

Entregable Fase 2

Segmentación de materia blanca en imágenes de Resonancia Magnética del cerebro. ITK-SNAP

Antonio Delgado Bejarano - 22 de mayo de 2019

Introducción

En este documento vamos a presentar los diferentes algoritmos probados para segmentar la materia blanca, explicando cual de ellos me ha parecido mejor en cuanto a resultados y ejecución. Además se tratará de explicar el fundamento teórico del algoritmo seleccionado.

Uso de ITK-SNAP

En mi caso he usado el programa de libre distribución ITK-SNAP el cual presenta una interfaz sencilla e intuitiva desarrollada sobre C++. Este programa ofrece una buen comportamiento como herramienta semiautomática. Se basa en segmentación por contornos activos, esto es, parte de una spline paramétrica deformable que busca los contornos de los objetos minimizando la energía. En contornos activos hay que tener una idea aproximada del contorno buscado. Se hace evolucionar la curva en la dirección de máximo descenso del gradiente de la energía.

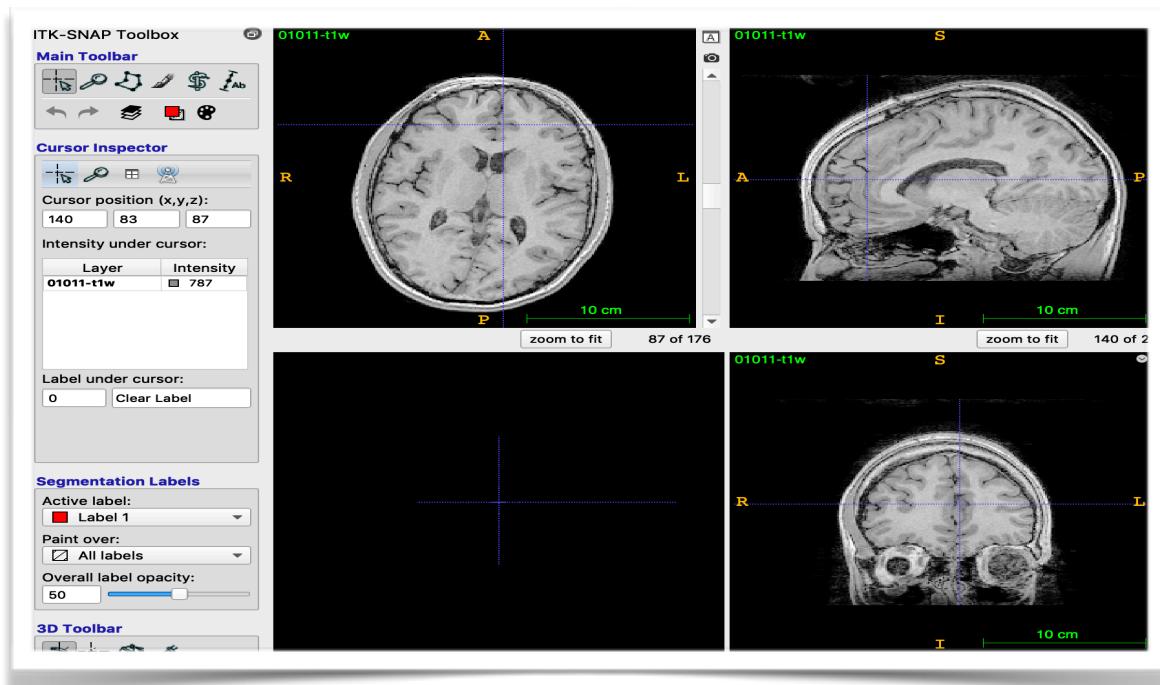


Fig 1.- Interfaz de ITK-SNAP

Para realizar la segmentación lo primero es cargar una imagen en cualquier formato soportado por la aplicación (NNRD, DICOM, NiFTI, etc.) y se aprecian los cortes en tres de las ventanas como se ve en la foto anterior. Para poner en marcha la segmentación escogemos la región de interés y en este punto debemos elegir un método de presegmentación:

El primero con el que probé fue con el método Thresholding, el cual te permite ajustar un umbral máximo y mínimo para resaltar la región del corte que nos interese. También se puede ajustar la suavidad. En mi caso elegí los valores 1254 y 3052, con una suavidad de 5.5. Los resultados obtenidos eran buenos pero a simple vista se podía apreciar alguna que otra diferencia al comparar la segmentación obtenida con la verdad de referencia.

El segundo método probado fue Edge Attraction, el cual consiste en elegir un factor de suavizado que realce una zona en concreto. En mi caso los mejores resultados los obtuve al poner el factor de suavizado muy bajo (en torno a 0,2) pero los resultados no fueron para nada buenos. Al segmentar la materia blanca no detectaba bien el borde con la materia gris y se desbordaba, cogiendo así como materia blanca bastantes más puntos de los indicados. He de decir que creo que seguro que se pueden conseguir mejores resultados con este método de presegmentación pero por desconocimiento del programa o por no saber ajustar los parámetros no lo conseguí.

El último método usado fue el método de Clustering. En este método había que ajustar el número de clusters y los clusters en primer plano. A simple vista se notaba que conseguía bastante mejores resultados que los demás métodos. En mi caso usé 7 como número de clusters iniciales (excepto en algún caso que tuve que usar 8) y 30 iteraciones antes de pasar al siguiente paso. Por tanto, el método elegido fue este.

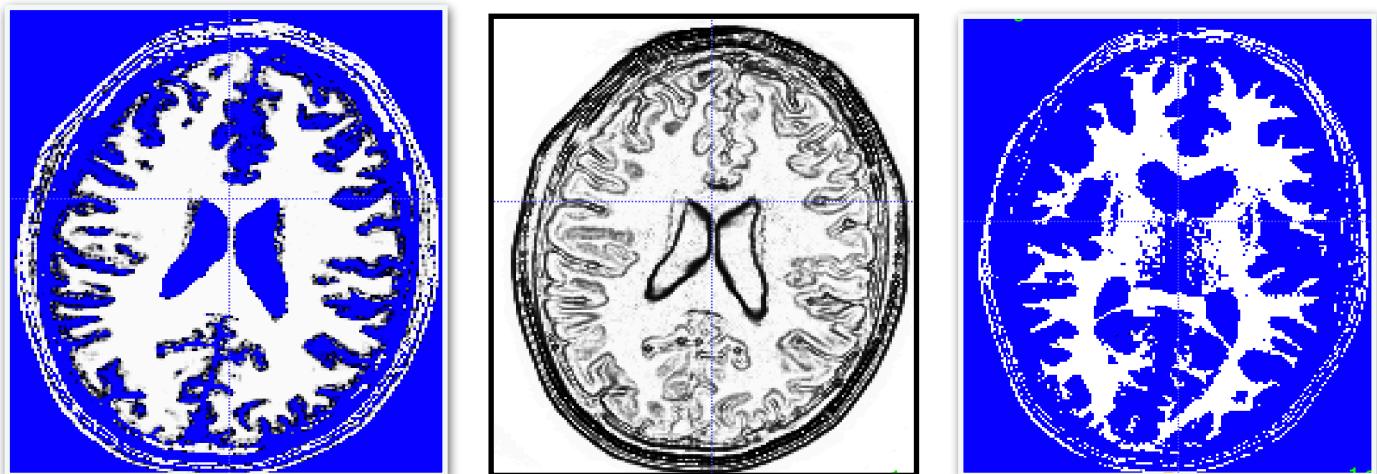


Fig. 2 - Presegmentación con cada uno de los métodos anteriores respectivamente

El siguiente paso consistió en colocar las semillas, en mi caso obtuve buenos resultados con semillas de radio 3,5. Puse especial cuidado en colocar las semillas bien por los bordes y en poner alguna en las zonas centrales para que detectase bien. Por último se deja actuar al algoritmo de crecimiento, parándolo antes de que desborde la zona deseada. Los resultados obtenidos fueron bastante buenos.

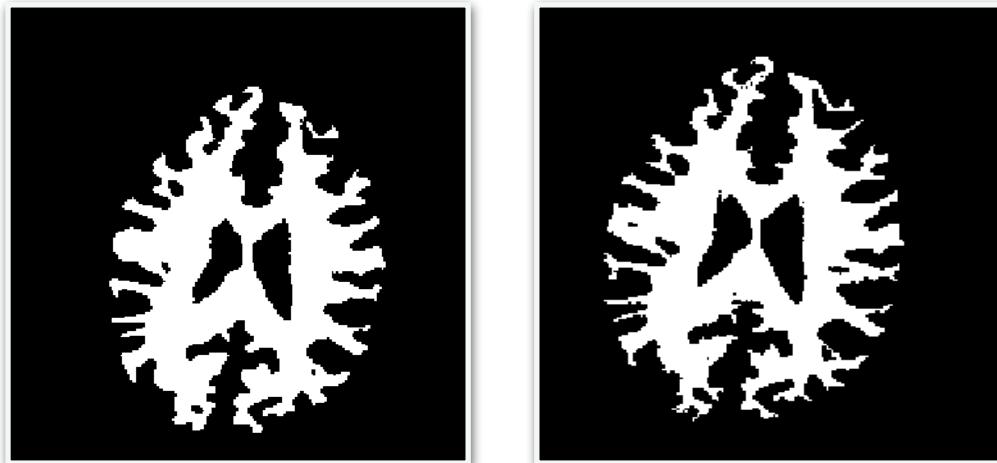


Fig. 3 - Segmentación obtenida (izquierda) y Verdad de Referencia (derecha)

Fundamento teórico

Este algoritmo semiautomático implementa dos algoritmos basados en contornos activos, el de Caselles y el de Zhu.

En el de Caselles las fuerzas se calcula la fuerza con tres parámetros de ponderación y la curvatura de la superficie de contorno. También tenemos una función de velocidad dependiente del gradiente de forma que si el gradiente es grande no debe moverse para quedarse en zonas de grandes variaciones (contornos). La función de velocidad siempre es positiva, por tanto al ser el primer término de la fuerza siempre empuja el contorno hacia afuera, mientras que el último término de la fuerza $\gamma(\vec{v}g_I \cdot n)$ es de advección, es decir, empuja hacia adentro si la función de velocidad crece hacia adentro.

En el de Zhu la fuerza depende de las probabilidades de que el voxel pertenezca al fondo o al objeto, estimándolas en función del nivel de gris del voxel y del nivel de gris del tejido a segmentar. Caselles es más útil cuando hay fronteras entre diferentes tejidos con grandes variaciones, mientras que Zhu es útil cuando el nivel de gris de los tejidos es bien diferenciable del nivel de gris del fondo.

Ambos algoritmos parten de una superficie expresada en forma paramétrica que se hace evolucionar en la normal a la superficie. Las fuerzas controlan cuánto y hacia dónde se mueve cada punto. Las fuerzas internas dependen de la forma de la superficie y las fuerzas externas de los valores de los voxels de la imagen).

$$\frac{\partial}{\partial t} C(t, u, v) = F\mathbf{n}$$

La evolución del contorno no se implementa parametrizando C, sino que emplea una función ϕ tal que los véxeles que pertenecen a C vale 0. Es decir, se hace evolucionar ϕ de forma que su superficie de nivel 0, C, evolucione de la siguiente forma:

$$\frac{\partial}{\partial t} \phi = F \vec{\nabla} \phi$$

Fuentes consultadas

Diapositivas de clase y manuales de IKT-SNAP disponibles en Internet y YouTube.