

Práctica Final

11763 - Procesamiento de Imagenes Médicas

Martí Gelabert Gómez

June 2023

Contents

1	Introduction	1
2	Ejercicio 1	2
2.1	Objetivo	2
2.2	Procedimiento	2
2.3	Resultados	2
2.4	Carencias del Algoritmo	2
3	Ejercicio 2	3
3.1	Objetivo	3
3.2	Procedimiento	3
3.2.1	Proporciones de CT	3
3.2.2	LandMarks	4
3.2.3	Obtención de Parámetros de Transformación	5
3.2.4	Segmentación	6
3.2.5	Resultados de la transformacion	7
4	Conclusiones	8

1 Introduction

En este proyecto se nos han pedido dos tareas. En la primera, debemos emplear la información de ficheros DICOM (en mi caso del paciente 5_*HCC*) para mostrar diferentes proyeccion del paciente. Y en la segunda debíamos aplicar *3D Rigid Coregistration* sobre otros ficheros DICOM de otro paciente. La estructura de este documentación estará divida en dos apartado, uno para cada uno de los ejercicios.

2 Ejercicio 1

2.1 Objetivo

El objetivo de este ejercicio consistió en utilizar los formatos DICOM vistos en clase para realizar diversas proyecciones y transformaciones en las imágenes de un paciente. El resultado deseado fue obtener un gif de proyección de intensidad máxima en los planos coronal y sagital, además de realizar la segmentación de ciertos volúmenes del paciente y proyectarlos en una imagen aparte.

2.2 Procedimiento

Primero, he obtenido los datos del paciente desde la base de datos de los ficheros DICOM. En este caso, seleccionamos los ficheros correspondiente a la fase 3, debido a que nuestro fichero de segmentación corresponden con estos. Una vez obtenidos los ficheros podemos obtener los *pixel_array* de cada uno de ellos y reorganizarlos utilizando el número de *slice*, de esta manera podemos empleando *np.stack* o una lista para formar un array de 3 dimensiones.

Después de esto, he extraído la segmentación. La forma en la que se guardan los elementos en este tipo de DICOM es un tanto peculiar. En este caso, hay *items* de segmentaciones que corresponden a cada una de las (*slices*) individualmente con información como el *tag* de elemento que segmenta y la posición del paciente. Gracias a esta última, podemos ordenar nuestra segmentación y separarla en diferentes tipos y generar mascarar para cada uno de los objetos.

Una vez con las máscaras separadas y las imágenes de la CT juntas, solo solo he aplicado las transformaciones que vimos en clase para realizar las proyecciones. En mi caso, también apliqué estas transformaciones a las máscaras de segmentación y ajusté el valor alfa para dar un efecto de opacidad.

2.3 Resultados

El output se encuentra dentro de la carpeta de resultados, donde mostramos el gif final de la ejecución y lo guardamos. En la Figura 1 podemos ver una de las proyecciones intermedias.

2.4 Carencias del Algoritmo

Esta claro que el algoritmo funciona con todos los elementos hardcoded, y no se ha pensado para un uso general del mismo, a demás de que la forma en la que se enseñan las mascarar sobre la proyección podrían haber sido más refinadas, ya que el uso del colormaps no es del todo correcto y probablemente hay formas mejores.

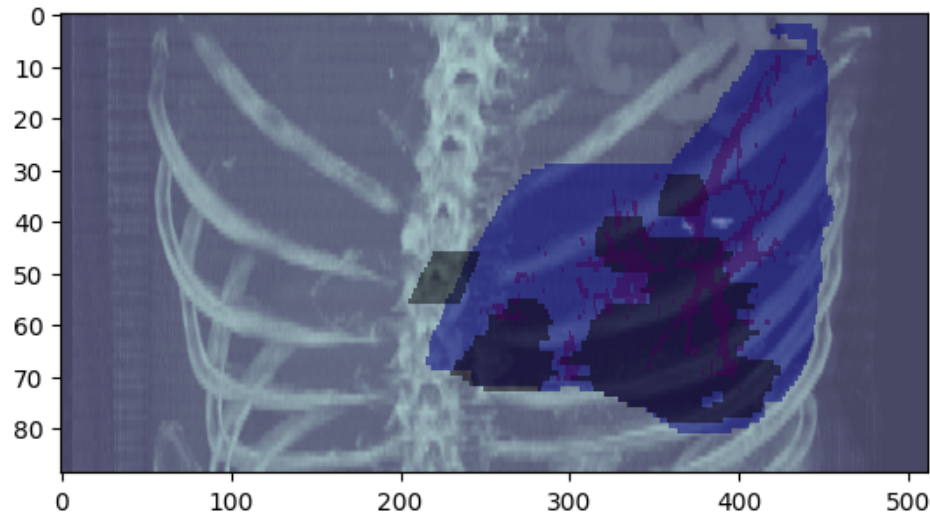


Figure 1: Una de las proyecciones de nuestro algoritmo donde se muestra la proyección y las 4 zonas segmentadas.

3 Ejercicio 2

3.1 Objetivo

El segundo ejercicio ha resultado ser considerablemente más desafiante que el primero. En este caso, se requería llevar a cabo un proceso de co-registración entre los resultados de una CT y otras imágenes, en este caso el *phantom* y el atlas, que son nuestras imágenes "ideales". Además de obtener una segmentación de la zona de la amígdala.

3.2 Procedimiento

Para este ejercicio he repetido parte de los procesos de anterior con respecto a la obtención de la información del paciente, pero en este caso además he tenido que obtener los datos del *phantom* y del atlas. Estos se van a emplear para saber como transformar nuestras imágenes originales, para obtener una imagen parecida a las otras.

3.2.1 Proporciones de CT

Las proporciones de la CT se han ajustado manualmente para obtener una forma parecida a la del *phantom* y a la del atlas, que como se pueden ver en la Figura 2. Por lo que se han aplicado diferentes transformaciones como crop y zoom manual para obtener algo más parecido al resto, como podemos ver en Figura 3

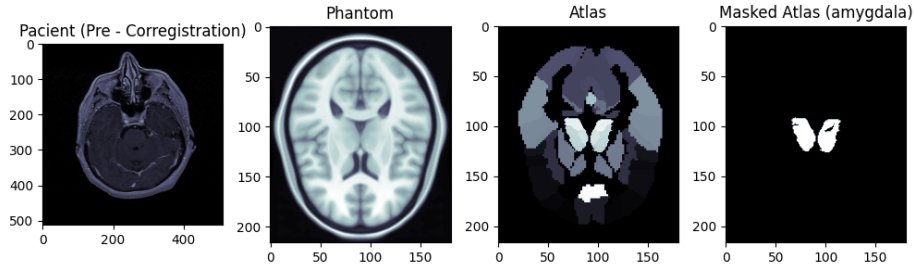


Figure 2: Diferencia entre las proporciones de la imagen de la CT y el resto.

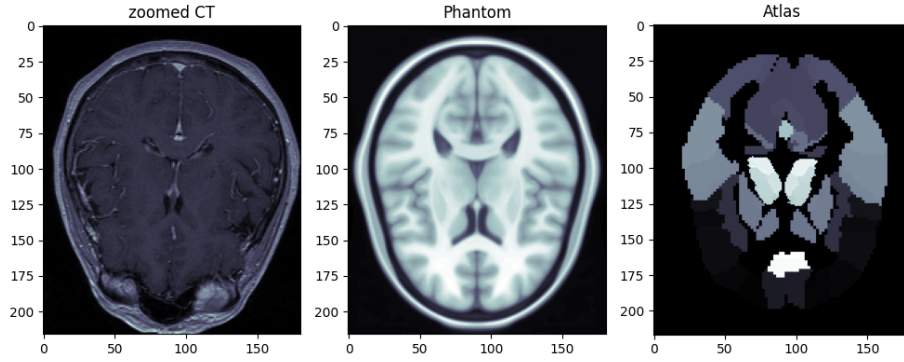


Figure 3: Imagen de la CT ajustada.

3.2.2 LandMarks

Para la obtención de los *landmarks*, simplemente con unos ajustes a las proporciones iniciales de los arrays y una transformación para obtener coordenadas tipo "voxel" nos han permitido obtener cientos de ellos. Es cierto que de esta manera podemos ser poco precisos, pero así nos ahorramos tener que ir enmiéndolos manualmente.

A la hora de calcular los parámetros de la transformación ideal de nuestra imagen original a nuestra imagen de referencia no hemos podido emplearlos todos, debido a que requieren un coste computacional muy alto (como se puede ver en la Figura 4). Por eso, se han escogido como *landmarks* los elementos de cada 15 "voxels", de esta manera podemos tener una idea general de la transformación que requerimos, y la podremos aplicar en nuestra imagen final. Es cierto que de esta manera estamos perdiendo precisión, pero a cambio obtener un mejor coste computacional.

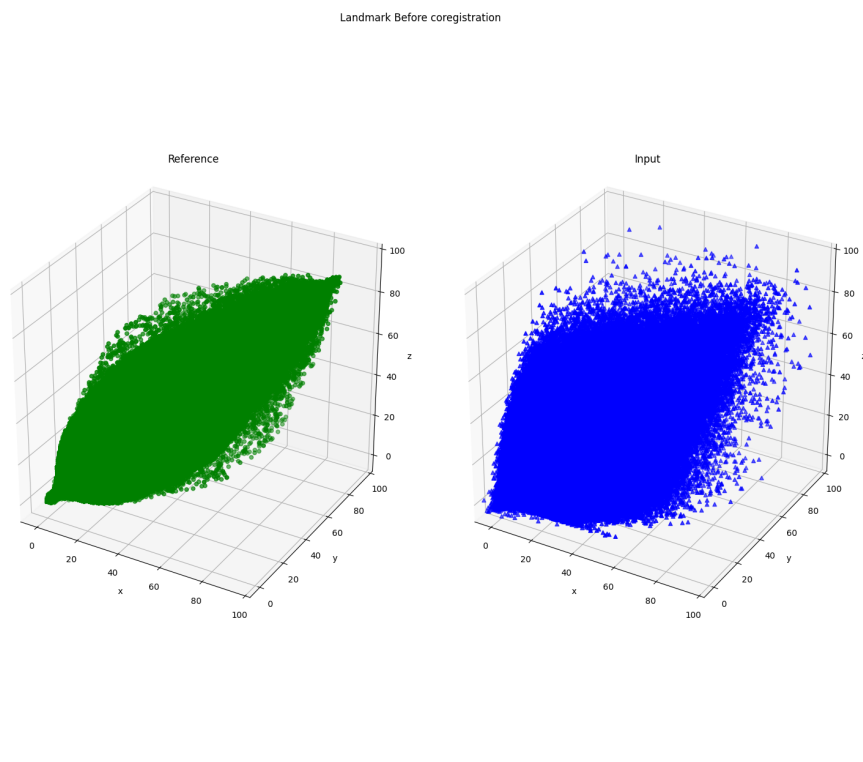


Figure 4: Todos los landmarks sin tratar.

3.2.3 Obtención de Parámetros de Transformación

Básicamente, a la hora de comparar los landmarks y realizar operaciones con ellos, se ha hecho con las funciones vistas en clase, donde se comparan entre ellos y en función de su distancia se van ajustando los parámetros de transformación y rotación. Para ello se ha empleado el mismo optimizador que hemos visto en clase el *least_squares*, quizá algo lento para los casos con muchos landmarks, pero si se reduce el número a cambio de precisión puede ser un optimizador ideal.

Después de esto podemos aplicar estas transformaciones a nuestras imágenes de CT y obtendríamos lo que estamos buscando.

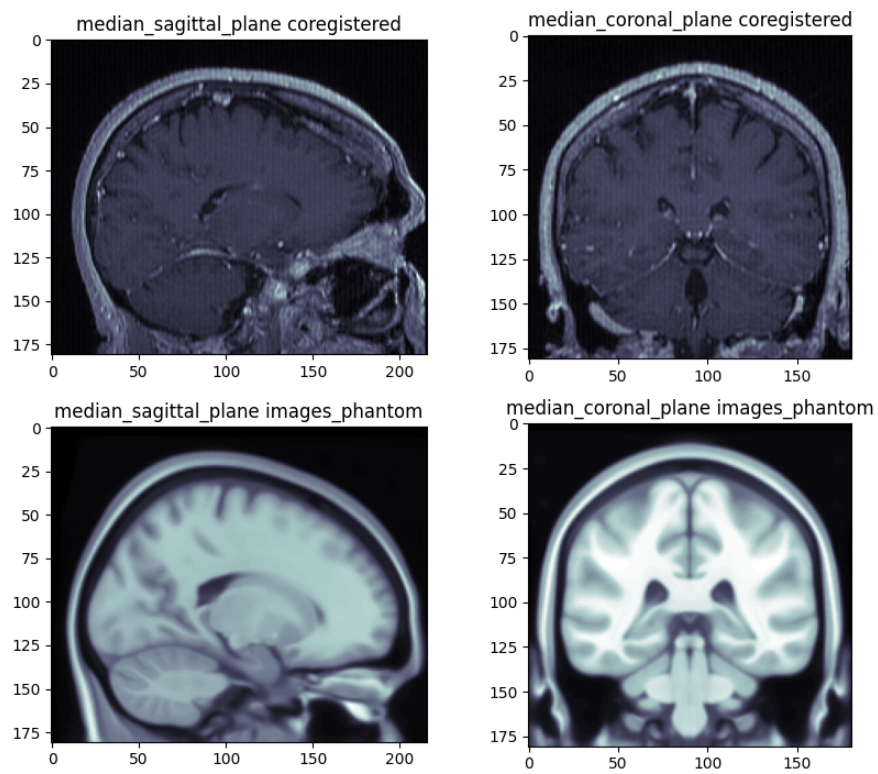


Figure 5: co-registración realizada.

3.2.4 Segmentación

Además se pedía mostrar la parte de tálamo segmentada, como se puede ver en la Figura 6.

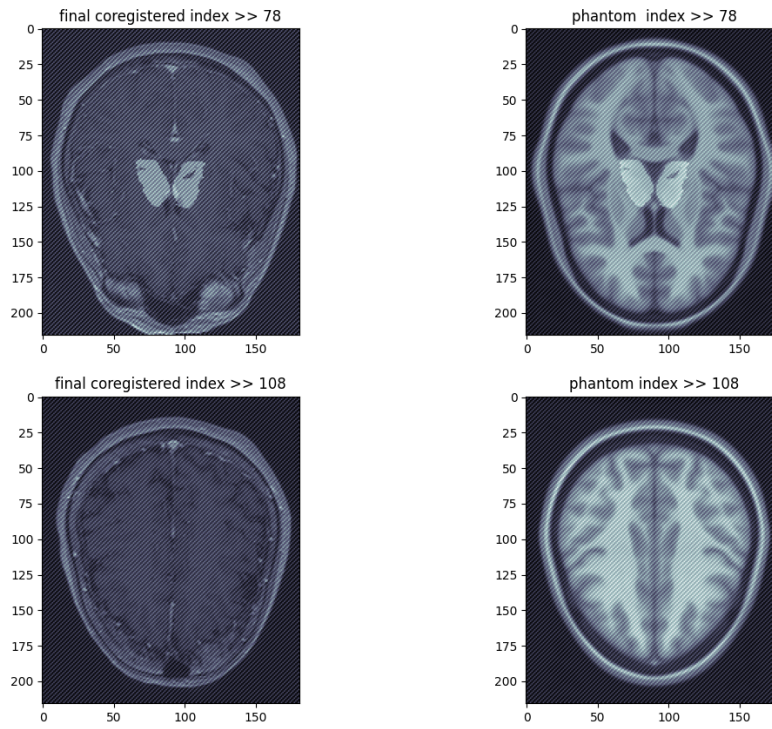


Figure 6: Segmentación de tálamo.

3.2.5 Resultados de la transformacion

Aparentemente, no parece que los landmarks estén muy bien escogidos, o el algoritmo de optimización podría ajustarse mejor. En la Figura 7 vemos como apenas se ha modificado la estructura general de los *landmarks*, y mantienen una forma muy parecida a la original.

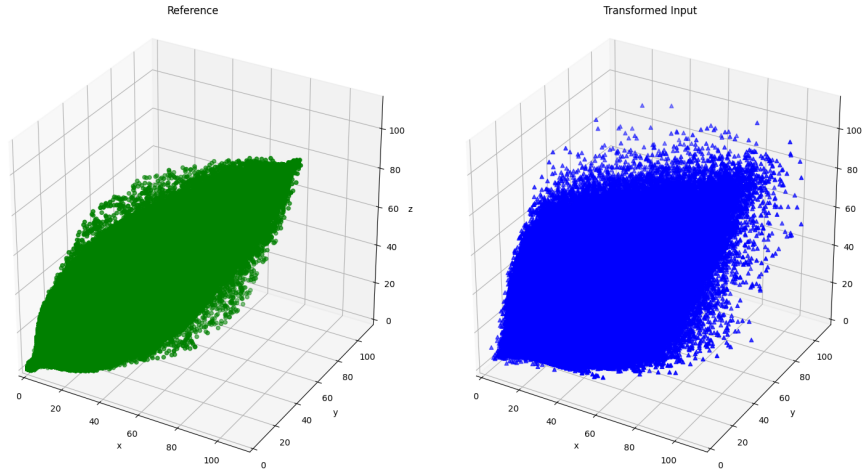


Figure 7: Segmentación de Amígdala.

Esto puede deberse a varios factores, como que estos no son los *landmarks* ideales y se podría haber optado por emplear otros manualmente, el uso de algoritmos de optimización más sofisticados, etc... Es probable que en lugar de haber seleccionado estas *landmarks* de forma "automática", habría sido una mejor práctica seleccionar *landmarks* de forma manual en puntos de interés anatómicos.

4 Conclusiones

Al aplicar los métodos aprendidos en clase, hemos logrado dos tareas fundamentales en el tratamiento de imágenes médicas, como lo son la segmentación de imagen, en este caso para mostrar órganos relevantes en nuestras pruebas, el co-registro de imágenes para alinearlas correctamente y el uso de ficheros DICOM y sus headers para extraer información relevante de los pacientes. Estas técnicas me han permitido obtener una visión más precisa y detallada de las estructuras anatómicas en las imágenes médicas.