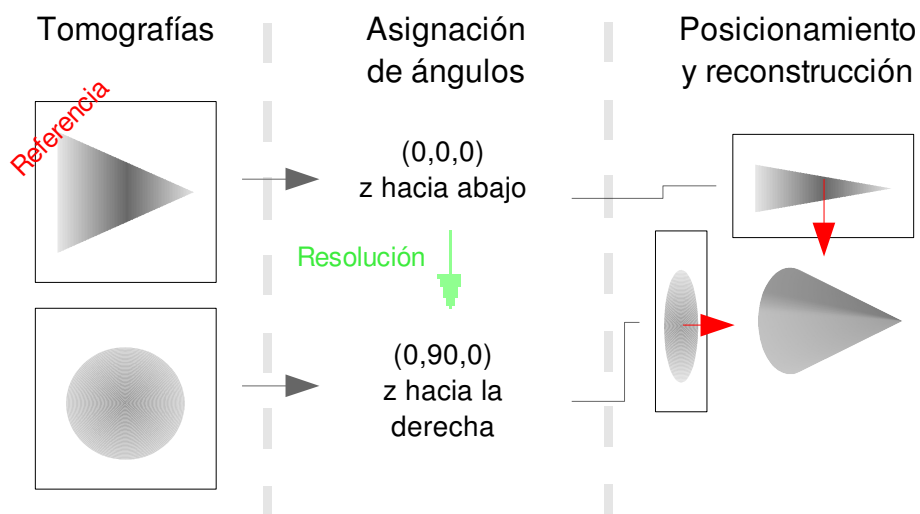


Ilustración 14: Problema de la referencia 1

Vamos a explicar por qué esta asignación arbitraria no afecta al resultado final. Supongamos que tenemos dos tomografías A y B de un volumen V sencillo, en el que no se tendrá en cuenta la simetría por simplicidad. En este caso solo hay dos posibles elecciones para la referencia, la tomografía A o la tomografía B. En la Ilustración 14

podemos ver el proceso de reconstrucción si elegimos la tomografía A como referencia. Se puede observar que la resolución del algoritmo asigna el ángulo de Euler para las tomografías que no son la referencia, en este caso simplificado solo es la tomografía B. El volumen reconstruido se puede ver a la derecha. Si lo comparamos con el otro caso, en la Ilustración 15, podemos ver que el volumen final es idéntico, pero con una rotación que coincide con la diferencia angular entre las dos proyecciones (90°). Dado que el fin de la reconstrucción es obtener la morfología del volumen y no importa su orientación espacial, la elección de la referencia puede ser arbitraria porque en cualquier caso se obtendrá el mismo resultado, solo que con diferente rotación.

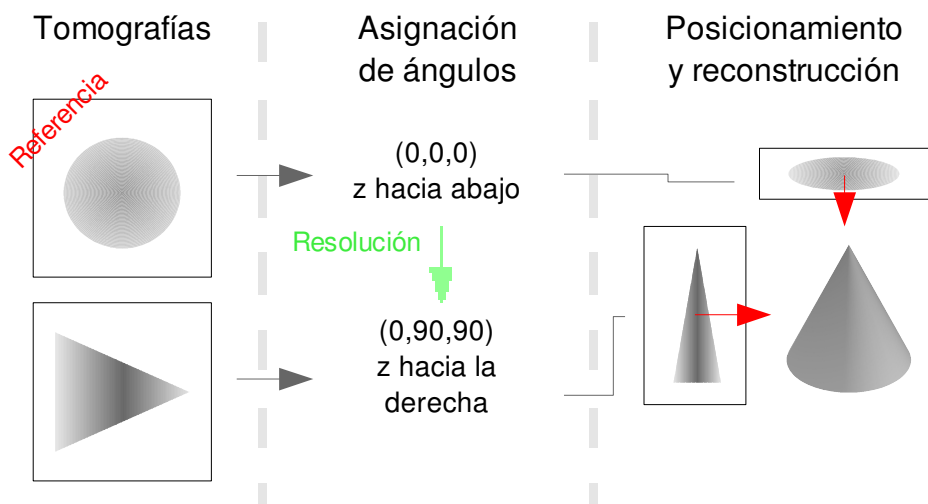


Ilustración 15: Problema de la referencia 2

Otro de los aspectos a tener en cuenta, esencial para cualquier tipo de algoritmo es el tiempo de ejecución. Dado el alto número de grados de libertad de la solución y el rango continuo de éstos, el tiempo de resolución para cualquier técnica de minimización es muy elevado ya que éstas se basan en un alto número de computaciones sucesivas de la misma función objetivo con diferentes datos de entrada (ver punto 3.3.4 Técnicas de minimización).

4.2 Algoritmo

4.2.1 Esquema general

El algoritmo investigado y desarrollado tiene como proceso central una minimización. Esto implica un proceso en el que se preparan los datos de entrada para adaptarlos a lo que se debe introducir en la función objetivo. Recordemos que nuestra entrada será un conjunto de imágenes, para facilitar el procedimiento al usuario se aprovecha la funcionalidad de los ficheros *.sel* (de selección) de *Xmipp*. Estos ficheros contienen los nombres de las *n* imágenes que forman parte del grupo como ya se ha visto en el apartado de *Xmipp* (punto 3.2). Por tanto, el primer paso del algoritmo será adquirir primero las imágenes descritas en ese fichero y luego las matrices de datos que representan dichas imágenes.

El segundo paso implica el establecimiento de los límites de búsqueda para el algoritmo así como otras variables. Los límites de búsqueda serán (0,360) para el primer y último ángulo de Euler de cada imagen y (0,180) para el segundo. Como veremos, cada individuo de cada generación estará formado por los tres ángulos de Euler de cada imagen uno tras otro, por lo que los límites inferiores serán un vector de ceros de $3 \cdot n$ (n = número de imágenes) de longitud. Los límites superiores serán de la forma:

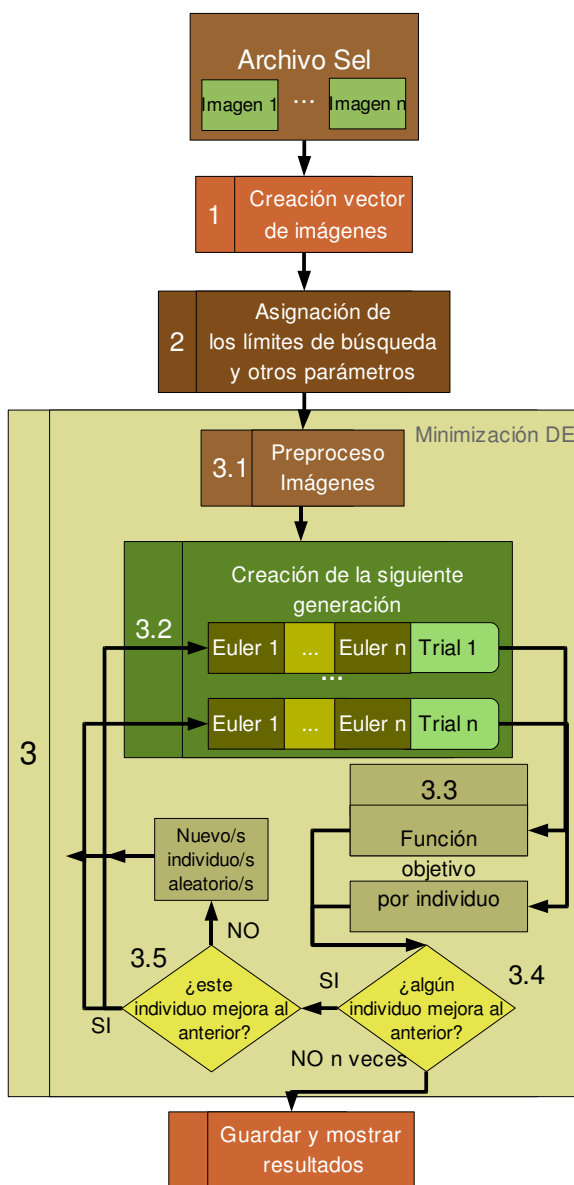
$$(360, 180, 360, 360, 180, 360, \dots, 360, 180, 360) \text{ con } 3 \cdot n \text{ elementos}$$

limites de la primera imagen limites de la segunda imagen limites de la imagen n

El resto de variables que deben ser definidas tienen que ver con el proceso de *Differential Evolution* (Número de generaciones, individuos por generación, tipo de minimización, etc.) o del criterio de finalización de la minimización (umbral de finalización, número de veces que puede no haber cambios, etc.). Todas estas variables definen aspectos esenciales de la búsqueda de la solución óptima, por lo que deberán

ajustarse en función de los resultados obtenidos, algunos de ellos podrán ser seleccionados por el usuario.

En este momento se inicia el proceso de minimización, dado que en él se produce una computación intensiva y reiterada de la función objetivo se debe simplificar ésta lo máximo posible, por ello hay que generar un conjunto de datos que nos aporte la información preprocesada para eliminar este cálculo de la función objetivo.



En el siguiente paso (3.2) se entra en el bucle principal de la minimización. En él se crea la nueva generación de individuos. En un principio esta generación es aleatoria entre los límites establecidos en el paso 2, pero los individuos de las siguientes generaciones serán producto de la mutación de los individuos de las generaciones anteriores en función del resultado que dieron al ser introducidos como entrada de la función objetivo. Cada individuo es una entrada de la función objetivo (3.3), si ningún individuo de la generación mejora el resultado general de la generación anterior en n repeticiones seguidas del bucle, suponemos que hemos llegado al resultado deseado porque no se ha evolucionado nada en n generaciones. Si algún individuo cambia, esto significa que se ha mejorado la

Ilustración 16: Algoritmo general

generación anterior y aún hay posibilidad de mejorar aún más, por lo que se crea una nueva generación. Se evalúa el resultado de cada individuo y si es mejor que el de la generación anterior prevalecerá frente al individuo del que procedía. Si el resultado es peor, se creará un nuevo individuo a partir de los parientes iniciales. Con suficientes generaciones e individuos por generación se puede crear una población lo suficientemente diversa como para que uno de los individuos sea el que genere el mínimo de la función objetivo y, por tanto, prevalezca ante los demás.

4.2.2 Idea inicial

La idea inicial consistía en utilizar una función objetivo que aplicaba una aproximación por línea común y la minimización general de los ángulos de Euler absolutos. Este primer planteamiento que se hizo trataba de unir el conocimiento que tenemos sobre la relación entre los ángulos de Euler, las transformadas de las proyecciones 2D y las líneas comunes con la potencia del algoritmo *Differential Evolution* (DE) para la minimización. El proceso consistía en, dado un número indefinido de proyecciones, se les asigna, mediante DE unos ángulos de Euler aleatorios dentro de los límites, para cada uno de esos ángulos. Después se calcula para cada pareja de proyecciones 2D la línea común entre ellas en los respectivos sistemas de coordenadas locales y se halla el error cuadrático entre ellas. Sumando el error generado por todas las parejas de proyecciones se obtiene la cantidad a minimizar. Por desgracia, en el planteamiento no se tuvieron en cuenta algunas características especiales de este tipo de procesos, lo que hace inservible el procedimiento.

Existen varios problemas con esta aproximación. El primero y más importante es que la comparación de las líneas comunes de las transformadas de Fourier (Ilustración 17 izquierda) es mucho más sensible a ruido (tanto ruido de la imagen como ruido de cuantificación) que la comparación de las proyecciones 1D comunes en el espacio real (Ilustración 17 derecha), debido a que las proyecciones 1D promedian todos los valores de la imagen.

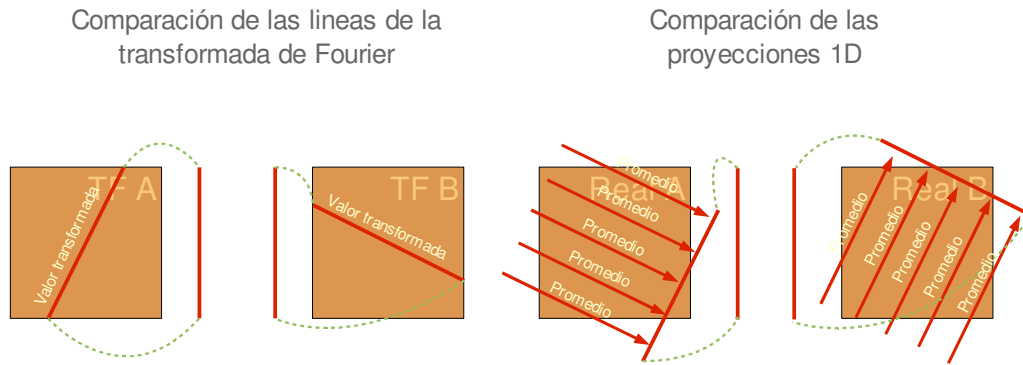


Ilustración 17: Comparación línea común y proyección común

El segundo error de planteamiento de esta aproximación, descubierto durante las pruebas realizadas, es que es imposible minimizar el valor de los ángulos de Euler absolutos de las tomografías, ya que hay infinitas combinaciones posibles al no haber un sistema de referencia concreto. Como se explica en la sección 4.1 en la página 51, y se puede ver en las ilustraciones 14 y 15, si poseemos los ángulos de Euler de una serie de tomografías podremos obtener el volumen que representan, si las rotamos 90 grados sobre el eje z obtendremos el mismo volumen solo que girado 90 grados respecto a la representación anterior. Por lo tanto también impide la convergencia el hecho de que para cualquier pareja de tomografías y dados unos ángulos cualquiera de la primera se pueden obtener unos ángulos de Euler cuyo error a lo largo de la línea común sea mínimo.

El último problema es que si se pone un número suficientemente alto de tomografías el número de variables para el algoritmo de *Differential Evolution* es demasiado alto por lo que no converge bien. Esto se debe a que por cada tomografía existen tres ángulos de Euler a resolver. Esto hace que el espacio en el que debe buscar el algoritmo aumente cuantas más proyecciones se incluyan en la ejecución. Por ejemplo con tres proyecciones, tomando una como referencia cuyos ángulos son fijos, nos encontramos con un espacio de seis dimensiones, pero con cinco las dimensiones aumentan a 12. Es muy difícil que el algoritmo encuentre la solución ya que debe probar en demasiadas dimensiones, pero esto se puede solucionar aumentando el número de individuos por generación, sin embargo esto produce que los tiempos de ejecución

crezcan exponencialmente, por lo que se debe llegar a un compromiso entre número de proyecciones a resolver y tiempo. Una solución adicional es resolver un número menor de proyecciones y, una vez que se han fijado sus ángulos de Euler, ir añadiendo más proyecciones al algoritmo, aunque es preferible, si se dispone de potencia de cálculo y tiempo suficiente la resolución de todas las proyecciones a la vez.

4.2.3 Uso de la proyección común

El segundo planteamiento intentó paliar los defectos del anterior aprovechando sus ventajas y tratando de comprobar si el número de variables a concretar para DE era demasiado alto para alcanzar la convergencia, o esto se debía solo a la falta de un sistema de referencia.

Tras bastantes pruebas con el anterior modelo se observó que el error sobre la línea común de la transformada de Fourier de las proyecciones 2D era más sensible a ruido que la otra opción, la covarianza entre las proyecciones 1D de las proyecciones 2D en el espacio real. Por ello se sustituyó la comparación anterior con la nueva.

Dado que resultó imposible obtener la convergencia del proceso se trató de averiguar a qué era debido. Aparecieron dos causas posibles descritas en la sección de Problemática (4.1). La primera, que afecta al proceso sin género de dudas, es la falta de un sistema de referencia, como ya se ha explicado sin éste la convergencia es imposible. La otra era la dimensión del problema de minimización que era tres veces el número de proyecciones 2D. A mayor número de proyecciones peor convergencia, al menos en principio. Para comprobar esto se plantearon dos aproximaciones, una minimización general con una de las proyecciones 2D como referencia (es decir con sus ángulos de Euler a $(0,0,0)$), que es la que nos ocupa, y una minimización particular por grupos de tres proyecciones manteniendo una de ellas constante y como referencia. El segundo caso se explica en el punto siguiente, mientras que el primero se explicará aquí.

El proceso es el siguiente: Primero fijamos los ángulos de Euler de una de las proyecciones a (0,0,0), luego realizamos DE con todas las proyecciones, asignando ángulos de Euler según el algoritmo. Para cada cálculo de la función objetivo se cogen las proyecciones por parejas, haciendo la combinación de todas las $p = \frac{n \cdot (n-1)}{2}$ parejas posibles, tomando n como el número de tomografías. Con los ángulos de Euler asignados por DE a cada una de las tomografías de la pareja se obtiene la línea común tridimensional entre ellas. De ésta se obtienen los ángulos planos en el espacio de sus respectivas tomografías, recordemos que la línea común debe, precisamente, pertenecer a ambos planos en el espacio tridimensional y por tanto representa una línea en el espacio de cada plano que la contiene. Desde cada uno de estos dos ángulos planos se obtiene la proyección 1D de su respectiva tomografía, obteniendo dos vectores $v_{1,p}$ y $v_{2,p}$. El objetivo será la minimización de la suma de la inversa del módulo de los índices de correlación entre las proyecciones 1D de las tomografías de todas estas parejas.

Visto desde una perspectiva matemática, se puede resumir este problema con el siguiente enunciado. Dada una pareja de proyecciones pertenecientes a uno de los individuos de una de las iteraciones del bucle de minimización, se tienen dos vectores $v_{1,p}$ y $v_{2,p}$ que representan las respectivas proyecciones 1D en sentido perpendicular a la línea común de las proyecciones 2D que generan los ángulos de Euler escogidos como posible solución para dicha pareja, la función a minimizar es

$$f(v_{1,p}, v_{2,p}) = \frac{-1}{P} \sum_{p=1}^P \text{cov}(v_{1,p}, v_{2,p})$$

obteniendo tras sucesivas asignaciones los ángulos de Euler de los que se obtiene el menor valor de ésta función (cercano a -1) para cada una de las proyecciones 2D.

4.2.4 Introducción de la minimización por partes

Esta aproximación es un derivado directo de la anterior en la que se intentaban obtener los ángulos de Euler de todas las proyecciones a la vez. Dado que era posible

que esto afectase a la convergencia del proceso DE, se ideó una forma de reducir el número de incógnitas. Para ello se realizará la minimización anterior pero para grupos de tres proyecciones. Una de ellas será siempre la de referencia y tendrá sus ángulos de Euler iguales a cero. El proceso es, por tanto, idéntico al anterior, solo que realizado poco a poco. Esto debería eliminar el problema de la no-convergencia para proyecciones cuyas normales sean linealmente independientes (es decir, que formen una base de R^3). Si alguna de las proyecciones no es independiente de las demás el proceso no convergerá, por lo que hay que tener especial cuidado en este punto.

Los resultados tras esta implementación eran esperanzadores ya que los ángulos se resolvían satisfactoriamente para algunas de las tomografías del proceso. La forma de calcular este éxito también implica cierto desarrollo. Dado que la proyección de referencia no tiene los ángulos de Euler específicos desde los que fue tomada, sino que tiene (0,0,0), como ya se ha explicado las

demás tomografías se situarán en el espacio con una rotación similar a la que tenían al principio, pero como la proyección de referencia se ha rotado (a no ser que sus ángulos ya fueran (0,0,0) desde un principio), las demás habrán rotado también un ángulo 3D que depende de la posición de sus normales y la orientación relativa con respecto a la proyección de referencia. Por

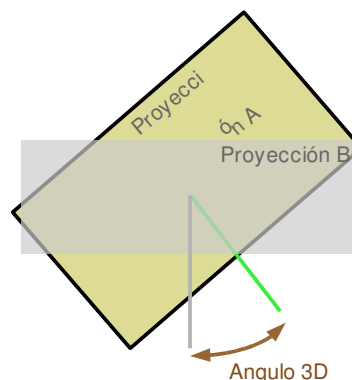


Ilustración 18: Cálculo del ángulo 3D
 tanto, para comprobar que todas las proyecciones se han situado en el lugar correcto utilizaremos los ángulos 3D entre ellas. Estos ángulos, como puede verse en la Ilustración 18, se calcularán como el ángulo 3D entre dos vectores, que serán las normales de los planos de proyección.

Por tanto, la forma de calcular la eficiencia del algoritmo comienza por el cálculo del ángulo 3D entre las normales de los planos de proyección tomados desde los ángulos de Euler reales. Entonces, estos ángulos se comparan con los ángulos 3D de las

normales de los planos proyectados según los ángulos de Euler calculados mediante el algoritmo, de tal forma que si son iguales para cada pareja la asignación de ángulos ha sido correcta.

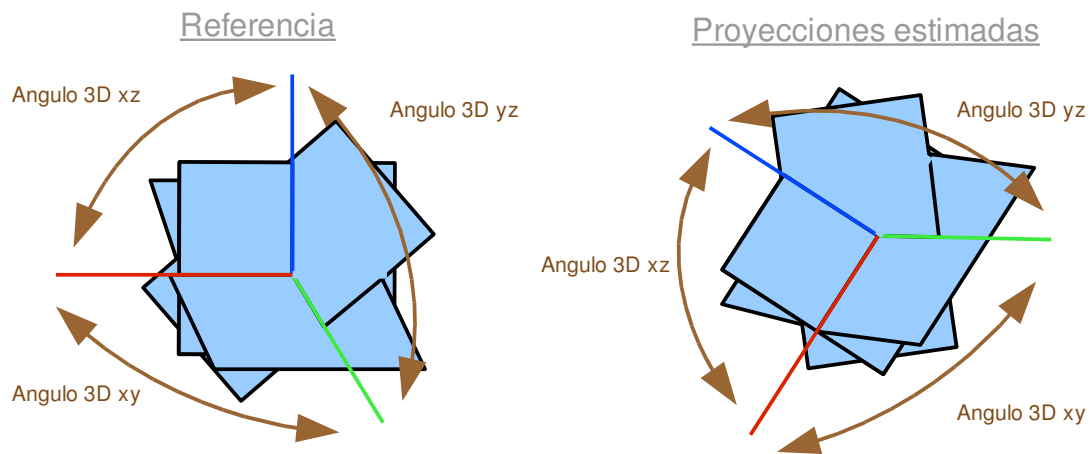


Ilustración 19: Comparación de ángulos 3D

4.2.5 Desarrollo final

Para el desarrollo final se han mejorado algunos aspectos que, pese a no ser errores de base sí amplían la funcionalidad del algoritmo en tres aspectos fundamentales: velocidad, generalidad y eficacia.

Para cumplir con la generalidad establecida en los objetivos de este proyecto se desechó la minimización por partes ya que no suponía una mejora respecto a lo ya existente al resolver los ángulos para grupos de tres proyecciones como máximo. Se retomó por tanto la minimización general de todas las proyecciones introducidas. Para solventar el problema de la convergencia se realizaron diversas pruebas que dieron como resultado un avance importante. La idea es que a medida que se aumenta el número de proyecciones introducidas se aumente proporcionalmente el número de individuos por generación, con lo que se obtienen buenos resultados como puede verse en la sección 5, denominada Resultados.

El proceso descrito hasta ahora conlleva el procesamiento de rotaciones, multiplicaciones de matrices y otras operaciones como la proyección 1D que, si bien no son demasiado costosas en un proceso singular con la tecnología actual, debido a la especial naturaleza de la minimización evolutiva, su cálculo reiterado puede consumir abundantes recursos incluso en los sistemas más avanzados. Por ello, cualquier mejora de tiempo en la función objetivo, aunque sea de milisegundos, se traduce en una velocidad mucho mayor del algoritmo general, llegando a evitarse horas de computación. Conociendo esto se consiguió integrar con éxito una mejora sustancial del algoritmo efectuando un preproceso de las imágenes a evaluar, incluyéndolo en el paso 2 del Algoritmo general (Ilustración 16 en la página 54). Dado que no estamos interesados en la información de la imagen, sino en la proyección 1D desde distintos ángulos, el paso evidente era procesar la transformada Radon (ver Punto 3.3.5) para cada tomografía en intervalos de 1° . Así cuando se necesitase la proyección 1D de la tomografía n en el ángulo a , solo habría que buscar el vector situado en la columna a de la transformada Radon n . Aquí se introduce en el proceso un pequeño error de cuantización, ya que la transformada Radon nos proporciona las proyecciones 1D de una imagen en ángulos discretos, por ello, en caso de que el ángulo en cuestión tenga parte decimal se utiliza un redondeo al ángulo entero más cercano y se coge esa columna de la transformada Radon.

El problema de la proyección reflejada, que se desarrolló en el apartado de Problemática en la página 50, afecta al cálculo de la proyección 1D común. Se da el hecho de que para calcular la línea común de dos planos de proyección A y B se necesitan las normales de dichos planos. En el problema que nos ocupa hay dos puntos desde los que se puede proyectar una misma proyección 2D, dando lugar a dos normales por proyección que son vectores con la misma dirección y sentido opuesto. Por tanto tendremos cuatro posibles normales: $normalA1$, $normalA2$, $normalB1$ y $normalB2$ y cuatro formas de combinar una $normalA$ con una $normalB$ para, mediante el producto vectorial, calcular la línea común con la que se podrá hallar la proyección común para cada plano. La línea común será la misma independientemente de la combinación utilizada, pero el problema viene porque los planos que las generan no contienen la

misma información, de hecho, la normalA2 hace referencia a un plano que contiene la misma información que el de la normalA1 pero reflejada. Esto implica que el mismo punto de proyección 1D obtenido de dos combinaciones equivalentes puede generar resultados distintos debido a la reflexión del plano sobre el que se aplica.

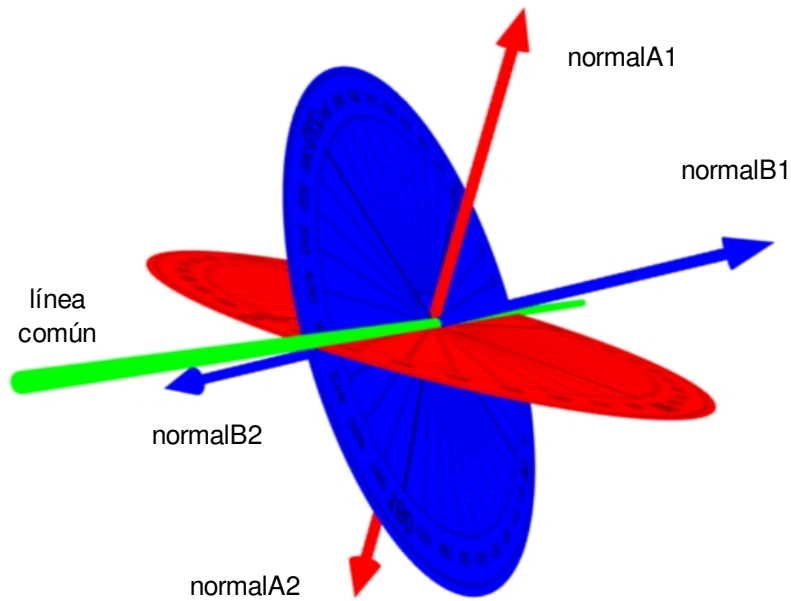


Ilustración 20: Problema de la proyección reflejada

Otro de los problemas se dio porque se observó que la correlación de las proyecciones, pese a ser un buen comparador, carecía de la potencia suficiente por sí sola para distinguir entre dos proyecciones parecidas y dos proyecciones exactamente iguales. Para mejorar este aspecto se añadió a la comparación la correlación de la derivada de las proyecciones, con la intención de que no solo los valores fuesen parecidos, sino también sus cambios a lo largo del vector. Así la función a minimizar será

$$f(v_{1,p}, v_{2,p}) = \frac{-1}{P} \frac{\sum_{p=1}^P \text{corr}(v_{1,p}, v_{2,p}) + \text{corr}(\partial v_{1,p}, \partial v_{2,p})}{2}$$

siendo p la pareja a la que pertenecen las dos tomografías, como ya se explicó en el punto anterior.

5 Resultados

5.1 Resumen

Durante el desarrollo del proyecto se han ideado e implementado diversos experimentos y pruebas para comprobar las teorías y los procesos aplicados. Los resultados de estos experimentos han servido para modificar algunas ideas iniciales, como el uso de la proyección 1D común de espacio real en vez de la línea común del espacio transformado descrito en la sección 4.1, y así ir mejorando el algoritmo hasta que los resultados fuesen los esperados o cumplieran los objetivos establecidos.

El primer experimento se correspondía con el objetivo de encontrar primero la línea común en el espacio transformado y después la proyección 1D común en el espacio real entre dos proyecciones bidimensionales que pertenezcan al mismo volumen. Una vez logrado ese objetivo y establecida la proyección 1D común como elemento de alineación entre dos proyecciones 2D se buscó conocer si de entre todas las proyecciones 1D entre las dos proyecciones 2D la que se correspondía con los Ángulos de Euler de proyección reales era la que mayor correlación tenía. Antes de este experimento se utilizaba el error cuadrático medio, pero se comprobó que la correlación era una medida mejor de la similitud entre dos proyecciones 1D para este proyecto, además, se añadió la correlación entre la primera derivada de las proyecciones 1D, lo que mejoró los resultados aún más. Por último, se implementó la minimización por *Differential Evolution* y se experimentó variando su modo de aplicación y los parámetros que la definen hasta que se obtuvieron ángulos de proyección prácticamente idénticos a los buscados para ciertos casos, con ello se corroboró la problemática del aumento de la dimensión del problema al aumentar el número de proyecciones involucradas y se decidió implementar el preproceso de la transformada Radon, ya que se comprobó que no introducía un error de cuantificación excesivo y los resultados eran muy parecidos con tiempos de ejecución drásticamente menores.

5.2 Volúmenes a estudio

En los experimentos de este proyecto se han utilizado dos volúmenes, la sub-unidad asimétrica del *Hexón del Adenovirus humano Tipo 5* [51] y la bacteria *Escherichia coli* [52].

Sub-unidad Asimétrica del Hexón del Adenovirus Humano Tipo 5 (1RUX)

Adenoviridae (adenovirus) es una familia de virus que infectan tanto humanos como animales. Los adenovirus fueron aislados en las adenoides (amígdalas) de los humanos, de donde deriva el nombre. Posee un genoma ADN bicatenario por lo que son clasificados en el Grupo I bajo el esquema de la Clasificación de Baltimore. Son de tamaño medio (90-100 nm), sin envoltura y con forma icosaédrica. Los capsómeros de su cápsida tienen forma hexagonal y pentagonal. Lo más importante desde el punto de vista de nuestro estudio es la estructura de uno de sus hexones, por eso se muestran a continuación una serie de vistas de este volumen.

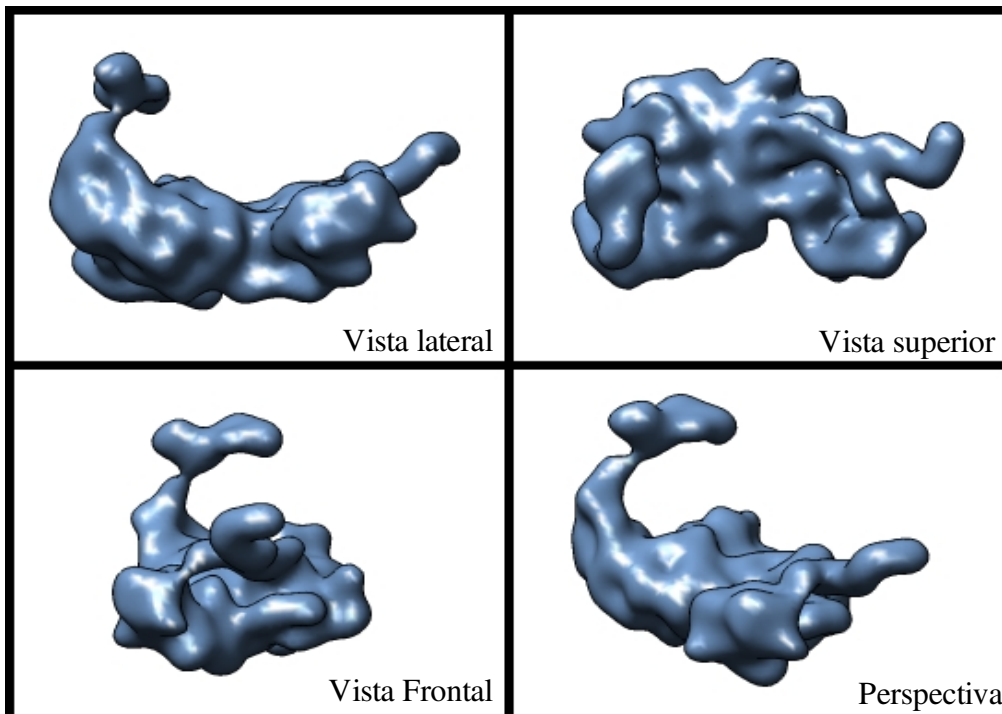


Ilustración 21: Vistas del volumen del hexón del adenovirus humano 5

Como se puede apreciar este volumen no tiene simetría en ningún eje, con lo que no existirán dos proyecciones iguales más allá de las reflejadas (tomadas desde ángulos opuestos).

Bacteriorodopsina (BRD)

La bacteriorodopsina es un pigmento fotosintético usado por las archaea, principalmente Halobacterias. Actúa como bomba de protones, es decir, captura energía de la luz y la utiliza para mover los protones a través de la membrana celular. El gradiente protónico que resulta se convierte posteriormente en energía química.

La bacteriorodopsina es una proteína integral de membrana encontrada generalmente en parches cristalinos bidimensionales de color púrpura, que pueden ocupar hasta casi el 50% del área superficial de la célula de la archaea. La celda elemental de la red hexagonal se compone de tres cadenas idénticas de proteína, cada una rotada 120 grados con respecto a las otras. Cada cadena esta compuesta de siete hélices alfa transmembrana y una molécula retiniana localizada profundamente en su interior, la estructura típica de las proteínas retinianas. El volumen a estudio se compone de las siete hélices de una cadena de la bacteriorodopsina.

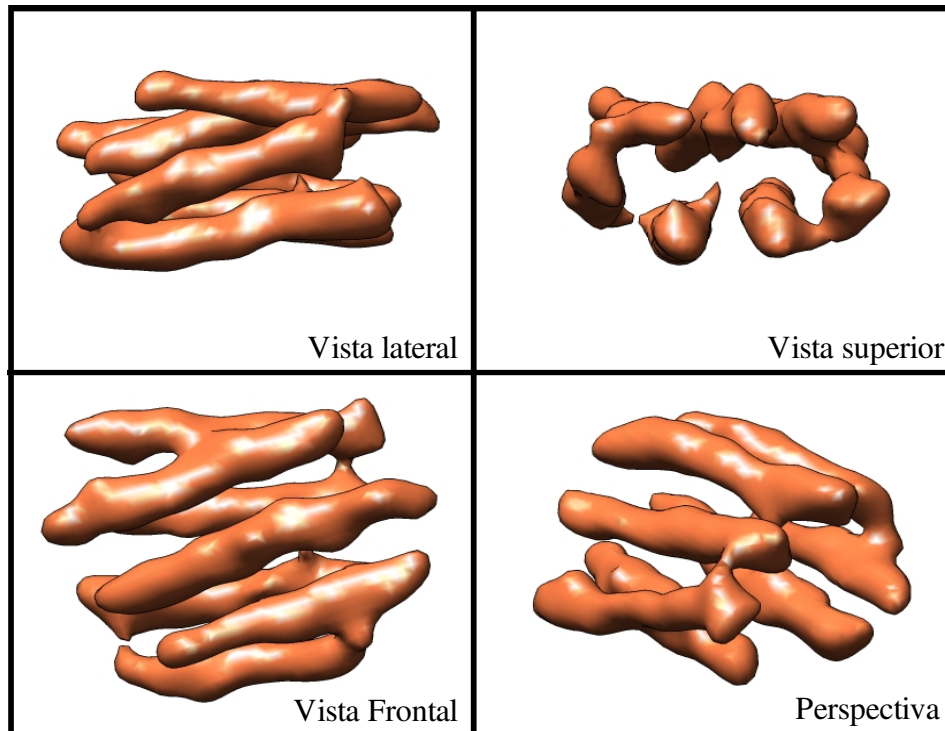


Ilustración 22: Vistas del volumen de las hélices alfa de la bacteriorodopsina

Este volumen tampoco es simétrico aunque se compone de siete cuerpos alargados por lo que puede que casualmente varias proyecciones sean iguales, pero sin seguir ningún tipo de patrón por simetría.

5.3 Obtención de la proyección común

Intentando comprobar la validez de la teoría de la proyección 1D común útil en la alineación entre dos proyecciones 2D de un mismo volumen se ha desarrollado un programa que, dadas éstas y los ángulos de Euler desde las que fueron tomadas obtiene la gráfica de la proyección 1D correspondiente a cada proyección y la correlación entre ellas. Como establece la teoría, si las tomografías pertenecen al mismo volumen esta correlación será cercana a uno mientras que si son de volúmenes diferentes la correlación podrá tomar cualquier valor, que en la mayoría de los casos no estará próximo a uno. En este experimento se comprobó dicha teoría y nos permitió decidir

que la proyección 1D en el espacio real es más efectiva que la línea común en el espacio transformado.

Experimento 1 Caso 1: Proyección de 1BRD con 1BRD

La primera comparación que haremos será entre dos proyecciones pertenecientes al mismo volumen. En este caso, utilizando la partícula *bacteriorodopsina*, las proyecciones 1D deberían ser iguales y por tanto su correlación cercana a uno.

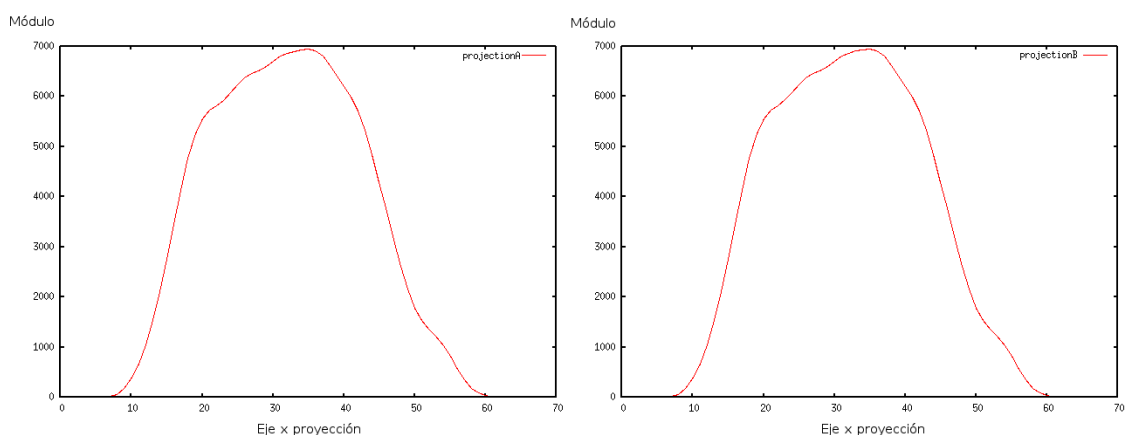


Ilustración 23: Comparación de las proyecciones 1D comunes para la partícula 1BRD

Resultados:

Nombre de la partícula	Tamaño de las proyecciones	Ángulos de Euler de cada proyección	Ángulos bidimensionales de rotación	Medida de la Correlación
1BRD	64x64	(329.76° , 61.39 ° , 99.69°) (206.95° , 129.65° , 149.25°)	70.57° -35.77°	0.99999

Tabla 1: Resultados del primer caso del primer experimento

Como vemos la medida de la correlación es, tal y como esperábamos, cercana a 1. Además podemos ver los ángulos bidimensionales de rotación que no son más que el número de grados que hay que girar cada proyección para que su línea común sea la línea horizontal que pasa por el centro. Desde esta posición y si las dos proyecciones 2D pertenecen al mismo volumen las dos proyecciones 1D tomadas desde arriba

coincidirán. Esto es útil para el procesamiento ya que nos permite proyectar simplemente sumando los valores de todas las columnas y es el procedimiento que se utiliza en el algoritmo.

Experimento 1 Caso 2: Proyección de 1BRD con 1LRUX

En este caso queremos calcular la medida de la correlación utilizando los ángulos de proyección reales en dos proyecciones que pertenecen a dos volúmenes distintos. Según la teoría, estas proyecciones no tienen por qué ser parecidas por lo que su medida de la correlación no será cercana a 1. Podemos ver que la teoría es correcta en la siguiente gráfica:

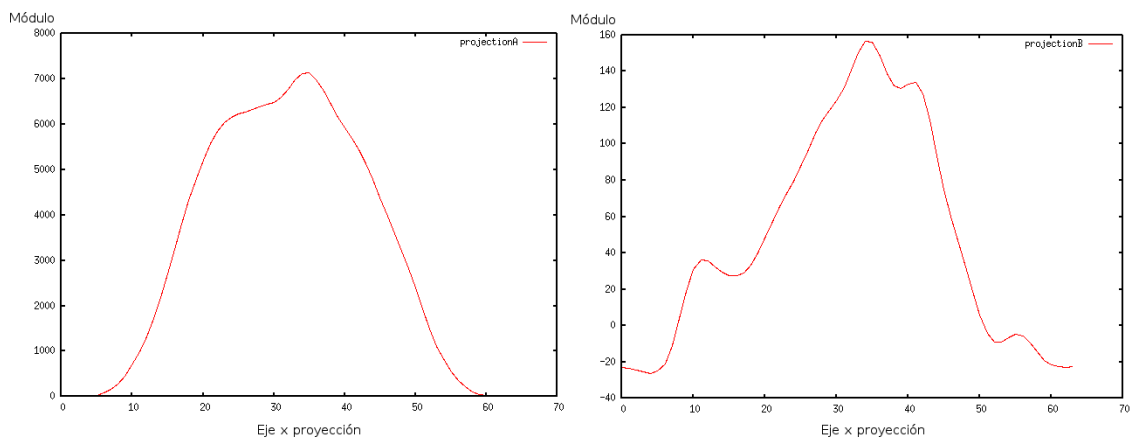


Ilustración 24: Proyecciones 1D comunes entre dos proyecciones de partículas distintas

Resultados:

Nombres de las partículas	Tamaño de las proyecciones	Ángulos de Euler de cada proyección	Ángulos bidimensionales de rotación	Medida de la Correlación
1BRD - 1RUX	64x64	(329.76° , 61.39 ° , 99.69°) (0° , 90 ° , 90°)	-40.90° -42.71°	0.802395

Tabla 2: Resultados del segundo caso del primer experimento

Experimento 1 Caso 3: Proyección de 1RUX con 1RUX

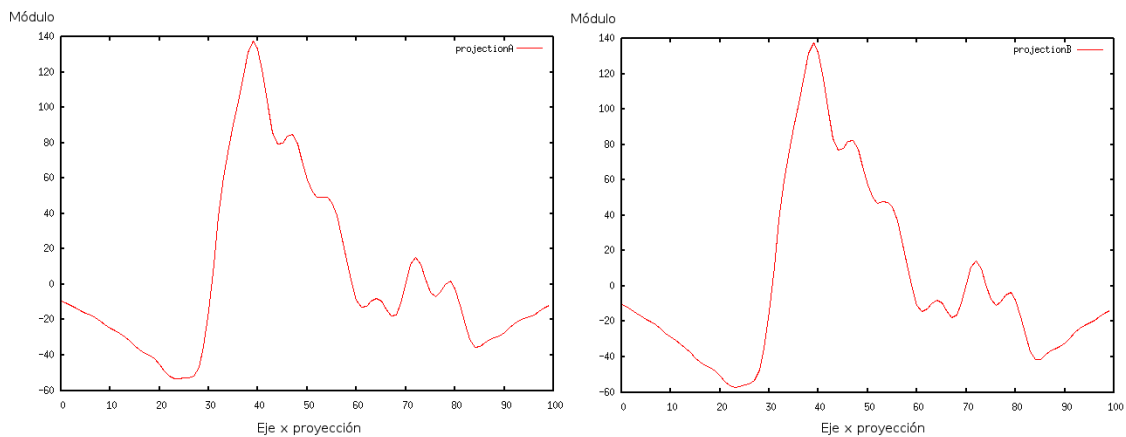


Ilustración 25: Proyecciones 1D comunes entre dos proyecciones de la partícula 1RUX

Resultados:

Nombre de la partícula	Tamaño de las proyecciones	Ángulos de Euler de cada proyección	Ángulos bidimensionales de rotación	Medida de la Correlación
1RUX	100x100	(224.27° , 58.09 ° , 222.96°) (267.06° , 67.98° , 273.68°)	46.72° -62.33°	0.99904

Tabla 3: Resultados del segundo caso del primer experimento

Como podemos observar la medida de la correlación es cercana a dos, pero no tanto como en el primer caso, esto puede deberse a varios factores como que el tamaño de las proyecciones es mayor, y por tanto hay más puntos en los que comparar, o que en el proceso de proyección se ha introducido más ruido.

5.4 Búsqueda de la proyección común

El experimento que se expondrá en este apartado se desarrolló para determinar si la implementación de la obtención de la correlación entre las proyecciones 1D comunes a partir de los ángulos de Euler era correcta y, sobre todo, útil para el proyecto. Para ello nos basamos en la teoría de la proyección común que establece que de entre todas las

combinaciones de las proyecciones 1D comunes de las dos tomografías las que mayor correlación tienen son las que corresponden a la proyección 1D desde la línea común.

El procedimiento comienza obteniendo la correlación entre las proyecciones 1D de ambas tomografías desde el ángulo que establece la línea común para cada una de ellas. Entonces se calculan las 358 correlaciones que se obtienen entre la proyección 1D de la primera proyección y las 358 proyecciones que se toman girando la segunda proyección en intervalos de un grado respecto al ángulo inicial. Después se gira un grado la primera proyección y se obtienen las 359 correlaciones mediante el giro de la segunda proyección. El proceso continúa hasta que la primera proyección ha sido girada 359 grados. Lo que se obtiene son todos los posibles valores de la covarianza entre las proyecciones 1D de las proyecciones 2D, y se puede comprobar que si las proyecciones pertenecen al mismo volumen y hemos calculado bien los ángulos de proyección la primera correlación será mayor que todas las demás y además muy próxima a uno.

Si se ejecuta esta prueba mostrando los datos como gráficas que van de izquierda a derecha (el valor más a la izquierda es el valor de la correlación para las proyecciones 1D desde la línea común), la mayor correlación debería ser el primer valor en caso de que las proyecciones pertenezcan al mismo volumen. A continuación se muestran los resultados de varios experimentos que se realizan girando solo la segunda proyección, no tomamos datos sobre el giro de la primera por simplicidad, en este caso, por tanto, podremos observar el cambio de la correlación sobre 359 muestras, en vez de 359×359 .

Experimento 2 Caso 1: Correlaciones entre 1BRD y 1BRD

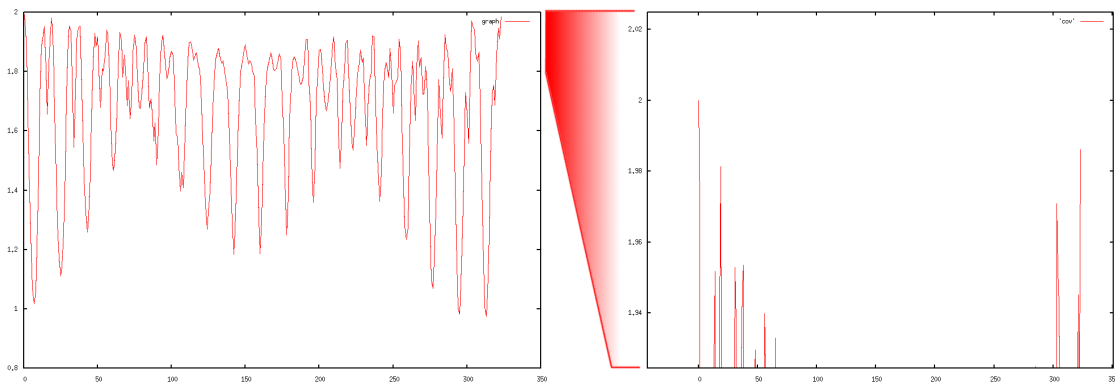


Ilustración 26: Comparación de la covarianza 1BRD

Nombre de la partícula	Tamaño de proyecciones	Ángulos de Euler de cada proyección	Ángulos 2D de rotación	Correlación máxima	Correlación mínima
1BRD	64x64	(276.91°, 61.76°, 237.94°) (264.18°, 98.96°, 90.53°)	-11.89° -18.43°	0.99999	0.487485

Tabla 4: Resultados del primer caso del segundo experimento

Como se puede observar se cumple que la primera correlación es mayor que todas las demás para este caso en el que las proyecciones pertenecen a un mismo volumen. También podemos ver que existen algunos puntos que se acercan al valor de la primera correlación, cerca del 1, lo cual nos conduce a pensar que si las dos proyecciones se viesan afectadas por ciertos niveles de ruido podrían confundirse estos picos que nada tienen que ver con los ángulos de proyección con el que se origina en las proyecciones 1D obtenidas de los ángulos de Euler reales.

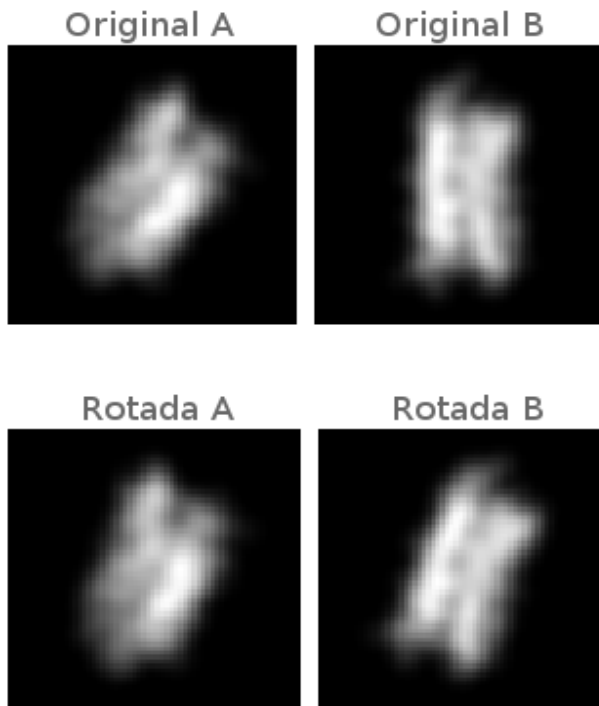


Ilustración 27: Alineación 2D en base a líneas comunes

En la Ilustración 27 podemos ver el resultado de alinear las imágenes originales (proyecciones 2D de un mismo volumen) según los ángulos 2D de rotación en los que se calcula que están situadas las líneas comunes. Estos ángulos se muestran en la Tabla 4. Al rotar cada proyección su ángulo correspondiente, las líneas también rotan colocándose en la horizontal, de tal forma que si proyectamos las imágenes desde arriba obtendremos dos proyecciones 1D idénticas. Se puede observar que como las dos proyecciones tienen una estructura

similar su orientación final es parecida siendo bastante evidente que, efectivamente, sus proyecciones desde arriba serán parecidas.

Experimento 2 Caso 2: Correlaciones entre 1BRD y 1RUX

Al buscar la línea común entre dos proyecciones de dos partículas diferentes podemos comprobar que la mayor correlación no se acerca a 1, que es el máximo de similitud, sino que tiene un valor de 0,91. De hecho, este máximo no se encuentra en el ángulo que correspondería a los ángulos de Euler reales desde los que se tomaron las proyecciones, ni en ningún punto relacionado. Estos dos factores nos hacen suponer que la mayor correlación se ha dado en un punto aleatorio en el que se da que las proyecciones son parecidas, aunque no iguales.

Esto nos hace comprender la importancia de discriminar previamente las proyecciones obtenidas que no provengan de la estructura bajo estudio, ya sea por que

pertenezcan a otra molécula o a causa de ruido concentrado. En caso de que una proyección extraña se introdujese en el proceso impediría la alineación de las proyecciones al introducir un error en las comparaciones de la correlación.

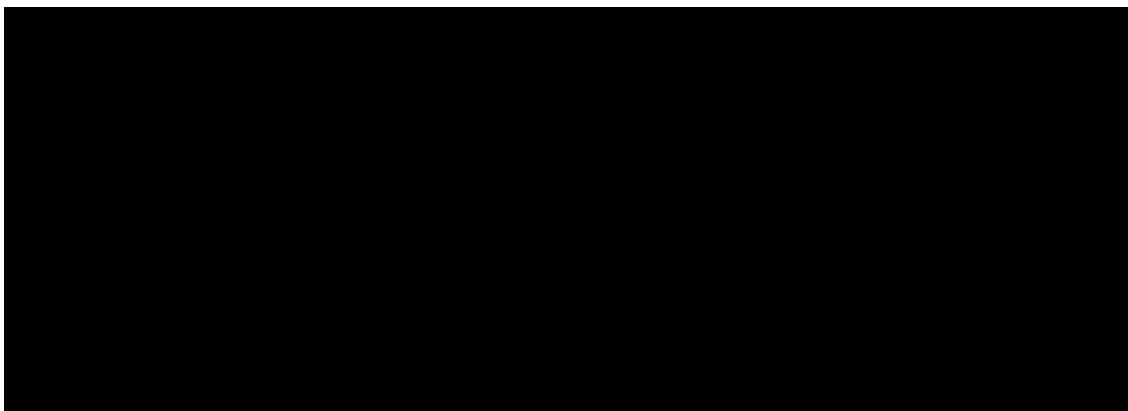


Ilustración 28: Comparación de la correlación 1BRD y 1RUX

Nombre de la partícula	Tamaño proyecciones	Ángulos de Euler de cada proyección	Ángulos 2D de rotación	Correlación máxima	Correlación mínima
1BRD y 1RUX	64x64	(26.68°, 107.86°, 112.24°) (0°, 90°, 200°)	-36.35° -55.66°	0.908955	0.453283

Tabla 5: Resultados del segundo caso del segundo experimento

Experimento 2 Caso 3: Correlaciones entre 1RUX y 1RUX

En este caso utilizamos proyecciones de mayor tamaño. Como se puede ver en la Ilustración 29 a la derecha la correlación en la línea común (en la primera muestra) se acerca a 2 y se diferencia claramente del resto de correlaciones en el espectro del experimento, por lo que el ruido afectará menos al proceso. Como era de esperar esto se ha dado debido a que al utilizar proyecciones de mayor tamaño, lo que significa mayor resolución, la cantidad de información que se posee para discriminar el punto de máxima correlación es mayor.

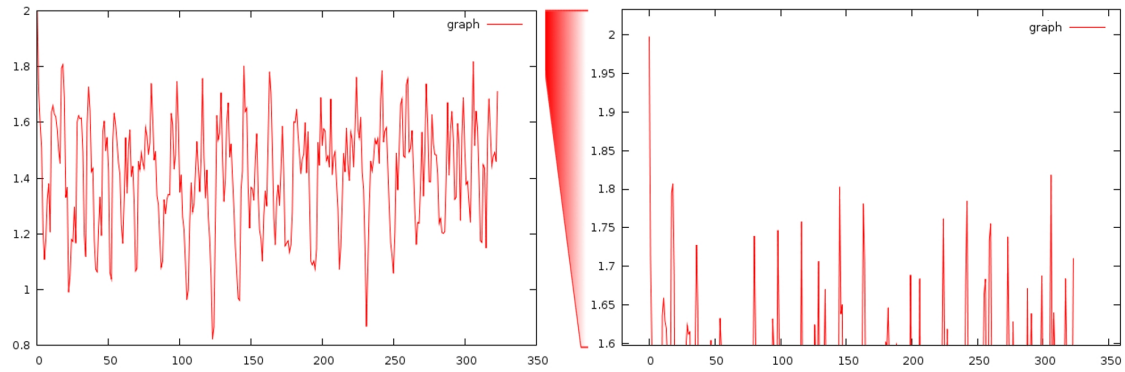


Ilustración 29: Comparación de la covarianza 1RUX

Nombre de la partícula	Tamaño proyecciones	Ángulos de Euler de cada proyección	Ángulos 2D de rotación	Correlación máxima	Correlación mínima
1RUX	100x100	(262.35°, 105.30°, 104.71°) (224.27°, 58.09°, 222.96°)	-22.63° 89.39°	0.9983	0.411

Tabla 6: Resultados del tercer caso del segundo experimento

En la Tabla 6, que contiene los resultados de este experimento la correlación máxima es de 0.9983, comparada con la del primer caso, que era de 0.99999 se aprecia una diferencia del orden de 0.001. Esto también es debido al aumento de tamaño de las proyecciones, al haber más muestras que comparar, es más probable que durante el procesado de la proyección se hayan variado ligeramente los valores a comparar, por lo que el nivel máximo de la correlación disminuye. Esto implicaría que el proceso sería más sensible al ruido si no fuera por la disminución de los máximos locales ya explicada, que contrarrestan los efectos negativos de la disminución de la correlación.

5.5 Obtención de ángulos de Euler de varios grupos de proyecciones

Recordemos que el objetivo final de este proyecto era la obtención de los ángulos de Euler desde los que se proyectaron un cierto número de proyecciones. En este experimento se satisface dicho objetivo gracias a la implementación de la Evolución Diferencial y el resto de herramientas que se expusieron en la sección 3, Herramientas.

Como medida de la funcionalidad del algoritmo se han obtenido los resultados de varios experimentos sobre las partículas del *Hexón del Adenovirus Humano (1RUX)* y la *bacteriorodopsina (1BRD)*. En ellos se estimaron los ángulos, calculando su diferencia con los reales, se calculó la correlación completa, que recordemos que era el elemento a minimizar en la Evolución Diferencial y las correlaciones entre los pares de proyecciones. Además se obtuvo el tiempo de ejecución del algoritmo en varios esquemas de ejecución.

Experimento 3 Caso 1: Cálculo de los ángulos de tres proyecciones (I)

Nombre de la partícula	Proyecciones	Generaciones	Individuos por generación	Tiempo de proceso
1BRD	3	500	200	43.12 s

Tabla 7: Descripción del primer caso del tercer experimento

En este experimento intentaremos obtener los ángulos de Euler desde los que se tomaron tres proyecciones de la partícula 1BRD. Al ser un número pequeño de proyecciones no son necesarios demasiados individuos por generación. Veremos que este parámetro está especialmente relacionado con el número de proyecciones a resolver, mientras que las generaciones afectan a la cantidad necesaria de iteraciones para resolver el algoritmo. El tiempo de proceso en este caso es de 43 segundos, lo que

no es demasiado teniendo en cuenta las limitaciones del equipo utilizado (*Intel Pentium* 1,4 Ghz con 256 Mib de memoria *RAM*) y la naturaleza de procesamiento intensivo del algoritmo.

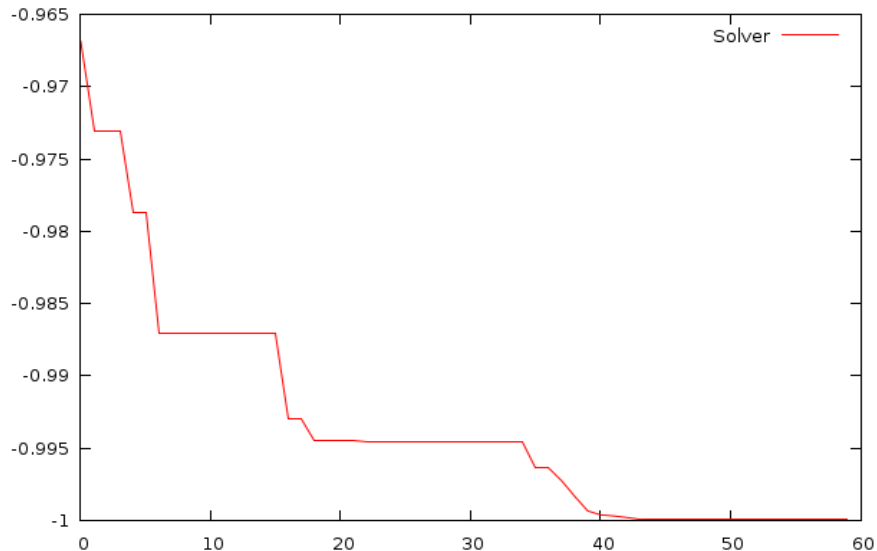


Ilustración 30: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 1

En la Ilustración 30 podemos ver como el algoritmo va acercándose a la solución final buscando entre cada individuo de cada generación. Podemos ver en la gráfica la forma de buscar de la Evolución diferencial, la gráfica tiene puntos en los que da saltos, esto significa que uno de los individuos ha mejorado al anterior, y en las iteraciones sucesivas ese mismo individuo va centrándose en el mínimo al que pertenece hasta que lo alcanza y la gráfica vuelve a ser constante, entonces otro individuo aparece en un mínimo de menor valor y vuelve a aparecer otro salto. El proceso de saltos y búsquedas de mínimos se repite hasta que se da con el individuo que pertenece al mínimo global, ya que cuando se vaya centrando en él no habrá ninguno que pueda ser inferior y por tanto el algoritmo terminará tras varias ejecuciones sin cambios.

Proyección	Ángulos de Euler originales	Ángulos de Euler estimados	Diferencia
1	0	0	0
	0	0	0
	0	0	0
2	272.46	272.13	0.33
	28.09	28.28	-0.19
	270.84	270.25	0.59
3	206.95	206.13	0.82
	129.65	128.68	0.97
	149.25	329.20	-179.95 (0.05)
Error medio (sin contar saltos de 180°)			0.49

Tabla 8: Comparación entre ángulos de Euler originales y estimados

Podemos ver en la Tabla 8 que la diferencia entre los ángulos originales y estimados es mínima, pero existe una diferencia de 180 grados en el tercer ángulo de la tercera proyección. Esto se debe al Problema de la referencia, descrito en la sección 4.1 en la página 51. Dado que una imagen se puede proyectar desde dos ángulos opuestos obteniendo el mismo resultado puede que el algoritmo encuentre la proyección reflejada en lugar de la original, sin perjuicio del volumen que se ha de reconstruir más adelante, ya que los ángulos obtenidos son correctos, pese a que no sean iguales a los originales.

Pareja de proyecciones	Ángulos entre normales originales	Ángulos entre normales estimados	Diferencia
1 y 2	28.09	28.28	-0.19
1 y 3	129.65	128.68	0.97
2 y 3	138.84	111.08	27.76
Error medio			9.64

Tabla 9: Comparación entre los ángulos de las normales de las proyecciones

En la Tabla 9 se han calculado los ángulos entre las normales de las proyecciones estimada y original en el espacio tridimensional para cada pareja de proyecciones. Esta es una buena medida ya que no depende de los ángulos de Euler escogidos, sino sólo de la orientación en el espacio de los planos. Vemos que los

ángulos estimados son cercanos a los originales, excepto el tercero que tiene una diferencia de 30 grados, esto puede deberse al problema de la referencia, explicado anteriormente.

Experimento 3 Caso 2: Cálculo de los ángulos de tres proyecciones (II)

Nombre de la partícula	Proyecciones	Generaciones	Individuos por generación	Tiempo de proceso
1BRD	3	300	300	37.53 s

Tabla 10: Descripción del segundo caso del tercer experimento

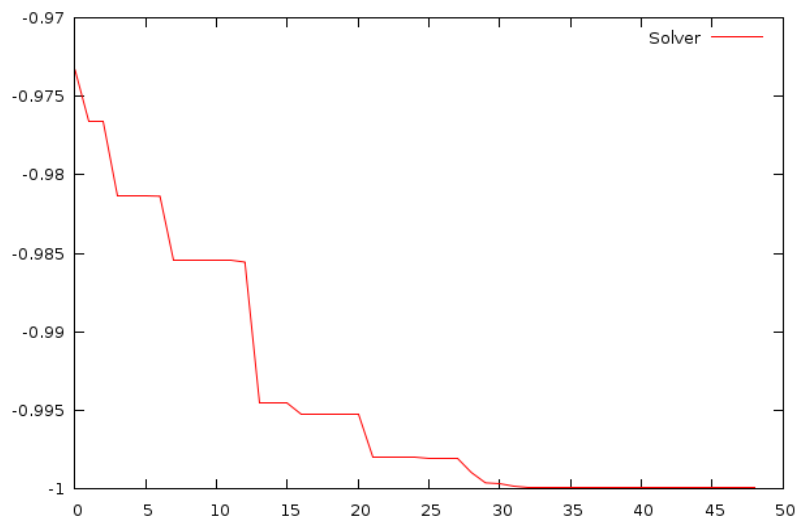


Ilustración 31: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 2

Proyección	Ángulos de Euler originales	Ángulos de Euler estimados	Diferencia
1	0	0	0
	0	0	0
	0	0	0
2	124.33	123.56	0.77
	108.95	104.72	4.23
	245.92	246.16	-0.24
3	249.09	248.10	0.99
	157.36	157.25	0.11
	94.30	94.06	0.24
Error medio (sin contar saltos de 180°)			1.1

Tabla 11: Comparación entre ángulos de Euler originales y estimados

Pareja de proyecciones	Ángulos entre normales originales	Ángulos entre normales estimados	Diferencia
1 y 2	108.95	104.72	4.23
1 y 3	157.36	157.25	0.11
2 y 3	91.17	95.5	-4.33
Error medio			2.89

Tabla 12: Comparación entre los ángulos de las normales de las proyecciones

En la Tabla 12 se pueden ver los ángulos entre las normales de los planos de proyección originales y los estimados de cada pareja. Se ha escogido este parámetro para la comparación por que los ángulos de Euler son dependientes de la referencia que se coja. En los casos estudiados hasta ahora en este tercer experimento se ha utilizado uno de los planos con ángulos (0,0,0) que es igual a los ángulos que tomamos como referencia para la primera proyección, como se vio en la sección 4.1 de Problemática. Si no se utiliza una primera proyección con los ángulos a (0,0,0) los ángulos estimados no se parecerán a los originales, sino que serán completamente diferentes, por lo que no nos servirán como medida de comparación. Esto no quiere decir que la reconstrucción posterior sea errónea, como ya se explicó en la Problemática lo único que ocurrirá será que el volumen estará rotado en el espacio. Este problema se verá en el caso 3, en el que

la única comparación posible será entre los ángulos entre las normales de las proyecciones.

Experimento 3 Caso 3: Cálculo de los ángulos de tres proyecciones (III)

Nombre de la partícula	Proyecciones	Generaciones	Individuos por generación	Tiempo de proceso
1BRD	3	300	1000	111.43 s

Tabla 13: Descripción del tercer caso del tercer experimento

En este caso se tuvo que aumentar el número de individuos por generación para que el algoritmo resolviera el problema satisfactoriamente, lo que produjo una ralentización en la ejecución. Se puede ver que al multiplicar por 3 el número de individuos por generación el tiempo también aumentó en un factor de 3. Esta relación no se mantiene si lo que aumentamos es el número de proyecciones, ya que el número de procesamientos aumenta mucho más.

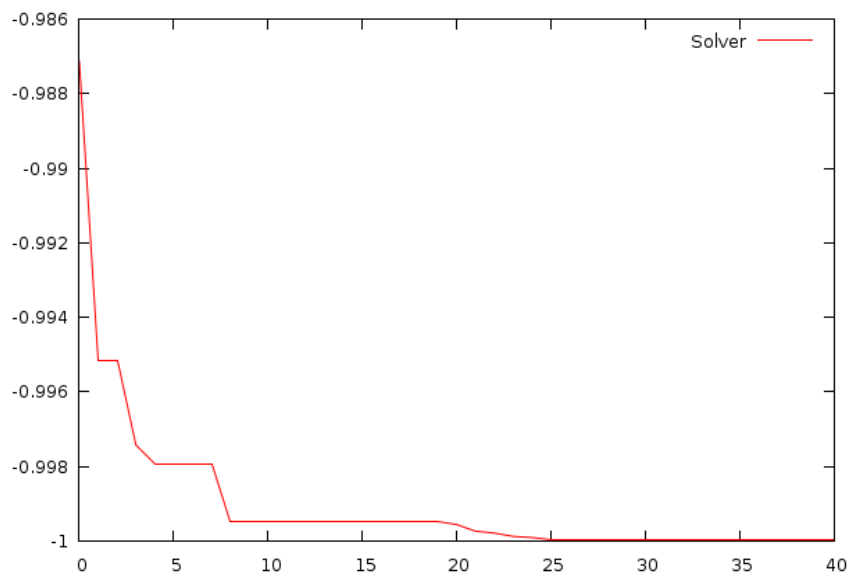


Ilustración 32: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 3

En la Ilustración 32 podemos observar que la correlación se acerca al mínimo más rápido que en los casos anteriores. Esto se debe a que se ha sobrestimado el número de individuos por generación necesarios para la resolución. Probablemente se podrían haber utilizado menos reduciéndose así el tiempo de ejecución.

Proyección	Ángulos de Euler originales	Ángulos de Euler estimados	Diferencia
1	139.72	0	139.72
	120.19	0	120.19
	189.75	0	189.75
2	177.44	199.09	-21.65
	149.83	33.95	115.88
	156.83	99.16	57.67
3	249.1	192.68	56.42
	157.36	61.47	95.89
	94.3	155.57	-61.27
Error medio (sin contar saltos de 180°)			N/A

Tabla 14: Comparación entre ángulos de Euler originales y estimados

Podemos ver que al no utilizar una primera referencia con los ángulos de Euler a (0,0,0) los ángulos estimados no se parecen a los originales. Esto es coherente con la teoría estudiada hasta ahora y no significa que el algoritmo haya fallado al intentar encontrar una solución. Como veremos por los resultados de los ángulos entre las normales la relación entre las tres normales que definen los planos de proyección se ha mantenido, por lo que el volumen que se obtenga de la reconstrucción será similar al buscado, pero estará rotado respecto al original.

Pareja de proyecciones	Ángulos entre normales originales	Ángulos entre normales estimados	Diferencia
1 y 2	36.93	33.95	2.98
1 y 3	64.37	61.47	2.9
2 y 3	27.47	48.12	-20.65
Error medio			8.84

Tabla 15: Comparación entre los ángulos de las normales de las proyecciones

Los ángulos entre las normales de los planos de proyección, como ya se ha visto, coinciden bastante bien con los originales. En este caso concreto el tercero tiene una diferencia importante, mientras que los otros dos se diferencian en tres grados. Probablemente se podría mejorar el tercer ángulo variando algunos parámetros del algoritmo, pero normalmente el error se mantendrá entre $\pm 3^\circ$, esto no es suficiente para una buena reconstrucción, ya que se introduciría un cierto error en el volumen final y dada la naturaleza ruidosa de las tomografías esto no es aceptable, ya que todo el esfuerzo realizado en mantener el proceso libre de errores se perdería. El mejor uso que se le puede dar a este algoritmo es para obtener un primer volumen de referencia con el que iterar con otros algoritmos de resolución de ángulos y reconstrucción 3D. Los resultados nos indican que es una buena primera aproximación rápida en caso de que no sepamos los ángulos de proyección originales.

Experimento 3 Caso 4: Cálculo de los ángulos de cuatro proyecciones (I)

Nombre de la partícula	Proyecciones	Generaciones	Individuos por generación	Tiempo de proceso
1BRD	4	1000	2000	1482.02 s

Tabla 16: Descripción del cuarto caso del tercer experimento

En este caso se introducen cuatro proyecciones en el algoritmo. Para que pueda converger se ha tenido que aumentar el número de individuos por generación sustancialmente, así como el número de generaciones para la resolución, lo que ha impuesto un aumento del tiempo considerable. En la Ilustración 33 podemos ver que estos parámetros son suficientes para que el algoritmo encuentre un mínimo muy cercano a uno.

Se da que por el simple hecho de aumentar el número de proyecciones el tiempo de proceso también aumenta, con lo que nos encontramos que en comparación con el Caso 3, donde se tardaban 111.43 segundos en resolver 300 generaciones de 1000

individuos ahora se tarda 15 veces más para el triple de generaciones con el doble de individuos. Como el aumento de generaciones e individuos es proporcional al aumento del tiempo, si no hubiésemos introducido una proyección más al algoritmo el tiempo habría aumentado en un factor de 6, pero como vemos lo ha hecho en un factor de 15 por lo que al pasar de tres a cuatro proyecciones el tiempo ha aumentado en un factor de 2'5, que no es proporcional. Más adelante se verá que al añadir más proyecciones el factor de aumento describe una función exponencial, por lo que el número de proyecciones a resolver será un parámetro crítico del algoritmo.

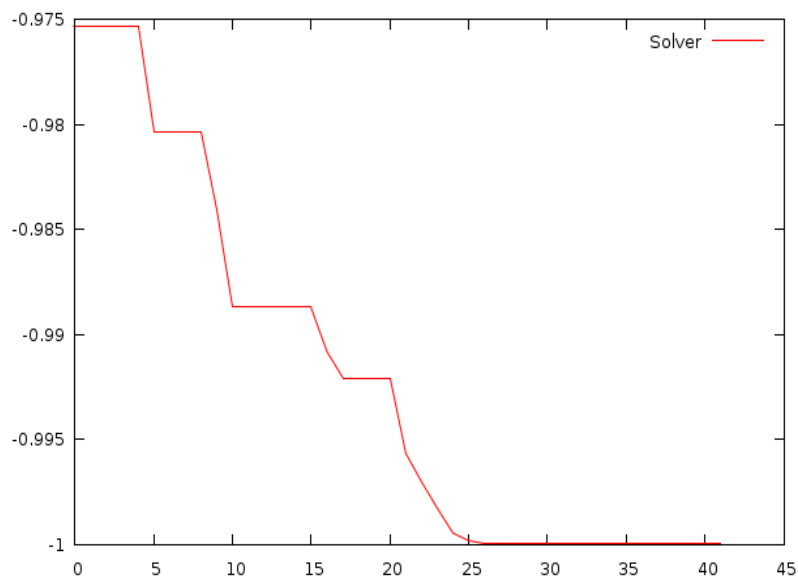


Ilustración 33: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 4

Proyección	Ángulos de Euler originales	Ángulos de Euler estimados	Diferencia
1	0	0	0
	0	0	0
	0	0	0
2	276.91	276.53	0.38
	61.76	61.21	-0.19
	237.94	238.07	-180.02 (-0.02)
3	26.69	26.05	0.64
	107.86	108.05	-0.19
	22.25	202.27	-180.02 (-0.02)
4	249.1	248.03	1.07
	157.36	157.19	0.17
	94.30	274.22	-179.92 (0.08)
Error medio (sin contar saltos de 180°)			0.31

Tabla 17: Comparación entre ángulos de Euler originales y estimados

En los datos de la Tabla 17 podemos ver que con la primera proyección tomada desde el ángulo de Euler (0,0,0) los ángulos obtenidos para las otras proyecciones son similares. Todos ellos tienen diferencia de 180° en su último ángulo. Esto se debe al problema de la proyección reflejada, ya estudiado, pero aplicado a la triangulación de tres proyecciones. Es probable que el volumen reconstruido sea muy parecido al real si solo fuésemos a utilizar estas tres proyecciones, pero al añadir más puede que en algún caso no encontrase la manera de encajar alguna de las nuevas proyecciones. Lo más probable es que se haya encontrado un mínimo de la correlación en unas posiciones simétricas de las tres proyecciones que guardan relación con los ángulos buscados pero no son realmente los buscados.

Pareja de proyecciones	Ángulos entre normales originales	Ángulos entre normales estimados	Diferencia
1 y 2	61.77	61.21	0.56
1 y 3	107.87	108.05	-0.18
1 y 4	157.36	157.19	0.17
2 y 3	145.71	58.22	87.49
2 y 4	135.21	99.78	35.43
3 y 4	66.67	66.43	0.24
Error medio			20.68

Tabla 18: Comparación entre los ángulos de las normales de las proyecciones

Como hemos deducido de la Tabla 17 y podemos corroborar en la 18 la ejecución del algoritmo no ha sido satisfactoria. Se ha encontrado un mínimo parecido al esperado, pero no es el exacto. Se comprueba la teoría de que se ha encontrado algún mínimo en una posición con las proyecciones reflejadas porque cuatro de seis ángulos entre normales son correctos (las proyecciones están colocadas correctamente entre sí) mientras que el grupo que forman entre la dos, la tres y la cuatro tiene algunas diferencias de ángulos significativas como son 90° y 35° . Se puede concluir que el algoritmo no ha encontrado las posiciones reales pero sí unas relacionadas, por lo que habría que volver a ejecutar el algoritmo cambiando diferentes parámetros como el número de generaciones o los individuos por generación.

Experimento 3 Caso 5: Cálculo de los ángulos de cuatro proyecciones (II)

Nombre de la partícula	Proyecciones	Generaciones	Individuos por generación	Tiempo de proceso
1BRD	4	1000	2000	1449.66 s

Tabla 19: Descripción del quinto caso del tercer experimento

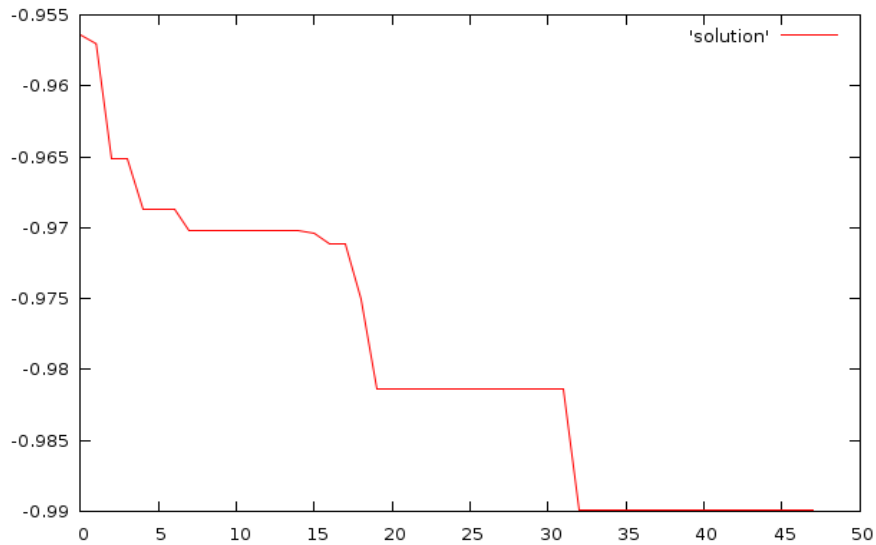


Ilustración 34: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 5

Proyección	Ángulos de Euler originales	Ángulos de Euler estimados	Diferencia
1	0	0	0
	0	0	0
	0	0	0
2	139.72	138.01	1.71
	120.19	137.51	-17.32
	189.76	186.24	3.52
3	206.95	206.1	0.85
	129.65	143.94	-14.29
	149.25	154.91	-5.66
4	177.45	190.43	-12.98
	149.83	161.46	-11.63
	156.83	170.85	-14.02
Error medio (sin contar saltos de 180°)			9.11

Tabla 20: Comparación entre ángulos de Euler originales y estimados

La Tabla 20 nos muestra que los ángulos estimados en este caso tienen diferencias con los originales de no más de 20°, esto nos hace pensar que el algoritmo

en este caso está sobre el mínimo global de la función y no sobre un mínimo local. Sin embargo, tenemos unas diferencias mayores que en casos anteriores, lo

que quiere decir que, aunque estemos sobre el mínimo global, no nos hemos centrado completamente en él, así que probablemente necesitemos algunas generaciones más para obtener ángulos más precisos. De hecho, observando atentamente la Ilustración 34 se observa que el mínimo al que llega el algoritmo no es tan cercano a 1 como en los casos anteriores. El índice de correlación al que se llega con la Evolución Diferencial es -0.989881 y cuando aplicamos el método de Powell (Sección 3.3.4 pág. 41) nos acercamos hasta -0.993474, que es bastante cercano a -1, pero estamos lejos de los valores del orden de -0.9999 de otros casos. Esto refuerza la teoría de que nos harían falta más generaciones para llegar a órdenes en los que los ángulos no se diferenciaban en más de 1 o 2 grados.

Pareja de proyecciones	Ángulos entre normales originales	Ángulos entre normales estimados	Diferencia
1 y 2	120.19	137.51	-17.32
1 y 3	129.65	143.94	-14.29
1 y 4	149.83	161.46	-11.63
2 y 3	34.21	20.66	13.55
2 y 4	36.93	25.01	11.92
3 y 4	20.73	18.84	1.86
Error medio			11,76

Tabla 21: Comparación entre los ángulos de las normales de las proyecciones

De nuevo vemos que los ángulos entre normales son muy parecidos a los originales pero existe mayor diferencias que en otros casos.

Experimento 3 Caso 6:

Nombre de la partícula	Proyecciones	Generaciones	Individuos por generación	Tiempo de proceso
1BRD	5	2000	3000	8823.37 s

Tabla 22: Descripción del sexto caso del tercer experimento

En este último caso queremos comprobar que pasaría si seguimos aumentando el número de proyecciones. Dada la naturaleza intensiva del cálculo del algoritmo esto sólo se ha podido hacer para un caso, en el que se han tenido que poner como parámetros 2000 generaciones de 3000 individuos cada una. El proceso ha tardado 8823.37 segundos, cerca de las dos horas y media de ejecución, lo que, teniendo en cuenta la máquina utilizada (descrita en el caso 1), no parece demasiado costosa. Los resultados nos indican que con cinco y probablemente más proyecciones en el mismo proceso de minimización es posible encontrar una solución aproximada siempre y cuando utilicemos un número suficiente de generaciones e individuos por generación.

En la Ilustración 35 (abajo), podemos ver que se ha alcanzado un valor satisfactorio para la variable a minimizar, del orden de -0.9999. Cuando se alcanzan estos órdenes de magnitud suele significar que estamos sobre el mínimo buscado a no ser que existan otros mínimos tan bajos o simetrías, lo primero es poco probable y de lo segundo tenemos conocimiento a priori y, en este caso, por simplicidad se ha escogido una partícula sin simetrías.

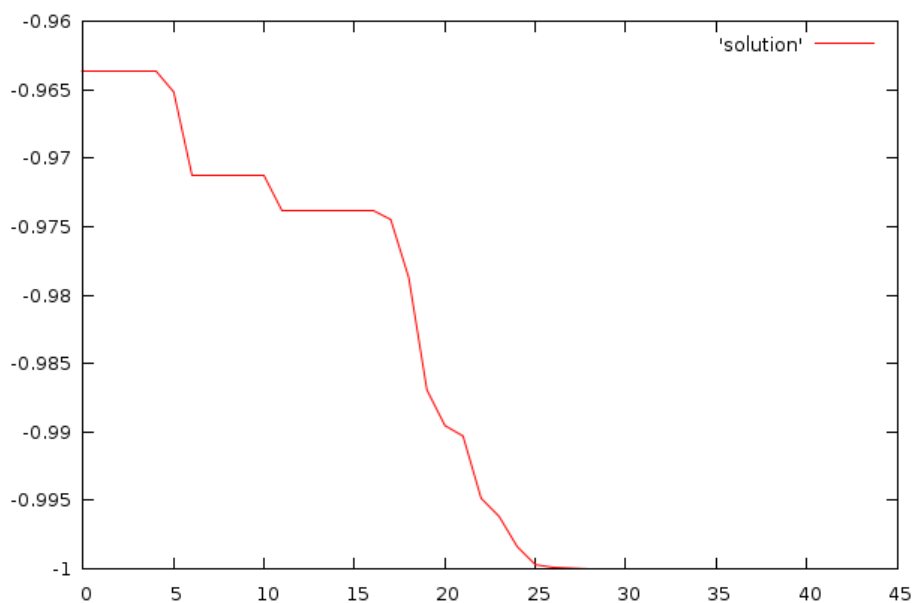


Ilustración 35: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 6

Proyección	Ángulos de Euler originales	Ángulos de Euler estimados	Diferencia
1	0	0	0
	0	0	0
	0	0	0
2	329.77	329.47	0.3
	61.4	59.76	1.64
	99.69	280.14	-180.45 (-0.45)
3	139.72	139.76	-0.04
	120.19	121.66	-1.47
	189.76	190.11	-0.35
4	177.45	175.6	1.85
	149.83	149.55	0.28
	156.83	336.48	-179.65 (-0.35)
5	249.1	248.72	0.38
	157.36	157.42	-0.06
	94.3	274.94	-180.64 (-0.64)
Error medio (sin contar saltos de 180°)			0.65

Tabla 23: Comparación entre ángulos de Euler originales y estimados

Volvemos a encontrar en la comparación entre los ángulos de Euler originales y los estimados saltos de 180 grados en los últimos ángulos de tres de los cuatro ángulos de Euler a estimar. Conociendo lo expuesto sobre simetría anteriormente podemos obviar esos saltos encontrándonos con que la media del error está en 0.65° y ninguna diferencia supera los 2° . Estos resultados, junto con los de la Tabla 24, donde se resalta que los ángulos entre todas las combinaciones son muy parecidos a los originales y, por tanto forman un sistema en el que no queda lugar a dudas de que hemos encontrado la solución al problema. Como siempre, estos resultados no serán tan buenos como los obtenidos por un proceso de asignación de ángulos y reconstrucciones sucesivas, pero es una buena aproximación por donde empezar a buscar. Incluso en el caso de que no nos valiesen los saltos de 180° por no resultar una reconstrucción adecuada podemos, simplemente, buscar en saltos de esa cantidad ángulos que resulten en una reconstrucción que dé mejores resultados.

Pareja de proyecciones	Ángulos entre normales originales	Ángulos entre normales estimados	Diferencia
1 y 2	61.4	59.76	1.64
1 y 3	120.19	121.66	-1.47
1 y 4	149.83	149.55	0.28
1 y 5	157.36	157.42	-0.06
2 y 3	103.98	105.35	-1.37
2 y 4	100.05	101.04	1.86
2 y 5	96.05	97.74	-1.69
3 y 4	36.93	34.65	2.28
3 y 5	64.37	59.06	5.31
4 y 5	27.48	27.29	0.19
Error medio			1.615

Tabla 24: Comparación entre los ángulos de las normales de las proyecciones

6 Conclusiones

Desde el principio este proyecto se planteó para encontrar una solución a un problema real en un campo de la bioingeniería aplicada. Esta búsqueda debía incluir una fase de investigación del estado de la técnica para conocer que otras soluciones habían propuesto otros ingenieros al enfrentarse al mismo problema. Pese a que la asignación de ángulos de Euler sin referencia para el proceso de reconstrucción ya tenía algunas soluciones basadas en geometría, éstas no eran capaces de resolver más de un conjunto de tres proyecciones a la vez, lo cual era insuficiente y dejaba mucho margen de error, puesto que cuanta más información se utilice en un proceso más aproximado será su resultado final. Además, al utilizar sólo tres proyecciones al azar, si éstas no formaban un sistema en el espacio tridimensional toda la asignación era errónea.

Para solventar todos los problemas de las soluciones actuales se consideró indispensable que el algoritmo resolviese más de tres proyecciones a la vez y, a ser posible, que los resultados fuesen razonablemente buenos independientemente del número de proyecciones introducidas, aunque el coste en tiempo de proceso fuese lógicamente superior. Esto suponía un gran reto ya que por la naturaleza del problema un aumento de una proyección suponía un aumento de tres dimensiones del espacio en el que se fijaban las soluciones, haciéndose necesario encontrar una fórmula para disminuir en lo posible los grados de libertad de los ángulos de Euler a estimar mediante el establecimiento de relaciones entre las diferentes proyecciones. Estas relaciones se obtuvieron a partir del Teorema de la Sección Central y sus variantes que nos indicaba que las transformadas de Fourier de dos proyecciones de un mismo volumen comparten una línea que pasa por el centro de ambas. Así, cada proyección añadida se podía fijar respecto a las demás hasta que todas encajasen. Explicado de forma más sencilla el algoritmo debía rotar todas las transformadas de Fourier de las proyecciones introducidas hasta encontrar la posición en la que todas las líneas de las intersecciones entre ellas fuesen lo más parecidas posibles.

Tras muchos desarrollos y pruebas se llegó a la conclusión de que era preferible utilizar la variante del Teorema de la Línea común del espacio real, relacionando las proyecciones 1D en el espacio real en vez de las líneas en el espacio de Fourier. La correspondencia se puede ver en la sección 3.3.3, en la Ilustración 6. El resultado era mejor y no había que procesar las transformadas de Fourier, pero esto implicaba a su vez la necesidad de realizar proyecciones 1D eficientemente. Se implementó un preproceso que calculaba la transformada Radon (sección 3.3.5) y así se consiguió mayor velocidad.

Con toda la investigación realizada el problema se había reducido a un problema de minimización con unas variables de entrada que se movían en un espacio multidimensional. Se optó por resolverlo utilizando la Evolución diferencial (sección 3.3.4 pág. 43) que había demostrado ser útil en procesos de similar naturaleza. Esto implicó el estudio de este algoritmo y la adaptación del desarrollo realizado a las restricciones que imponía. Finalmente se comprobó que las expectativas puestas en él fueron acertadas dando los primeros resultados aceptables.

A partir de este punto se comenzaron a obtener resultados pero no tenían la fiabilidad requerida por lo que hubo que volver al principio planteando de nuevo todos los problemas sus soluciones y comprobando los desarrollos. Tras muchos cambios menores que mejoraron el algoritmo y otros que se descartaron se llegó al desarrollo final con el que se han realizado las pruebas del apartado 5. En ellas, dentro de las posibilidades del hardware de que se disponía, se intentó forzar al algoritmo desarrollado para resolver problemas de mayor envergadura comportándose razonablemente bien en todos los casos y confirmando las teorías utilizadas y las decisiones de diseño tomadas.

Más allá de este Proyecto Fin de Carrera es posible que este algoritmo se presente como una opción de las existentes en el paquete Xmipp con posibles

desarrollos y mejoras posteriores que no entraban dentro del ámbito de este estudio. Realmente se ha conseguido un avance respecto a lo ya existente y supone una alternativa a los procesos actuales permitiendo la estimación relativamente rápida de los ángulos de proyección sin volumen de referencia lo que para estudios reales de partículas supone disponer de una colocación de los ángulos iniciales mucho mejor de lo que se conseguía antes.

Referencias

- [1] Gao et al., 2002; Huang et al., 2003; Mindell and Grigorieff, 2003; Saad et al., 2001; Zhou et al., 1996
- [2] M. van Heel, “Multivariate statistical classification of noisy images (randomly oriented biological macromolecules),” *Ultramicroscopy*, vol. 13, pp. 165–183, 1984.
- [3] A. Pascual-Montano, L. E. Donate, M. Valle, M. Bárcena, R. D. Pascual-Marqui, and J. M. Carazo, “A novel neural network technique for analysis and classification of M single-particle images,” *J. Struct. Biol.*, vol. 133, pp. 233–245, 2001.
- [4] R.A. Crowther, L.A. Amos, J.T. Finch, D.J. DeRosier and A. Klug, *Nature* 226 (1970) 421.
- [5] R.A. Crowther, *Phil. Trans. Roy. Soc. London B*261 (1971) 221.
- [6] J. Frank, *Three Dimensional Electron Microscopy of Macromolecular Assemblies*, Oxford University Press, 2005.
- [7] Harauz and Van Heel, 1986b
- [8] Radermacher, 1998
- [9] Sorzano, 2004d
- [10] Linux en la Wikipedia: <http://es.wikipedia.org/wiki/Linux>
- [11] Página del proyecto gcc: <http://gcc.gnu.org/>

- [12] Página del proyecto make: <http://www.gnu.org/software/make/>
- [13] Página del proyecto em2em: <http://www.imagescience.de/em2em/>
- [14] J. Bernard Heymann. David M. Belnap. 2007. *Bsoft: Image processing and molecular modeling for electron microscopy*. *Journal of Structural Biology*. 157. 1. 3 - 18.
- [15] Página de *Differential Evolution*: <http://www.icsi.berkeley.edu/~storn/code.html>
- [16] Pascual-Marqui RD. Pascual-Montano AD. Kochi K. Carazo JM. Smoothly Distributed Fuzzy c-Means: a New Self-Organizing Map. *Pattern Recognition* 34:2395-2402, 2001
- [17] Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. *Biol. Cybernet.* 43, 59-69.
- [18] Marabini, R., Carazo, J.M., 1994. Pattern recognition and classification of images of biological macromolecules using artificial neural networks. *Biophys. J.* 66, 1804–1814.
- [19] Van Heel, M., Frank, J., 1981. Use of multivariate statistical statistics in analysing the images of biological macromolecules. *Ultramicroscopy* 6, 187–194.
- [20] Sammon, J.W., 1969. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.* 18, 401–409.
- [21] Carazo, J.M., Rivera, F.F., Zapata, E.L., Radermacher, M., Frank, J., 1990. Fuzzy sets-based classification of electron microscopy images of biological macromolecules with an application to ribosomal particles. *J. Microsc.* 157, 187–203.

- [22] de Alarcón, P.A., Pascual-Montano, A., Gupta, A., Carazo, J.M., 2002. Modeling shape and topology of low-resolution density maps of biological macromolecules. *Biophys. J.* 83, 619–632.
- [23] Pascual-Montano, A., Donate, L.E., Valle, M., Bárcena, M., Pascual-Marqui, R.D., Carazo, J.M., 2001. A novel neural network technique for analysis and classification of EM single-particle images. *J. Struct. Biol.* 133, 233–245.
- [24] Pascual, A., Bárcena, M., Merelo, J.J., Carazo, J.M., 2000. Mapping and fuzzy classification of macromolecular images using self organizing neural networks. *Ultramicroscopy* 84, 85–99.
- [25] Frank, J., 1996. *Three Dimensional Electron Microscopy of Macromolecular Assemblies.* Academic Press, San Diego, CA.
- [26] Sorzano, C.O.S., Marabini, R., Herman, G.T., Censor, Y., Carazo, J.M., 2004b. Transfer function restoration in 3D electron microscopy via iterative data refinement. *Phys. Med. Biol.* 49, 509–522.
- [27] Zubelli, J.P., Marabini, R., Sorzano, C.O.S., Herman, G.T., 2003. Three-dimensional reconstruction by Chahine's method from electron microscopic projections corrupted by instrumental aberrations. *Inverse Probl.* 19, 933–949.
- [28] Lewitt, R.M., 1990. Multidimensional digital image representations using generalized Kaiser-Bessel window functions. *J. Opt. Soc. Am.A* 7, 1834–1846.
- [29] Lewitt, R.M., 1992. Alternatives to voxels for image representation in iterative reconstruction algorithms. *Phys. Med. Biol.* 37, 705–716.
- [30] Herman, G.T., 1980. *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography.* Academic Press, New York.

- [31] Gilbert, P., 1972. Iterative methods for the three-dimensional reconstruction of an object from projections. *J. Theor. Biol.* 36, 105–117.
- [32] Andersen, A., Kak, A., 1984. Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm. *Ultrasonic Imag.* 6, 81–94.
- [33] Censor, Y., Gordon, D., Gordon, R., 2001a. BICAV: A block-iterative parallel algorithm for sparse systems with pixel-related weighting. *IEEE Trans. Med. Imag.* 20, 1050–1060.
- [34] Censor, Y., Gordon, D., Gordon, R., 2001b. Component averaging: an efficient iterative parallel algorithm for large and sparse unstructured problems. *Parallel computing* 27, 777–808.
- [35] Censor, Y., Elfving, T., Herman, G., 2000. Averaging strings of sequential iterations for convex feasibility problems. In: Butnariu, D., Censor, Y., Reich, S. (Eds.), *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*. Elsevier Science, Amsterdam, pp. 101–114.
- [36] Area, E., Martín-Benito, J., Gastaminza, P., Torreira, E., Valpuesta, J., Carrascosa, J., Ortín, J., 2004. 3D structure of the influenza virus polymerase complex: localization of subunit domains. *Proc. Natl. Acad. Sci. USA* 101, 308–313.
- [37] Bárcena, M., Donate, L., Ruiz, T., Dixon, N., Radermacher, M., Carazo, J.M., 2001. The DnaB–DnaC complex: a structure based on interactions among assymmetric dimers. *EMBO J.* 20, 1462–1468.
- [38] "Gramphics Gems IV", Paul S. Heckbert, ed., Academic Press Professional, Toronto, 1994

- [39] Introduction to Robotics. Mechanics and Control. Craig, J.J. (1989).Addison-Wesley Publising Company
- [40] Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo , Jesús Ramón Pérez López(2005).
- [41] R. Fletcher, “Practical methods of optimization”, John Wiley & Sons, New York, 1987.
- [42] J.A. Nelder, R. Mead, “A simplex method for function minimization”, The Computer Journal, Vol. 7, No. 4, January 1965, pp. 308-313.
- [43] <http://www.nist.gov/dads/HTML/divideconqr.html>
- [44] F. Glover, “Future paths for integer programming and links to artificial intelligence”, Computer and Operations Research, Vol.13, No. 5, May 1986.
- [45] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, “Optimization by simulated annealing”, Science, Vol. 220, No. 4598, May 1983, pp. 671-680.
- [46] D.E. Goldberg, “Genetic algorithms in search, optimization, and machine learning”, Addison-Wesley, New York, 1989.
- [47] J. Kennedy, R.C. Eberhart, “Particle swarm optimization”, Proceedings of the IEEE International Conference on Neural Networks-ICNN’95, Perth (Australia), December 1995, Vol.4, pp. 1942-1948.
- [48] Lueder, E. (1990), Optimization of Circuits with a Large Number of Parameters, Archiv fuer Elektronik und Uebertragungstechnik, Band 44, Heft 2, pp 131 – 138.
- [49] Moebus, D. (1990), Algorithmen zur Optimierung von Schaltungen und zur Loesung nichtlinearer Differentialgleichungen, Dissertation am Institut fuer Netzwerk- und Systemtheorie der Univ. Stuttgart.

[50] Brayton, H., Hachtel, G. and Sangiovanni-Vincentelli, A. (1981), A Survey of Optimization Techniques for Integrated Circuit Design, Proceedings of the IEEE 69, pp. 1334 – 1362.

[51] Artículo en la Wikipedia sobre adenovirus: <http://es.wikipedia.org/wiki/Adenovirus>

[52] Artículo en la Wikipedia sobre la bacteriorodopsina: <http://es.wikipedia.org/wiki/Bacteriorodopsina>

Índice de ilustraciones

Ilustración 1: Intersección de planos en tres dimensiones.....	16
Ilustración 2: Zona de intersección de planos con grosor 1/D.....	18
Ilustración 3: Arquitectura de Xmipp.....	21
Ilustración 4: Proceso gráfico de la rotación por ángulos de Euler.....	33
Ilustración 5: Líneas Comunes.....	36
Ilustración 6: Teorema sección central: Correspondencia de espacios.....	37
Ilustración 7: Sistema General.....	38
Ilustración 8: Función objetivo.....	39
Ilustración 9: Greedy Criterion problema de los mínimos locales.....	39
Ilustración 10: Algoritmo evolutivo.....	40
Ilustración 11: Simulated Annealing.....	40
Ilustración 12: Clasificación de los métodos de optimización más relevantes.....	41
Ilustración 13: Transformada Radon por rotación.....	48
Ilustración 14: Problema de la referencia 1.....	51
Ilustración 15: Problema de la referencia 2.....	52
Ilustración 16: Algoritmo general.....	54
Ilustración 17: Comparación línea común y proyección común.....	56
Ilustración 18: Cálculo del ángulo 3D.....	59
Ilustración 19: Comparación de ángulos 3D.....	60
Ilustración 20: Problema de la proyección reflejada.....	62
Ilustración 21: Vistas del volumen del hexón del adenovirus humano 5.....	64
Ilustración 22: Vistas del volumen de las hélices alfa de la bacteriorodopsina.....	66
Ilustración 23: Comparación de las proyecciones 1D comunes para la partícula 1BRD.....	67
Ilustración 24: Proyecciones 1D comunes entre dos proyecciones de partículas distintas	68
Ilustración 25: Proyecciones 1D comunes entre dos proyecciones de la partícula 1RUX	69
Ilustración 26: Comparación de la covarianza 1BRD.....	71
Ilustración 27: Alineación 2D en base a líneas comunes.....	72

Ilustración 28: Comparación de la correlación 1BRD y 1RUX.....	73
Ilustración 29: Comparación de la covarianza 1RUX.....	74
Ilustración 30: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 1.....	76
Ilustración 31: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 2.....	78
Ilustración 32: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 3.....	80
Ilustración 33: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 4.....	83
Ilustración 34: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 5.....	86
Ilustración 35: Mínima correlación negativa alcanzada por el algoritmo en cada iteración del caso 6.....	89

Índice de tablas

Tabla 1: Resultados del primer caso del primer experimento.....	67
Tabla 2: Resultados del segundo caso del primer experimento.....	68
Tabla 3: Resultados del segundo caso del primer experimento.....	69
Tabla 4: Resultados del primer caso del segundo experimento.....	71
Tabla 5: Resultados del segundo caso del segundo experimento.....	73
Tabla 6: Resultados del tercer caso del segundo experimento.....	74
Tabla 7: Descripción del primer caso del tercer experimento.....	75
Tabla 8: Comparación entre ángulos de Euler originales y estimados.....	77
Tabla 9: Comparación entre los ángulos de las normales de las proyecciones.....	77
Tabla 10: Descripción del segundo caso del tercer experimento.....	78
Tabla 11: Comparación entre ángulos de Euler originales y estimados.....	79
Tabla 12: Comparación entre los ángulos de las normales de las proyecciones.....	79
Tabla 13: Descripción del tercer caso del tercer experimento.....	80
Tabla 14: Comparación entre ángulos de Euler originales y estimados.....	81
Tabla 15: Comparación entre los ángulos de las normales de las proyecciones.....	81
Tabla 16: Descripción del cuarto caso del tercer experimento.....	82
Tabla 17: Comparación entre ángulos de Euler originales y estimados.....	84
Tabla 18: Comparación entre los ángulos de las normales de las proyecciones.....	85
Tabla 19: Descripción del quinto caso del tercer experimento.....	85
Tabla 20: Comparación entre ángulos de Euler originales y estimados.....	86
Tabla 21: Comparación entre los ángulos de las normales de las proyecciones.....	87
Tabla 22: Descripción del sexto caso del tercer experimento.....	88
Tabla 23: Comparación entre ángulos de Euler originales y estimados.....	89
Tabla 24: Comparación entre los ángulos de las normales de las proyecciones.....	90