



Trabajo de Final de Grado
GRADO DE INGENIERÍA INFORMÁTICA
Facultat de Matemàtiques Universitat de Barcelona

ENTORNO PARA SEGMENTACIÓN DE IMÁGENES POR RESONANCIA MAGNÉTICA CEREBRALES

Isabel Lobato González

Director: Laura Igual

Realizado en: Departament de Matemàtica aplicada i Anàlisi. UB

Barcelona, 20 de Junio, 2013

Resumen

Cada día son más las investigaciones en el campo de la neurociencia cognitiva centradas en estudiar estructuras subcorticales del cerebro relacionadas con ciertos desórdenes mentales, como por ejemplo el núcleo caudado que se ha relacionado con el Trastorno por Déficit de Atención e Hiperactividad (TDAH). Estos estudios se basan en la segmentación de estas estructuras en imágenes por resonancia magnética para su posterior análisis y soporte al diagnóstico.

En este trabajo se ha creado un entorno informático para realizar este tipo de segmentaciones de forma automática. El objetivo es proporcionar una herramienta que facilite el trabajo a los médicos y, que al mismo tiempo, se pueda utilizar en el desarrollo de nuevas investigaciones con el fin de continuar mejorando el resultado de segmentación.

Para el desarrollo de este proyecto se han utilizado métodos de visión por computador, que nos permiten la extracción de la información de las imágenes por resonancia magnética necesaria para segmentar.

Abstract

Every day more research in the field of cognitive neuroscience study is focused on subcortical brain structures associated with certain mental disorders, such as the caudate nucleus, which has been associated with Attention Deficit Disorder and Hyperactivity Disorder (ADHD). These studies are based on the segmentation of these structures in magnetic resonance imaging for subsequent analysis and diagnostic support.

This work has automatically created a computing environment for this kind of segmentation. The aim is to provide a tool to facilitate the work of physicians and, at the same time, it can be used in the development of new research in order to continue to improve the segmentation result.

For the development of this project we have used computer vision methods, we allow the extraction of information from magnetic resonance imaging necessary to segment.

Resum

Cada dia són més les investigacions en el camp de la neurociència cognitiva centrades en estudiar estructures subcorticals del cervell relacionades amb certs desordres mentals, com ara el nucli caudat que s'ha relacionat amb el Trastorn per Déficit d'Atenció i Hiperactivitat (TDAH). Aquests estudis es basen en la segmentació d'aquestes estructures en imatges per ressonància magnètica per al seu posterior anàlisi i suport al diagnòstic.

En aquest treball s'ha creat un entorn informàtic per realitzar aquest tipus de segmentacions de forma automàtica. L'objectiu és proporcionar una eina que faciliti la feina als metges i, que al mateix temps, es pugui utilitzar en el desenvolupament de noves investigacions per tal de continuar millorant el resultat de segmentació.

Per al desenvolupament d'aquest projecte s'han utilitzat mètodes de visió per computador, que ens permeten l'extracció de la informació de les imatges per ressonància magnètica necessària per segmentar.

Índice

1. Introducción	4
1.1. Antecedentes	4
1.2. Objetivo del trabajo	4
1.3. Motivación	4
1.4. Estructura del Documento	5
2. Descripción del problema	5
2.1. Imagen por Resonancia Magnética	5
2.2. Estructuras subcorticales del cerebro	6
2.3. Métodos de segmentación	6
2.3.1. Método de segmentación basado en atlas	7
2.3.2. Segmentación basada en Graph Cut	8
2.4. Propuesta de método de segmentación	9
3. Desarrollo del entorno de segmentación	10
3.1. Diseño	10
3.1.1. Requisitos de software	10
3.1.2. Casos de uso	11
3.1.3. Especificación de los casos de uso	12
3.1.4. Fases del proceso	13
3.2. Implementación	15
3.2.1. Programación en MATLAB	15
3.2.2. SPM toolbox	16
3.2.3. OpenCV y Visual Studio	16
3.2.4. Organización de la implementación	16
4. Experimentos	18
4.1. Datos	19
4.2. Resultados cualitativos	19
4.3. Resultados cuantitativos	24
4.4. Tiempo de ejecución	27
5. Planificación	27
5.1. Tareas	27
5.2. Diagrama de Gantt	27
5.3. Evaluación económica	29
6. Conclusiones	30
6.1. Objetivos alcanzados	30
6.2. Mejoras posibles	30
7. Anexos	32
7.1. Anexo A: Guía de usuario	32
7.1.1. Distribución de carpetas	32
7.1.2. Fases del proceso	33
7.1.3. Información general para el uso de la aplicación	35

1. Introducción

1.1. Antecedentes

Este trabajo se enmarca dentro de un proyecto de colaboración de profesores de la Universitat de Barcelona con el Grupo de Neurociencia Cognitiva del Hospital del Mar liderado por Oscar Vilarroya y que cuenta con expertos en Neurociencia Cognitiva.

Muchos de los estudios de investigación realizados por este equipo clínico requieren la segmentación de estructuras subcorticales en imágenes por resonancia magnética (IRM) cerebrales para el estudio de ciertos desórdenes mentales, como el Trastorno por Déficit de Atención con Hiperactividad (TDAH). Uno de los descubrimientos más replicados en las investigaciones sobre TDAH es que los sujetos que sufren este trastorno presentan un volumen de la estructura del núcleo caudado significativamente menor.

Para comprobar esta relación entre TDAH y el volumen del núcleo caudado en nuevas muestras el personal clínico utiliza procedimientos manuales ayudados por programas de visualización de IRM. Estos procedimientos manuales son lentos y subjetivos, provocando que los estudios estadísticos pierdan valor.

Con el objetivo de acelerar y sistematizar los estudios médicos en IRM, en este trabajo se propone un entorno automático de segmentación de estructuras subcorticales del cerebro.

1.2. Objetivo del trabajo

El objetivo del presente proyecto es la creación de un entorno para segmentación automática de imágenes por resonancia magnética cerebrales basado en técnicas de procesamiento de imágenes y visión por computador.

Este proyecto tiene distintas aplicaciones dentro del estudio de disfunciones cerebrales, puede ser utilizado por médicos que quieran segmentar estructuras subcorticales cerebrales, y así evitar tenerlo que hacer manualmente, ya que es un procedimiento difícil y subjetivo. Por otro lado puede ser empleado por investigadores técnicos, con el fin de modificar y mejorar la eficacia del resultado en el futuro.

Lo que se desea conseguir es poder segmentar estas estructuras de forma automática empleando distintos métodos de segmentación, y mejorar algunos ya existentes. El entorno debe ser intuitivo, usable, fiable, seguro y funcional, de modo que no les resulte costoso a los médicos utilizarlo. Por otro lado, el software debe tener un diseño modular, para así tener facilidad a posibles cambios, y existir la posibilidad de reutilización de algunos de estos módulos.

1.3. Motivación

La motivación personal que me ha llevado a hacer este proyecto ha sido el poder diseñar un entorno médico, basándome en una investigación ya existente la cual me resultó muy interesante a causa de mi predilección por la medicina y sobre todo por la informática.

En mi experiencia personal he visto lo que implica ser médico, y desde luego no es una profesión común, las horas que estos profesionales dedican a su trabajo y a sus pacientes no son las de un trabajador corriente, muchas de estas horas se dedican a hacer trabajos costosos que la informática podría solventarles en muy poco tiempo. Como informática me gusta crear nuevas aplicaciones que faciliten la vida y el trabajo a las personas, ¿Por qué no facilitarles este trabajo a los profesionales de la salud y mejorar la calidad de la atención sanitaria?

1.4. Estructura del Documento

El documento queda estructurado de la siguiente forma. En primer lugar, en el capítulo 2, se describirá detalladamente el problema y su entorno, se profundizará en el contexto médico y en los métodos de segmentación empleados. A continuación, en el capítulo 3, se explicará detalladamente como se ha desarrollado todo el entorno, explicando su diseño e implementación. En el capítulo 4 se muestran los resultados obtenidos en los experimentos realizados y se hace un análisis de estos. En el capítulo 5 se muestra como se ha planificado todo el proyecto, y así poder ver la evolución de este. Se concluye explicando en el capítulo 6 que objetivos se han logrado alcanzar y que mejoras posibles existen. El documento consta de 1 anexo, en el que se incluye una guía de usuario completa para poder utilizar el entorno.

2. Descripción del problema

Este capítulo se centra en contextualizar el problema. Se presenta una pequeña aproximación a las imágenes por resonancia magnética y a las estructuras subcorticales del cerebro, y se explican los métodos de segmentación empleados en este proyecto.

2.1. Imagen por Resonancia Magnética

¿Qué es una imagen por resonancia magnética (IRM)? *"Es una técnica no invasiva que utiliza el fenómeno de la resonancia magnética para obtener información sobre la estructura y composición del cuerpo a analizar. Esta información es procesada por ordenadores y transformada en imágenes del interior de lo que se ha analizado. Es utilizada principalmente en medicina para observar alteraciones en los tejidos y detectar cáncer y otras patologías [1]."*

¿Cómo se extraen las imágenes? La resonancia magnética emite unas señales de radiofrecuencia que hacen que los átomos del organismo sean manipulados. Estas señales son detectadas por una gran antena que las envía al ordenador para procesarlas. El ordenador realiza millones de cálculos y crea imágenes en escala de grises de cortes transversales del organismo. Estas imágenes (cortes) se pueden convertir en imágenes tridimensionales (volúmenes), para que puedan ser analizadas por médicos e investigadores.

La Figura 1 ilustra un ejemplo de algunas de las proyecciones axiales de un volumen IRM de un cerebro humano.

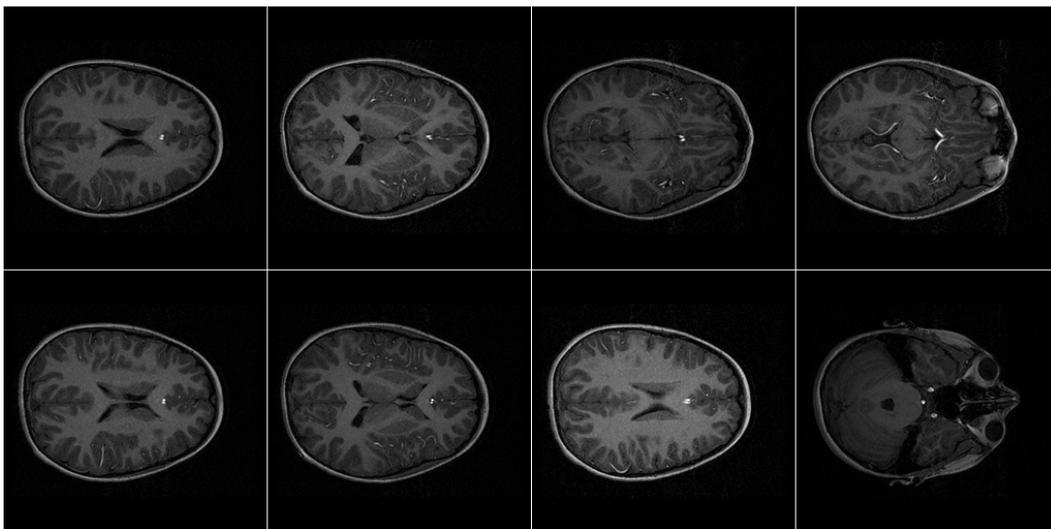


Figura 1: Ejemplo de cortes axiales de un cerebro humano (no consecutivos).

2.2. Estructuras subcorticales del cerebro

Las estructuras subcorticales cerebrales, se encargan de distintas funciones tanto cognitivas como motoras. Existen distintas áreas cerebrales como la *médula espinal* que se encarga de los reflejos, el *cerebro posterior*, constituido por el bulbo, el puente y el cerebelo, que se encarga de las funciones vitales y el equilibrio. El *cerebro medio* o mesencéfalo que produce los reflejos relacionados con el sistema visual y auditivo. El *diencéfalo*, que lo forman el tálamo, el subtálamo, el epitálamo, y el hipotálamo, encargado de la percepción táctil, mantener la temperatura corporal, etc. Y por último los *ganglios basales* constituidos por el caudado, el putamen, el globo pálido, el núcleo subtalámico y la sustancia negra, que intervienen en el control motor y cognitivo [5].

Este proyecto se ha centrado en el núcleo caudado perteneciente a los ganglios basales. Esta estructura se encuentra cerca del centro del cerebro, y hay uno en cada hemisferio. Tiene forma de “C” y se divide en cabeza, cuerpo y cola que se conectan a distintas áreas del cerebro. Esta estructura controla distintas funciones, es importante en el proceso de memorización y aprendizaje, regula la actividad de la corteza cerebral y controla el funcionamiento del mecanismo de retroalimentación. Se ha descubierto que si hay alguna disfunción en el funcionamiento de esta estructura, se pueden dar trastornos como el TOC (trastorno obsesivo compulsivo), TDAH (trastorno por déficit de atención con hiperactividad), depresión, o el síndrome PAP (perdida de motivación), entre otros [6].

En la Figura 2 se muestran marcados en rojo, los núcleos caudados en un cerebro humano

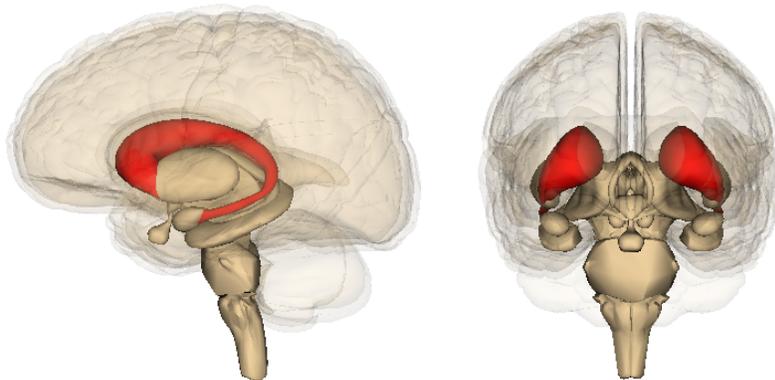


Figura 2: Ilustración de los núcleos caudados (en rojo) [1].

2.3. Métodos de segmentación

Una definición de segmentación en el procesamiento de imágenes, “*La segmentación en el campo de la visión artificial es el proceso de dividir una imagen digital en varias partes (grupos de píxeles) u objetos. El objetivo de la segmentación es simplificar y/o cambiar la representación de una imagen en otra más significativa y más fácil de analizar. La segmentación se usa tanto para localizar objetos como para encontrar los límites de estos dentro de una imagen* [1].”

Para conseguir segmentar, existen distintos métodos de segmentación de imágenes. Este proyecto se centra en dos: método de segmentación basado en atlas y segmentación basada en Graph Cut. Actualmente dentro de la segmentación de estructuras subcorticales en IRM los métodos más comunes y exitosos se basan en atlas. Además, recientemente el método de segmentación basado en Graph Cut destaca por obtener los mejores resultados de segmentación entre los métodos del estado del arte. A continuación se explican estos métodos detalladamente.

2.3.1. Método de segmentación basado en atlas

En el contexto de segmentación de imágenes médicas, en general, un atlas está definido como la asociación de una imagen y la información de la estructura segmentada manualmente. Este atlas contiene una etiqueta para cada una de las estructuras del cerebro, y con esta información podemos prever donde debe estar nuestra estructura objetivo a segmentar.

Los principales pasos de este método están ilustrados en la Figura 3 y son los siguientes [2]:

1. Clasificar la imagen en materia gris (MG), materia blanca (MB) y fluido cerebroespinal (FCS).
2. La imagen MG está elásticamente registrada a partir de su espacio geométrico original para que coincida con una imagen de la plantilla (que representa la distribución esperada de la materia gris en los sujetos de estudio) en el denominado *espacio normalizado*.
3. El campo de deformación obtenido se invierte para mapear del espacio normalizado a el espacio original. Esta deformación se aplica a la segmentación de la estructura en el espacio normalizado, produciendo de este modo una primera segmentación de esta.
4. Con el fin de refinar esta primera segmentación, la máscara MG del sujeto se combina con la máscara obtenida al deshacer la deformación de la estructura en el espacio normalizado. Se combinan de la siguiente manera: el mapa de probabilidades de la estructura y la MG se multiplican y se establece un umbral T_p sobre el resultado, se considera que un vóxel pertenece a la estructura sólo donde el producto es mayor que T_p .

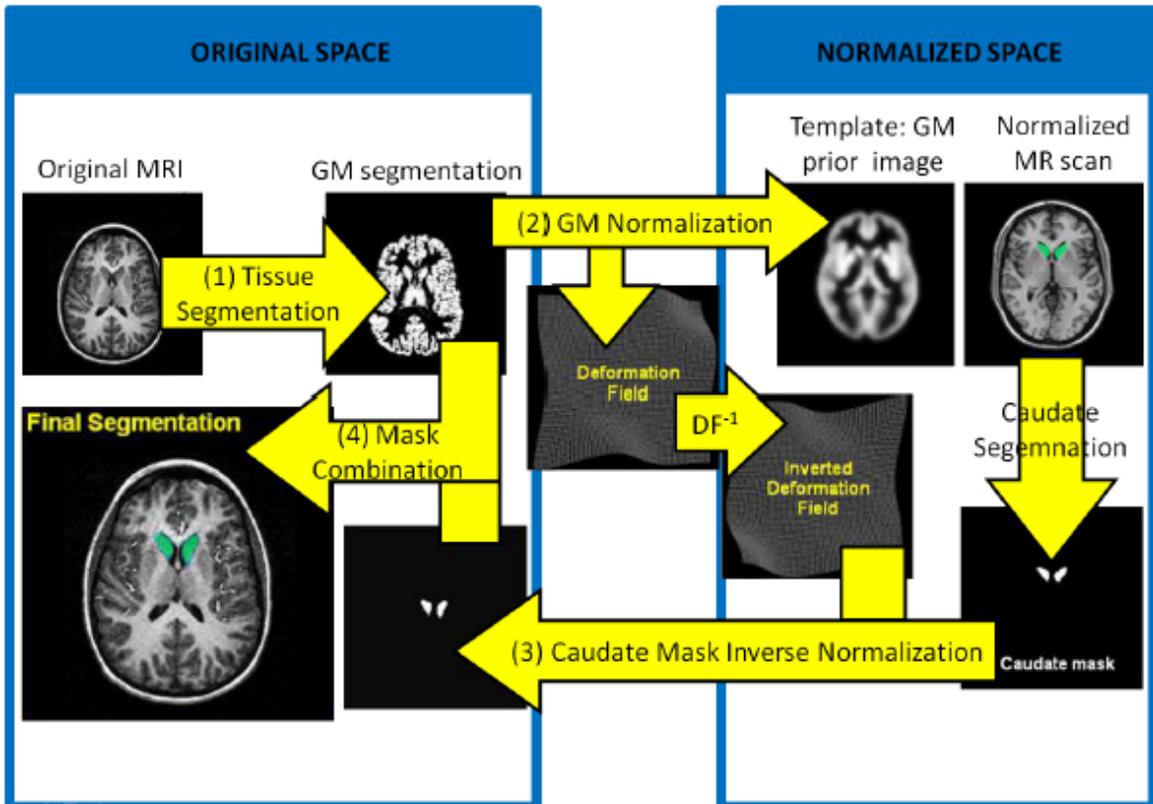


Figura 3: Organigrama del método de segmentación basado en atlas para el caudado [2].

2.3.2. Segmentación basada en Graph Cut

El método de segmentación basado en Graph Cut pertenece al tipo de métodos de segmentación definidos como un problema de minimización de una energía. El método es el siguiente [2]. Se define:

- $X = (x_1, \dots, x_p, \dots, x_{|P|})$ como el conjunto de píxeles de la imagen a segmentar en escala de grises.
- $P = (1, \dots, p, \dots, |P|)$ como el conjunto de índices de la imagen I.
- N como el conjunto de pares no ordenados $\{p, q\}$ bajo un sistema de vecindad de 4-(8-) donde p y q implica que dos píxeles son vecinos.
- $L = (L_1, \dots, L_p, \dots, L_{|P|})$ como un vector binario cuyas componentes especifican asignaciones a píxeles de la imagen. Donde este valor indica si pertenece a “background” o “foreground”, (en nuestro caso “foreground” pertenecería a la estructura, y “background” al resto del cerebro). Es decir, este vector define la segmentación de la imagen.
- La función energía a minimizar como:

$$E(L) = U(L) + \delta B(L),$$

donde $U(L)$ es el término unario, siendo U_p la penalización que se le da a p , por ser “foreground” o “background”

$$U(L) = \sum_{p \in P} U_p(L_p), \quad \begin{cases} U_p(\text{“foreground”}) \\ U_p(\text{“background”}) \end{cases}$$

y donde $B(L)$ es el término de frontera, que tendrá una penalización mayor por discontinuidad entre p y q

$$B(L) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \varphi(L_p, L_q), \quad \varphi(L_p, L_q) \begin{cases} 1, & L_p \neq L_q \\ 0, & \text{en otro caso} \end{cases}$$

Por último el coeficiente δ especifica la importancia de $U(L)$ respecto $B(L)$.

El objetivo de Graph Cut es encontrar la segmentación que minimiza globalmente la energía $E(L)$ de todas las posibles segmentaciones L satisfaciendo algunas restricciones. En nuestro caso estas restricciones están impuestas por medio de unas semillas que son de “background” y de “foreground”, dando por sentado que estas semillas están seguro en sus respectivas regiones. Para poder calcular este mínimo global se representa el problema en un grafo (Figura 4), donde cada nodo representa un píxel, y tomando los nodos T como “background” y S como “foreground”. Las aristas que unen cada píxel a T y S tendrán el valor de los términos unarios $U(L)$ explicados anteriormente, y las aristas entre píxeles vecinos tendrán los términos de frontera $B(L)$. Por último p y v (Figura 4.b y 4.c) corresponden a las semillas. El resultado óptimo se da al encontrar el coste mínimo de corte en el grafo, teniendo la restricción de que p y v deben estar en regiones separadas. Para poder hacer el corte que minimiza la energía utilizamos el algoritmo de max-flow/min-cut [2].

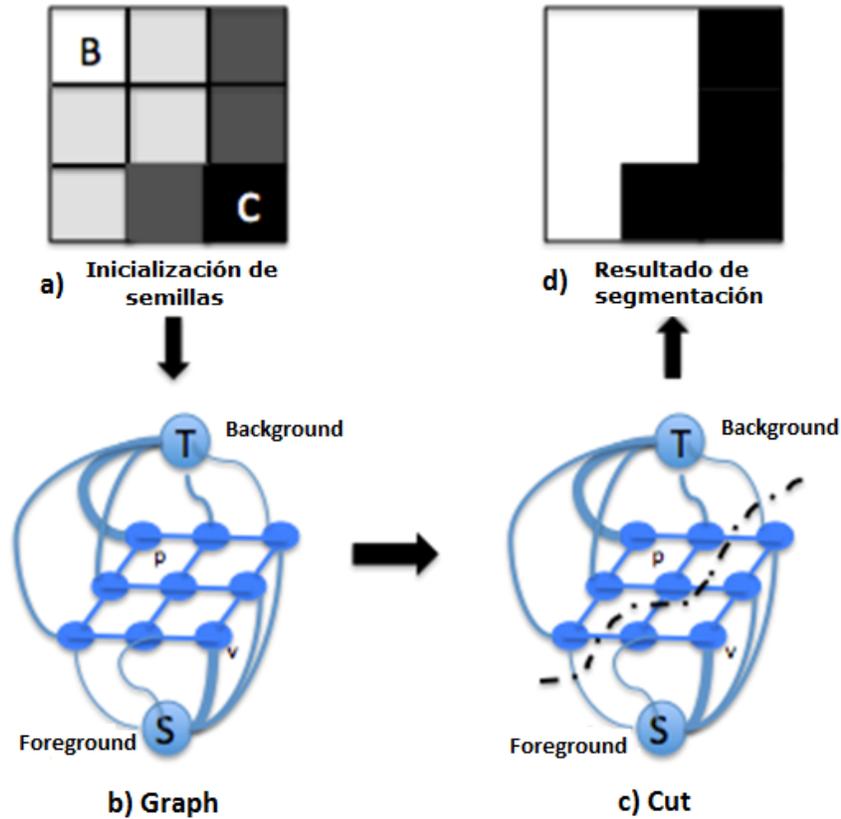


Figura 4: Ejemplo de utilización del método GC. a) Inicialización de las semillas B y C. b) Representación del grafo, donde T y S representan “background” y “foreground” respectivamente, y p y v las semillas inicializadas en a). c) Corte del grafo utilizando el algoritmo max-flow/min-cut. d) Resultado de la segmentación.

2.4. Propuesta de método de segmentación

Nuestra propuesta es segmentar estructuras subcorticales del cerebro en IRM, utilizando tanto el método basado en atlas (BA) como Graph Cut (GC). La idea es segmentar primero con BA y con los resultados que nos ha proporcionado la segmentación, hacer un ajuste mejor a la estructura. Las semillas utilizadas en GC son calculadas a partir del resultado de BA. Se computan haciendo una erosión (disminuyendo el área) del resultado para definir las semillas de “foreground” (estructura objetivo) y una dilatación (aumentando el área) para las de “background” (fondo). Una vez definidas se procede a ejecutar GC teniendo en cuenta que cada semilla corresponde a una región, pudiendo ser “estructura a segmentar” o “resto del cerebro”. El objetivo es minimizar el coste de corte en el grafo separando estas dos regiones y así segmentar la estructura automáticamente. La energía a minimizar por GC se adapta a estructuras pequeñas y poco contrastadas, a este método se le llama CuadateCut y está detallado en el artículo [2].

Este método se ha utilizado para segmentar el núcleo caudado, pero se puede utilizar para otras estructuras que estén etiquetadas en el atlas utilizado. A partir de la segmentación del núcleo caudado se podrá ayudar a detectar si los sujetos sufren TDAH o no.

3. Desarrollo del entorno de segmentación

En este capítulo se explica cómo se ha desarrollado el entorno. Empezando con el diseño de la aplicación donde se han definido los requisitos, casos de uso, y fases del proceso. A continuación se explica que herramientas se han necesitado para la implementación del entorno, y como se han dividido las distintas funciones implementadas.

3.1. Diseño

Se definen todos los requisitos que debe tener el sistema, los casos de uso y se introducen los procesos definidos, con sus respectivas entradas y salidas que serán los módulos o fases a implementar.

3.1.1. Requisitos de software

Los usuarios de este entorno de programación son tanto médicos como investigadores. El médico quiere poder segmentar estructuras del cerebro automáticamente, y así poder evitar el hacerlo manualmente. Esto le ayudará en la detección de trastornos como el TDAH de forma automática. Por otro lado, el investigador requiere este entorno para futuras investigaciones o para mejoras del mismo.

A continuación se definen los requisitos tanto funcionales como no funcionales. Siendo los funcionales los requisitos que el usuario necesita que contenga el sistema y los no funcionales los recursos que debe tener el sistema tales como fiabilidad, usabilidad, estabilidad, seguridad, mantenibilidad, facilidad de prueba y reusabilidad.

Funcionales

- El sistema debe poder leer distintas imágenes por resonancia magnética (IRM).
- El sistema debe poder segmentar diferentes estructuras (indicadas previamente por el usuario).
- El sistema debe poder segmentar la estructura deseada en el volumen, con el método de segmentación basado en atlas.
- El sistema debe poder segmentar la estructura deseada en el volumen, con el método de segmentación basado en Graph Cut.
- El sistema debe mostrar los resultados obtenidos de varios sujetos numéricamente y visualmente.
- El sistema debe poder comparar ambos métodos.

No Funcionales

- El nombre de los sujetos no puede ser público (seguridad).
- El sistema debe ser fácil de ejecutar (usabilidad).
- El sistema debe almacenar y mostrar los resultados en una hoja de cálculo (Excel o OpenOffice) (usabilidad).
- El software debe ser fácil de modificar (modularidad, mantenibilidad, reusabilidad).
- El sistema tiene que estar validado (fiabilidad).

3.1.2. Casos de uso

Una vez especificados todos los requisitos, se realiza un diseño del diagrama de casos de uso, definiendo los actores y sus interacciones con el sistema. El resultado final queda ilustrado en la Figura 5. A continuación se detallan los actores y casos de uso.

Actores

- *Usuario*. Es el actor que interactúa con el sistema.
- *Médico*, tipo de Usuario. Este actor lo único que hace es ejecutar la aplicación, dándole unas imágenes con una estructura a segmentar y esperando la segmentación de esta, donde podrá visualizar los resultados cualitativos obtenidos.
- *Investigador*, tipo de Usuario. Este actor aparte de poder ejecutar las funciones del médico, puede modificar parámetros internos, y evaluar resultados intermedios del programa, dar imágenes segmentadas manualmente para hacer comparaciones y visualizar los resultados cuantitativos, así como medidas de evaluación o porcentajes de ganancia entre métodos.

Casos de uso

- UC1: Cargar IRM a segmentar
- UC2: Elegir estructura a segmentar
- UC3: Visualización resultados cualitativos
- UC4: Cargar imágenes segmentadas manualmente
- UC5: Modificar parámetros
- UC6: Elegir método de segmentación
- UC7: Visualización resultados cuantitativos

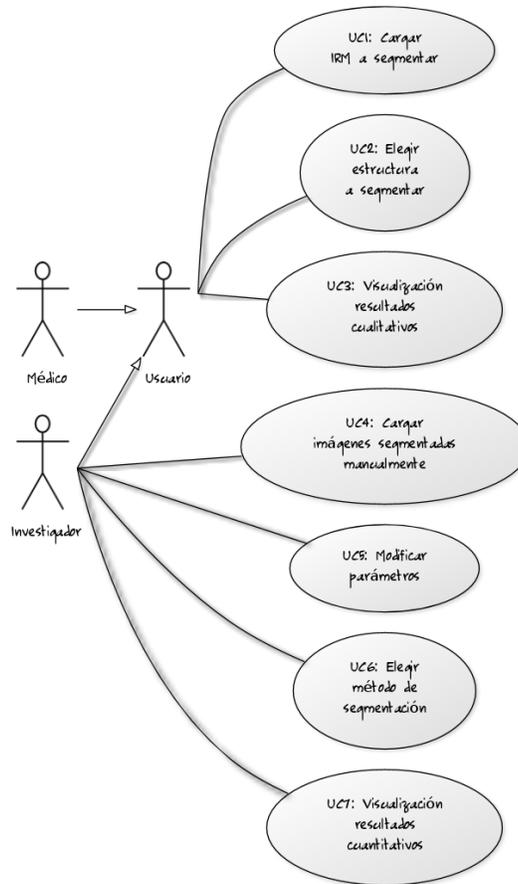


Figura 5: Diagrama de casos de uso.

3.1.3. Especificación de los casos de uso

A continuación se muestra en distintas tablas la especificación de todos los casos de uso que contiene el diagrama diseñado previamente.

Descripción: El usuario carga la IRM que se desea segmentar
Actores: Usuario
Precondiciones: Estas imágenes deben estar en formato .nii o .hdr
Flujo básico: 1. El actor mueve/copia todas las imágenes en la carpeta "Input"
Flujo alternativo: 1a. El sistema comprueba que tengan el formato correcto, y solo segmentará las que cumplan esta precondición.
Postcondiciones: El sistema lee las imágenes a segmentar

Tabla 1: UC1 Cargar IRM a segmentar.

Descripción: El usuario elige las estructuras subcorticales que desea segmentar
Actores: Usuario
Precondiciones: Estas estructura deben existir en el atlas
Flujo básico: 1. El actor escribe en el campo “structures” de la estructura “parameters” los nombres de las estructuras deseadas
Flujo alternativo: 1a. El sistema comprueba que existan en el atlas
Postcondiciones: El sistema lee el nombre de las estructuras a segmentar

Tabla 2: UC2 Elegir estructura a segmentar.

Descripción: El usuario visualiza los resultados cualitativos (imágenes segmentadas)
Actores: Usuario
Precondiciones: La segmentación se ha realizado con éxito
Flujo básico: 1. El actor visualiza los resultados almacenados en la carpeta “Output/Result Slices”
Flujo alternativo: -
Postcondiciones: El actor ha visualizado los resultados

Tabla 3: UC3 Visualización resultados cualitativos.

Descripción: El usuario carga las IRMs segmentadas manualmente por médicos
Actores: Investigador
Precondiciones: Estas imágenes deben estar en formato .nii
Flujo básico: 1. El actor mueve/copia todas las imágenes en la carpeta “GT”
Flujo alternativo: 1a. El sistema comprueba que tengan el formato correcto, y solo comparará las imágenes segmentadas automáticamente con las manuales que cumplan esta precondición.
Postcondiciones: El sistema lee las imágenes segmentadas manualmente

Tabla 4: UC4 Cargar IRMs segmentadas manualmente

Descripción: El usuario modifica los parámetros de la aplicación
Actores: Investigador
Precondiciones: -
Flujo básico: 1. El actor modifica la estructura “parameters” para realizar cambios en la aplicación
Flujo alternativo: -
Postcondiciones: El sistema modifica los parámetros

Tabla 5: UC5 Modificar parámetros.

Descripción: El usuario elige con que método se desea segmentar la imagen
Actores: Investigador
Precondiciones: Este método debe ser AB (basado en atlas) o GC (basado en Graph Cut)
Flujo básico: 1. El actor modifica el campo “method” de la estructura “parameters”
Flujo alternativo: 1a. El sistema comprueba que el método exista
Postcondiciones: El sistema lee el nombre del método con el que se segmentará la imagen

Tabla 6: UC6 Elegir método de segmentación.

3.1.4. Fases del proceso

Como se observa en la Figura 6, el sistema está formado por módulos o fases que se explicarán a continuación, donde cada una de estas está representada como una *caja negra*, que tiene una entrada y una salida, para así poder tener una visión clara de todo el proceso.

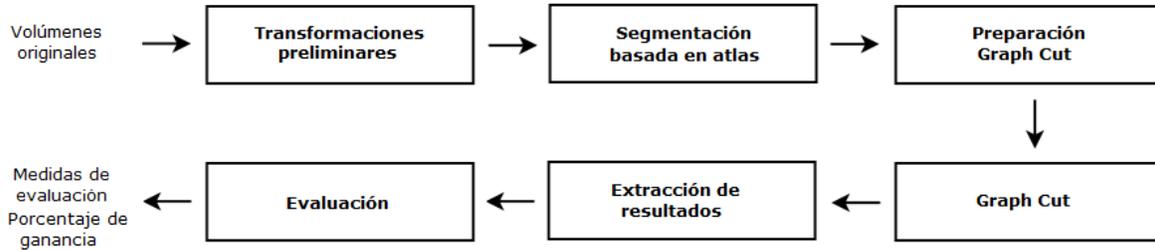


Figura 6: Fase del proceso.

1. *Transformaciones preliminares*

En esta fase se hacen las transformaciones preliminares necesarias de los ficheros de las imágenes. La entrada son los volúmenes de los sujetos a segmentar, y la salida, estos volúmenes transformados. Estas transformaciones pueden consistir en la copia de los volúmenes en una nueva carpeta o el cambio de nombre de estos volúmenes.

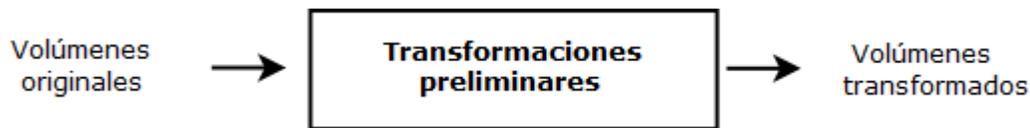


Figura 7: Fase 1, transformaciones preliminares.

2. *Segmentación basada en atlas*

En esta fase a partir de los volúmenes transformados previamente se ejecuta la segmentación basada en atlas, y devuelve los volúmenes de la entrada segmentados.

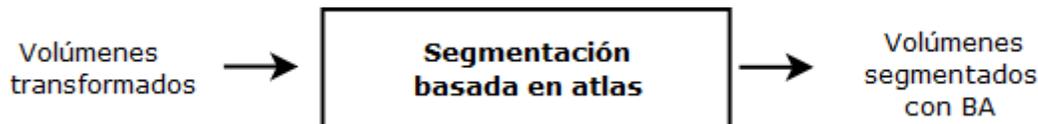


Figura 8: Fase 2, segmentación basada en atlas.

3. *Preparación Graph Cut*

Esta fase prepara las imágenes y las semillas a partir de la segmentación del método BA para aplicar Graph Cut. Tiene dos partes principales:

- Definición del bounding-box que contiene la estructura a segmentar. Extracción de esta bounding-box en una nueva imagen con la que se operará en el GC.
- Inicialización de las semillas para el GC, haciendo erosión para las semillas de la estructura (foreground) y dilatación para las del resto del cerebro (background).

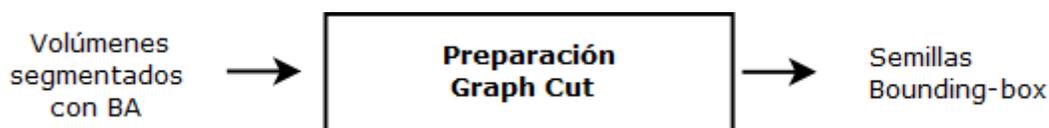


Figura 9: Fase 3, preparación Graph Cut.

4. Graph Cut

Con las semillas y el bounding-box definidos, esta fase hace la segmentación basada en Graph Cut para cada uno de los cortes de la IRM, y devuelve los cortes de los volúmenes finalmente segmentados. Esta fase consta de tres partes:

- Definición de los potenciales unarios.
- Definición de potenciales de frontera.
- Algoritmo max-flow/min-cut.

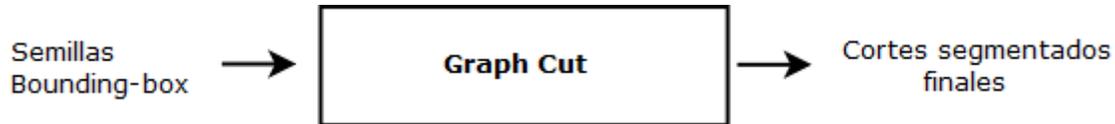


Figura 10: Fase 4, Graph Cut.

5. Extracción de resultados cualitativos

En esta fase se extraen todos los resultados obtenidos. Creando los volúmenes finales y creando los cortes de los volúmenes iniciales con la segmentación dibujada.

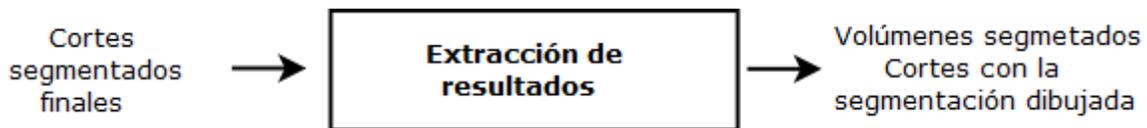


Figura 11: Fase 5, extracción de resultados.

6. Evaluación cuantitativa

Por último esta fase evalúa los resultados obtenidos. Calcula las medidas de evaluación comparando la segmentación automática con la manual (Ground Truth), y mide el porcentaje de ganancia de GC respecto BA.

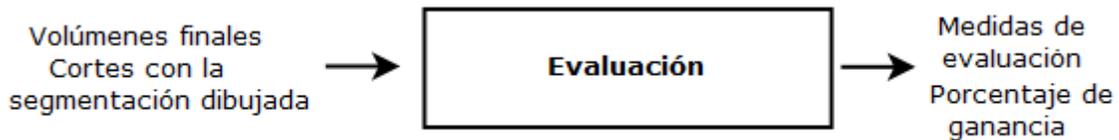


Figura 12: Fase 6, evaluación.

3.2. Implementación

Aquí se explicarán todas las herramientas empleadas para programar el entorno, y se hará una pequeña mención a los scripts y funciones implementadas.

3.2.1. Programación en MATLAB

MATLAB es un lenguaje de alto nivel y un entorno interactivo para el cálculo numérico, la computación, la visualización y la programación, donde los problemas y soluciones se expresan en una notación matricial con la cual estamos familiarizados. Los usos típicos incluyen:

- Matemáticas y computación
- Algoritmo de desarrollo
- Modelado, simulación y creación de prototipos
- Análisis de datos, exploración y visualización
- Ciencia e ingeniería de gráficos
- Desarrollo de aplicaciones

El elemento básico de datos que emplea MATLAB es una matriz que no requiere ser dimensionada. Esto le permite resolver muchos problemas de computación técnica, en mucho menor tiempo en el que se tardaría en escribir un programa en un lenguaje como C o Fortran.

MATLAB ha evolucionado a lo largo de los años gracias a muchos usuarios. Es la herramienta de enseñanza estándar en ambientes universitarios, para cursos de iniciación y perfeccionamiento de las matemáticas, la ingeniería y la ciencia. En la industria, MATLAB es la herramienta de elección para la investigación de alta productividad, desarrollo y análisis.

3.2.2. SPM toolbox

Al trabajar con IRM, en este proyecto se ha requerido del conjunto de herramientas de software SPM para MATLAB.

El toolbox SPM es un conjunto de herramientas de software para el análisis de secuencias de datos de imágenes cerebrales. Este paquete software creado para ser utilizado desde MATLAB sirve para usuarios que estén familiarizados con conceptos estadísticos, matemáticos y de procesamiento de imágenes. Fundamentalmente se utiliza para la manipulación de imágenes IRM en múltiples formatos de neuroimagen.

3.2.3. OpenCV y Visual Studio

El método GC utilizado ha sido implementado en C++ en Microsoft Visual Studio con la librería OpenCV. Ambas definidas a continuación:

“OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado, realizando su programación en código C y C++ optimizados, aprovechando además las capacidades que proveen los procesadores multi núcleo [1].”

“Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET. aunque actualmente se han desarrollado las extensiones necesarias para muchos otros [1].”

3.2.4. Organización de la implementación

En este apartado se hace una pequeña mención a todos los scripts y funciones implementados con MATLAB. Se ha optado por añadir un prefijo en el nombre de las funciones indicando la fase a la que pertenecen. En la Figura 13 se puede observar como estos se relacionan entre ellos, y a continuación la breve explicación de cada uno.

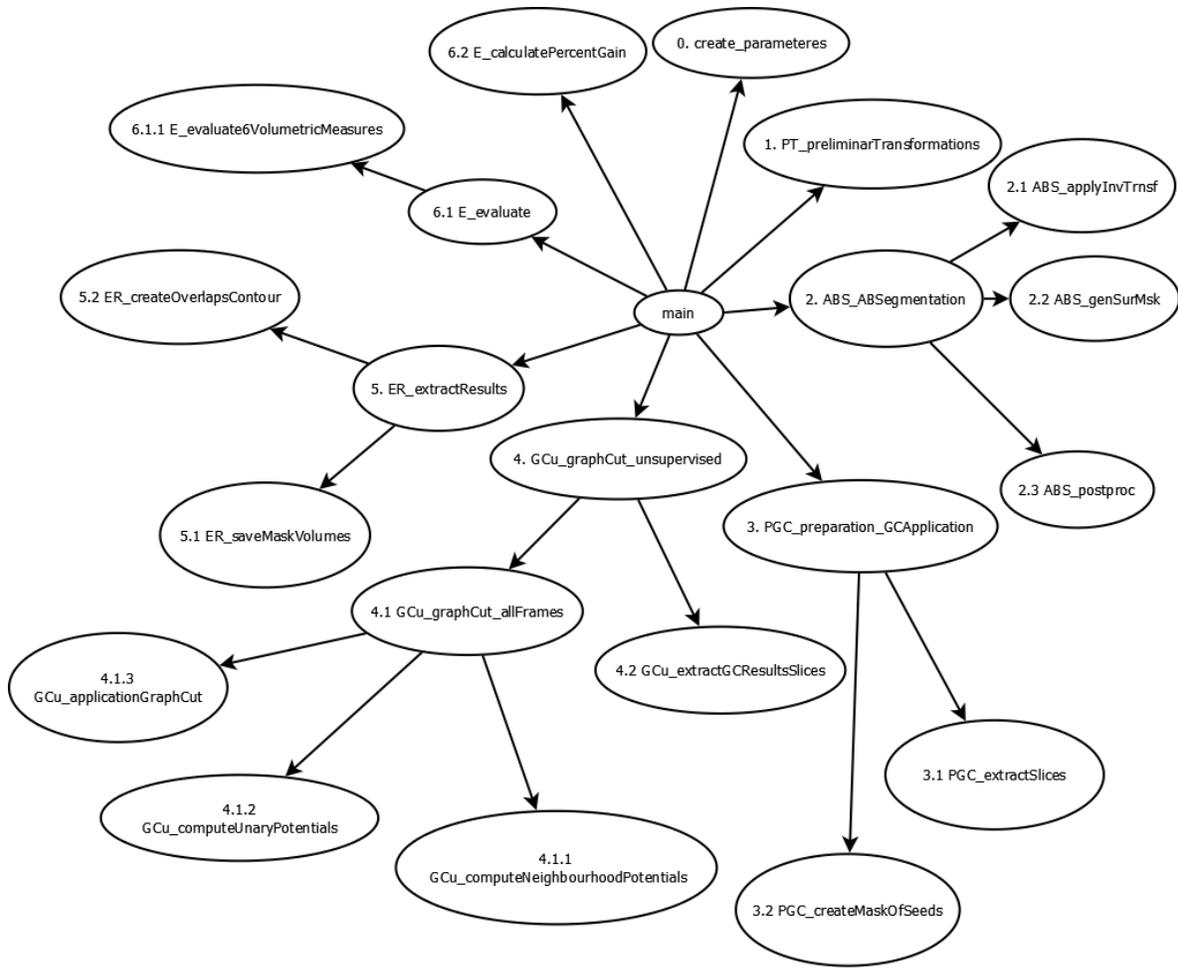


Figura 13: Grafo de llamadas de scripts y funciones de la aplicación.

Todas estas funciones están documentadas en un html obtenido con la herramienta *m2html* [10] en el código adjunto a este proyecto.

Scripts

Main: es el único script existente en el software, y es lo primero que se ejecuta. Desde aquí se hacen las llamadas a cada una de las funciones que corresponden a cada una de las fases explicadas en el diseño.

Funciones

0. *Create_parameters*: está función inicializa la estructura “parameters” que contendrá todos los parámetros necesarios para el buen funcionamiento del programa.

1. *PT_preliminarTransformations*: función que comprueba el formato de las IRMs de entrada y las copia para poder ser procesadas.

2. *ABS_ABSegmentation*: función que ejecuta el método basado en atlas, para ello ejecuta las siguientes funciones:

2.1 *ABS_applyInvTrnsf*: aplica la transformada inversa.

- 2.2 *ABS_genSurMsk*: genera la máscara que rodea la estructura.
- 2.3 *ABS_postproc*: hace un post-proceso de los resultados.
- 3. *PGC_preparation_GCAApplication*: prepara el bounding-box y las semillas para ejecutar el método basado en Graph Cut, llama a las siguientes funciones:
 - 3.1 *PGC_extractSlices*: extrae cada uno de los cortes de los volúmenes.
 - 3.2 *PGC_createMaskOfSeeds*: genera las semillas de “background” y “foreground” que empleará Graph Cut.
- 4. *GCu_graphCut_unsupervised*: función que ejecuta el método basado en Graph Cut, con el bounding-box y las semillas calculadas previamente. Para ello emplea las siguientes funciones:
 - 4.1 *GCu_graphCut_allFrames*: ejecuta Graph Cut para cada uno de los cortes donde se ha detectado que hay estructura.
 - 4.1.1 *GCu_computeNeighbourhoodPotentials*: computa los potenciales de frontera.
 - 4.1.2 *GCu_computeUnaryPotentials*: computa los potenciales unarios.
 - 4.1.3 *GCu_applicationGraphCut* : ejecuta la aplicación Graph Cut.
 - 4.2 *GCu_extractGCResultsSlices*: extrae los cortes de los resultados obtenidos.
- 5. *ER_extractResults*: extrae todos los resultados obtenidos en la segmentación, tanto los volúmenes, como cada uno de los cortes con la imagen segmentada dibujada, llama a:
 - 5.1 *ER_saveMaskVolumes*: guarda los volúmenes de los resultados obtenidos.
 - 5.2 *ER_createOverlapsContour*: guarda los cortes de los resultados obtenidos con la estructura segmentada marcada.
- 6.1 *E_evaluate*: calcula las medidas de evaluación de los resultados obtenidos comparándolos con las imágenes segmentadas manualmente por médicos.
 - 6.1.1 *E_evaluate6VolumetricMeasures*: hace el cálculo de las medidas de evaluación.
- 6.2 *E_calculatePercentGain*: calcula el porcentaje de ganancia del método basado en Graph Cut respecto al método basado en atlas.

4. Experimentos

En este apartado se explican que datos se han utilizado para hacer los experimentos, y se muestran los resultados cualitativos, es decir, las imágenes segmentadas para poder hacer un estudio visual, los resultados cuantitativos, que muestran las medidas de evaluación calculadas para poder hacer un estudio numérico y por último los tiempos de ejecución de cada fase para estudiar la eficiencia en el tiempo.

4.1. Datos

Los datos utilizados para hacer los experimentos han sido extraídos de una base de datos que incluye 58 sujetos distintos, la mitad de ellos con TDAH. El diagnóstico fue establecido por la Unidad de Psiquiatría Infantil en el Hospital Vall d’Hebron, en Barcelona, España, y coordinado por la Unidad de Investigación en Neurociencia Cognitiva (URNC) de la Fundación IMIM. La edad media de los sujetos es de 11,2 años. Estos sujetos fueron elegidos por su capacidad de usar ambas manos (ambidiestros) y su coeficiente intelectual. Las imágenes por resonancia magnética del cerebro fueron extraídas por el sistema 1,5-T. La resolución de las exploraciones es de 256 x 256 x 60 píxeles con cortes de un grosor de 2 mm.

4.2. Resultados cualitativos

Aquí se muestran algunos ejemplos de los resultados obtenidos segmentando el caudado izquierdo/derecho y el putamen izquierdo/derecho.

En la Figura 14 se pueden ver los resultados de proyecciones axiales con la segmentación del caudado izquierdo y derecho. En la primera columna se muestra la imagen original, en la segunda la comparación de la segmentación automática con la manual del método basado en atlas, y en la tercera la misma comparación pero con el método basado en Graph Cut.

Como se puede apreciar en la primera y en la segunda fila GC hace una mejor segmentación, ya que en la primera elimina la parte de caudado que BA detecta y no existe en la segmentación manual, y en la segunda abarca más caudado del que consigue abarcar BA. En la tercera fila la segmentación es casi idéntica, la diferencia es ínfima, ya que BA ya hace una segmentación casi óptima. Por otro lado, en las dos últimas filas se puede observar que GC no mejora el resultado de BA, sino que lo empeora, ya que toma por caudado zonas donde este no existe, y donde ni siquiera BA lo había detectado.

En la Figura 15 se muestran los resultados de la segmentación del putamen en proyecciones axiales distintas, donde cada fila corresponde a un sujeto. En la primera columna se muestra la imagen original, en la segunda la segmentación automática realizada con el método BA, y en la tercera la realizada con GC. En esta ocasión no hay GT y es difícil analizar los resultados.

En estas imágenes se puede apreciar que en comparación con BA la segmentación con GC es mas suave, los contornos parecen mas redondeados y parece que se ajusta mejor a la estructura del putamen.

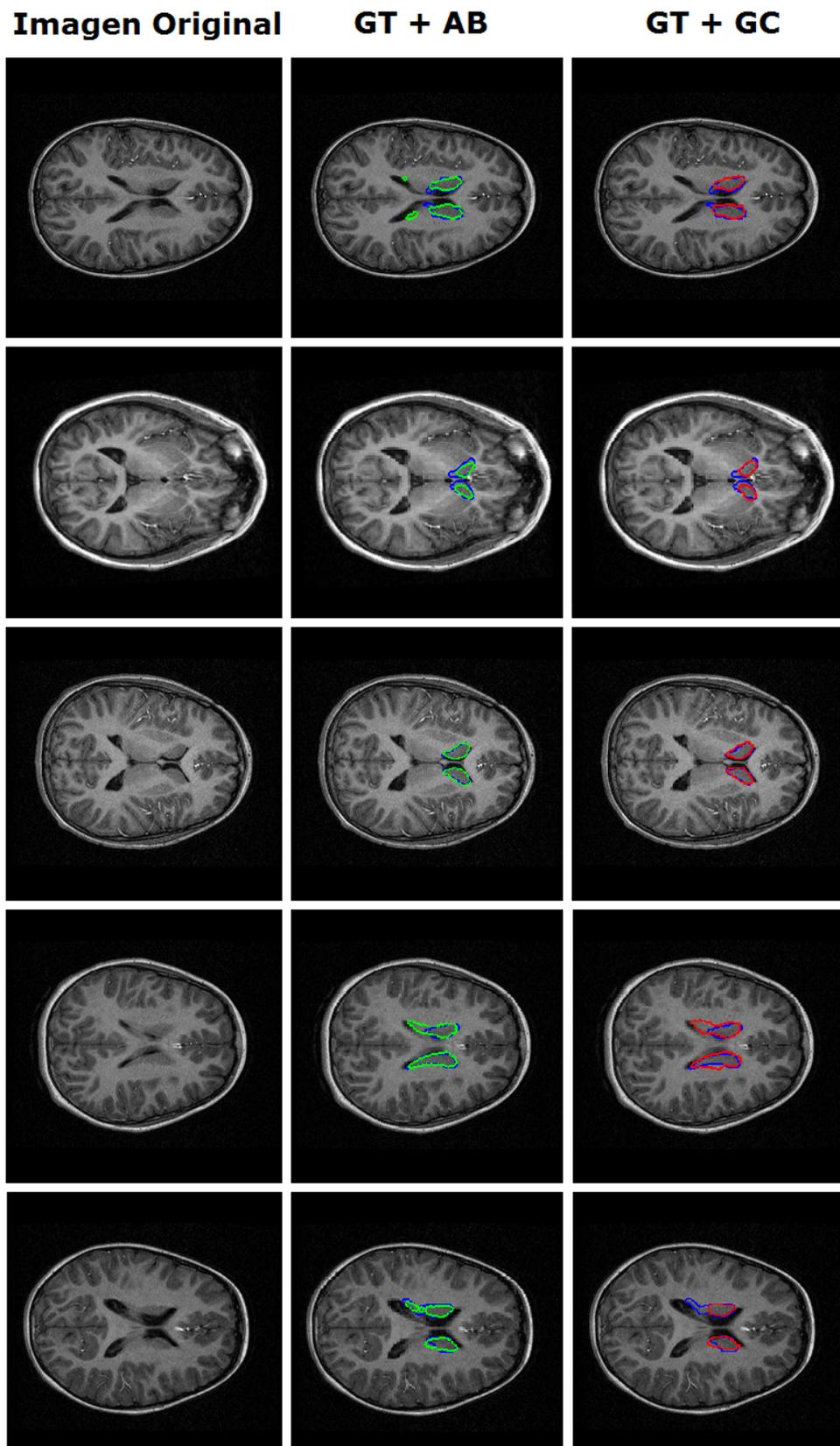


Figura 14: Ejemplos de la segmentación del Caudado.

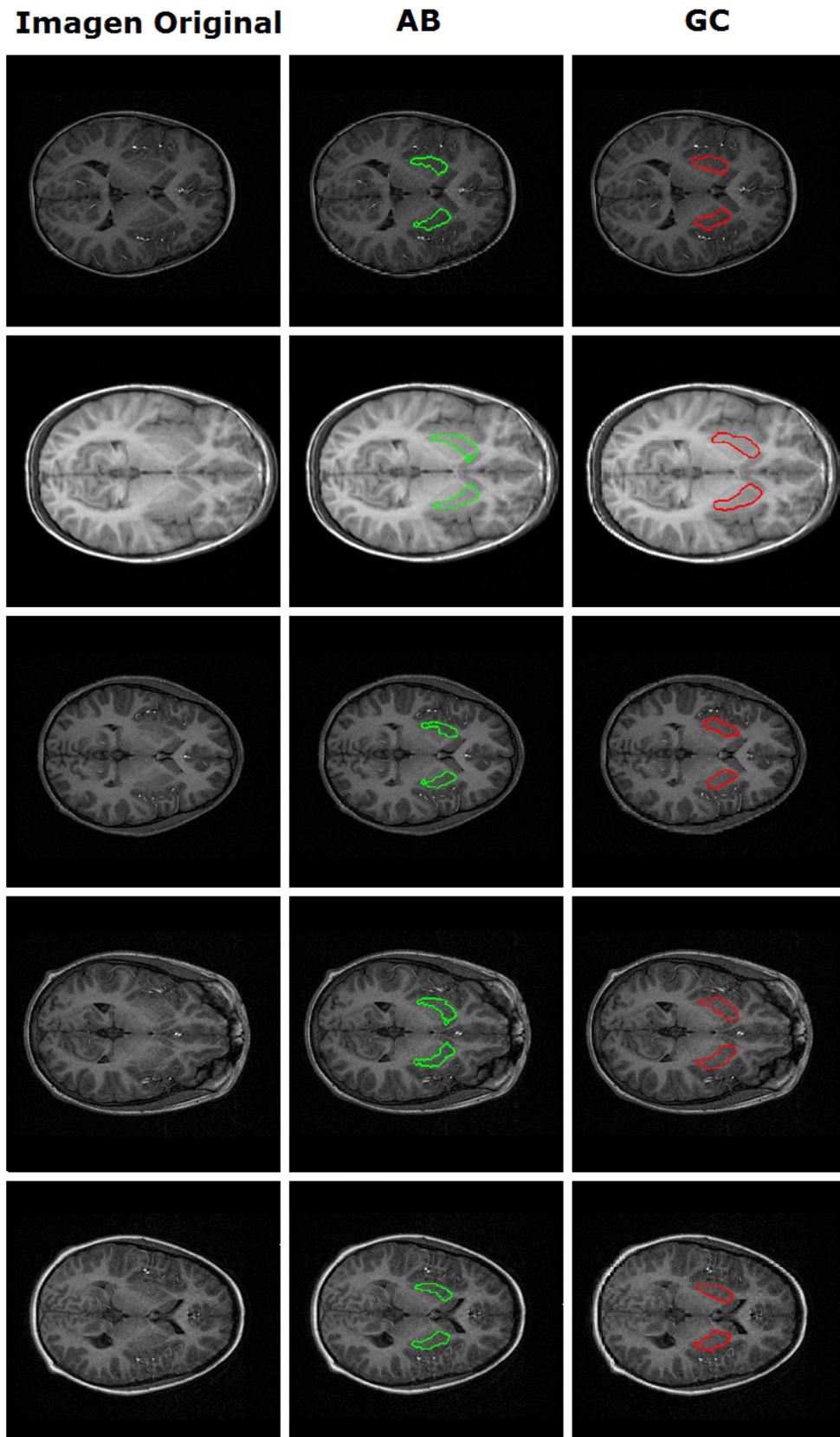


Figura 15: Ejemplos de la segmentación del Putamen.

En la Figura 16 se muestran ejemplos del caudado segmentado automáticamente comparado con la segmentación manual. En la comparación se puede apreciar que en las 3 primeras filas GC consigue ajustar mejor el caudado a la segmentación manual previa realizada . En cambio, en las 2 últimas no parece mejorar el resultado de BA.

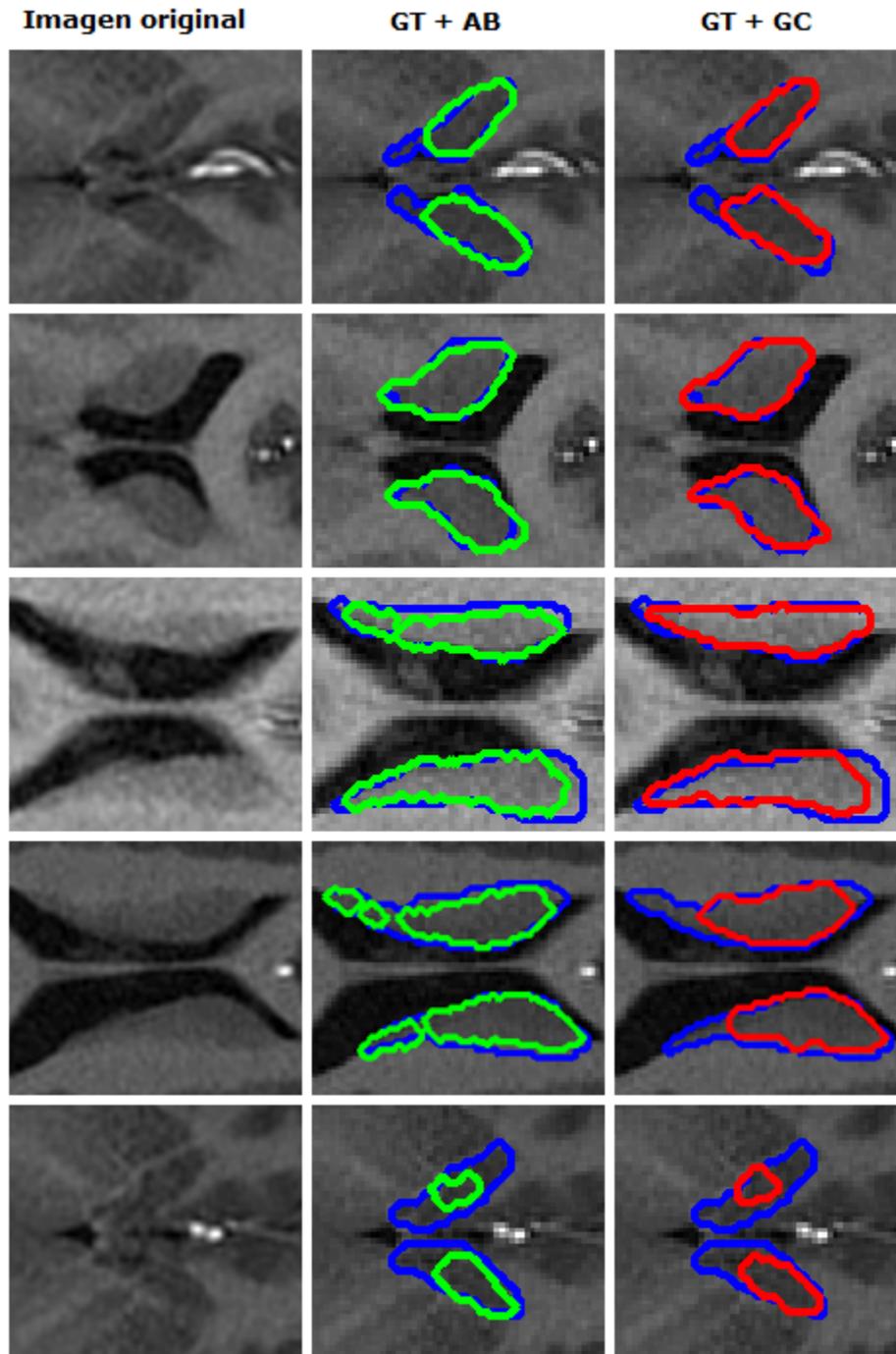


Figura 16: Ejemplos del caudado segmentado.

En la Figura 17 se observan ejemplos del putamen segmentado automáticamente, en estas imágenes es difícil de valorar cuan correcta es la segmentación a causa de la ausencia de GT. Más o menos comparando con la imagen original se puede apreciar como en las dos primeras filas GC consigue segmentar mejor al putamen, y en las 2 últimas como empeora el resultado.

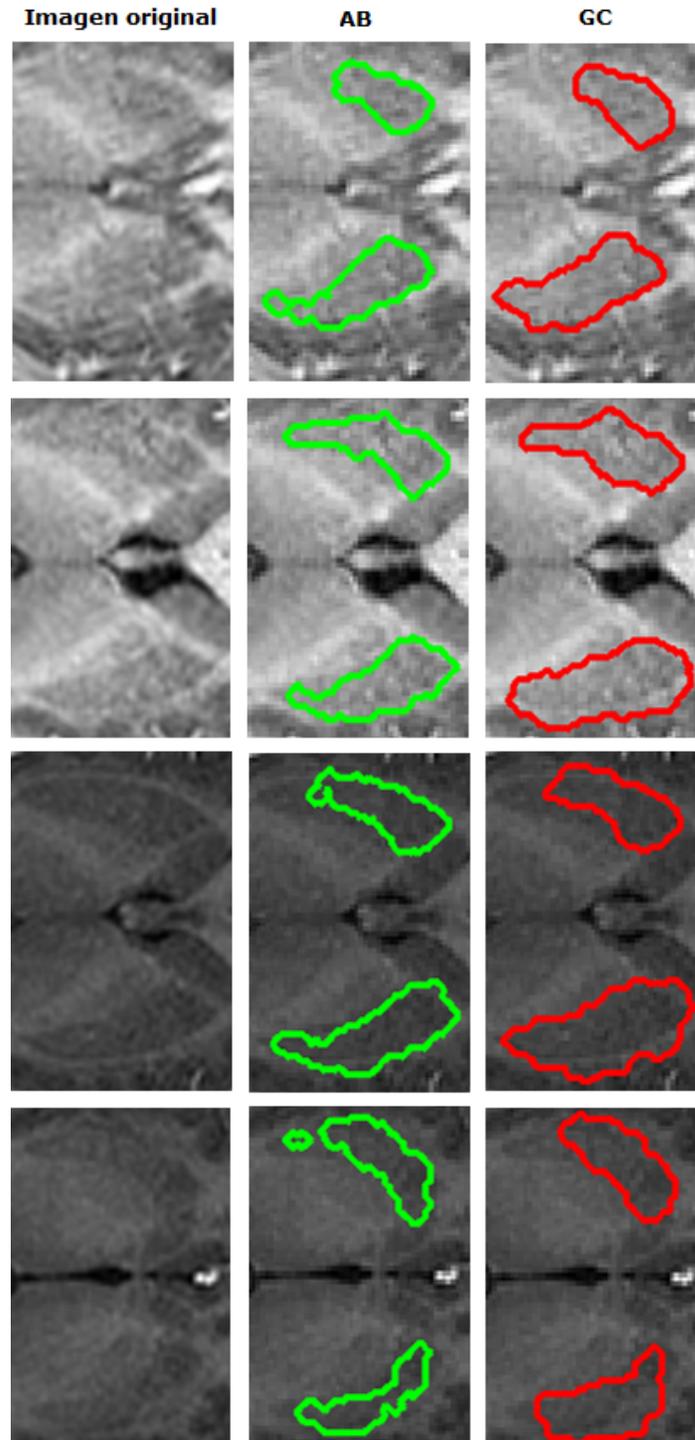


Figura 17: Ejemplos de la segmentación del putamen.

4.3. Resultados cuantitativos

Para obtener estos datos se han realizado las pruebas segmentando ambos caudados (izquierdo y derecho) de 58 sujetos distintos. Se comparan las segmentaciones automáticas con las segmentaciones manuales para valorar cuanto se acercan a la realidad. Para estos cálculos se han utilizado las siguientes medidas de evaluación [11]:

- **VO: Volumetric overlap (en %):** Número de voxels en la intersección de la segmentación automática y de la segmentación manual, dividido por el número de vóxeles en la unión de ambas segmentaciones, el resultado se multiplica por 100. Este valor es 100 para una segmentación perfecta y tiene 0 como el valor más bajo posible, cuando no hay superposición en absoluto entre ambas segmentaciones.
- **SI: Volumetric similarity index (en %):** Intersección de las segmentaciones partido el volumen medio de ambas. 100 para una segmentación perfecta y tiene 0 como el valor más bajo posible.
- **VD: Relative absolute volume difference (en %):** El volumen total de la segmentación automática se divide por el volumen total de la segmentación manual. A partir de este número se resta 1, el valor absoluto será tomado, y el resultado se multiplica por 100. Este valor es 0 para una segmentación perfecta y mayor que cero en caso contrario. El valor perfecto de 0 también se puede obtener para una segmentación no perfecta, siempre y cuando el volumen de que la segmentación es igual al volumen de la referencia.
- **AD: Average symmetric absolute surface distance (en mm):** Se determinan los voxels fronterizos en ambas segmentaciones (auto y manual). Para cada vóxel en estos conjuntos, se determina el voxel más cercano en el otro conjunto (utilizando la distancia euclídea). Todas estas distancias se almacenan, para los voxels fronterizos de ambas segmentaciones. El promedio de todas estas distancias da los promedios distancia de la superficie absoluta simétrica. Este valor es 0 para una segmentación perfecta.
- **RMSD: Symmetric RMS (Root Mean Square) surface distance (en mm):** Esta medida es similar a la medida anterior, pero guarda las distancias al cuadrado entre los dos grupos de voxels fronterizos. Se extrae la raíz del promedio de los valores al cuadrado y obtenemos el valor final. Este valor es 0 para una segmentación perfecta.
- **MC: Maximum symmetric absolute surface distance (en mm):** Se toma el máximo valor de las distancias calculadas en las medidas anteriores. Este valor es 0 para una segmentación perfecta.

En la Tabla 7 se muestra la media y la desviación estándar de los resultados obtenidos de 58 sujetos en ambos caudados empleando la segmentación basada en atlas.

En la Tabla 8, se muestran los resultados de los 58 sujetos estudiados empleando el método basado en Graph Cut, a partir de las semillas del método basado en atlas.

Con estos valores se puede ver que los resultados se ajustan bastante a la realidad con ambos métodos, ya que *VO* y *SI* tienen un tanto por ciento bastante alto (superior al 70%), *VD* oscila entre el 10% y el 15%, siendo 0% una segmentación perfecta, *MC* nos dice que en el peor de los casos, ha habido una distancia de entre 6 y 8 mm, pero *AD* y *RMSD* que son la medias de todas las distancias dan un valor muy bajo que se aproxima mucho a 0, así que se puede decir que el resultado es bastante exitoso.

Caudado izquierdo		
<i>Medida de evaluación</i>	<i>Media</i>	<i>Desviación estándar</i>
VO	71,29	4,43
SI	83,16	3,05
VD	15,10	7,02
AD	0,002	0,001
RMSD	0,07	0,01
MC	6,75	4,22
Caudado derecho		
<i>Medida de evaluación</i>	<i>Media</i>	<i>Desviación estándar</i>
VO	72,87	4,27
SI	84,23	2,95
VD	12,55	6,54
AD	0,002	0,001
RMSD	0,07	0,02
MC	7,74	8,98

Tabla 7: Media y desviación estándar de los resultados obtenidos con BA.

Caudado izquierdo		
<i>Medida de evaluación</i>	<i>Media</i>	<i>Desviación estándar</i>
VO	70,58	4,45
SI	82,67	3,14
VD	10,39	6,20
AD	0,003	0,001
RMSD	0,07	0,01
MC	6,81	4,24
Caudado derecho		
<i>Medida de evaluación</i>	<i>Media</i>	<i>Desviación estándar</i>
VO	71,29	3,66
SI	83,19	2,57
VD	10,12	7,50
AD	0,002	0,0005
RMSD	0,07	0,02
MC	7,50	9,05

Tabla 8: Media y desviación estándar de los resultados obtenidos con GC.

Caudado izquierdo		
<i>Medida de evaluación</i>	<i>Ganancia Media</i>	<i>Desviación estándar</i>
VO	-0,71	3,41
SI	-0,49	2,38
Caudado derecho		
<i>Medida de evaluación</i>	<i>Ganancia Media</i>	<i>Desviación estándar</i>
VO	-1,57	2,54
SI	-1,04	1,73

Tabla 9: Valores de ganancia de GC respecto BA.

En la Tabla 9 se muestran los valores de ganancia de BA respecto GC, es decir, si el valor es positivo, implica que el método GC se acerca más a la realidad que el método BA, en caso contrario, el método BA es el que más se aproxima. En la media se puede apreciar que GC no mejora BA en la mayoría de casos, pero se aproxima bastante. El porcentaje de éxito se calcula a partir del número de

sujetos estudiados, en los 58 sujetos que se han estudiado 12/58 segmentaciones son mejores después de aplicar GC, es decir el 21% de las veces. Para el caudado izquierdo vemos que hay 18/58, es decir, un 31%. Estos porcentajes no son muy altos a causa de que se está ejecutando un Graph Cut no supervisado. Implementando el Graph Cut supervisado de [2] estos valores serían mucho mejores. En las Figuras 18 y 19 se muestra el gráfico donde se pueden apreciar las diferencias entre ambos métodos comparando el *Volumetric Overlap (VO)* y el *Volumetric Similarity Index (SI)*.

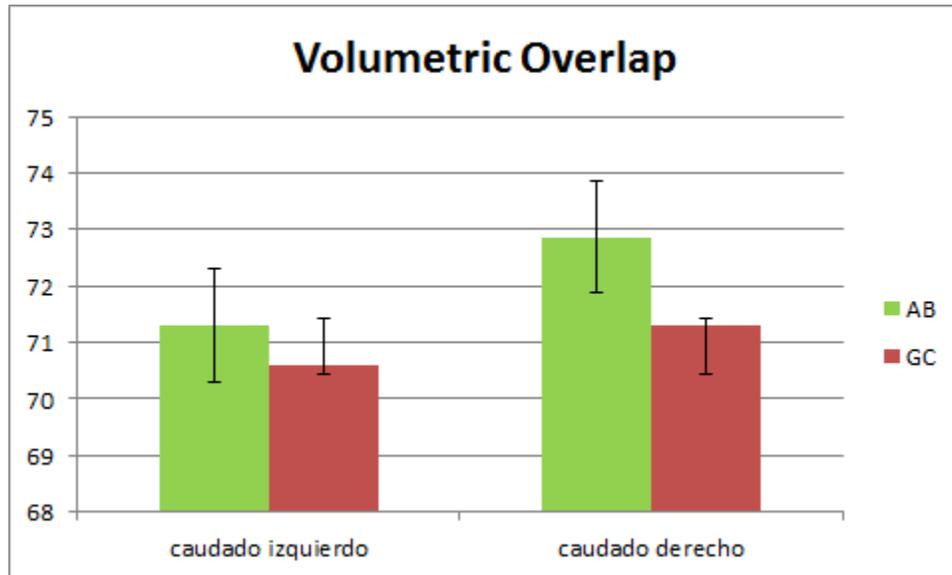


Figura 18: Gráfico comparativo del Volumetric Overlap (VO).

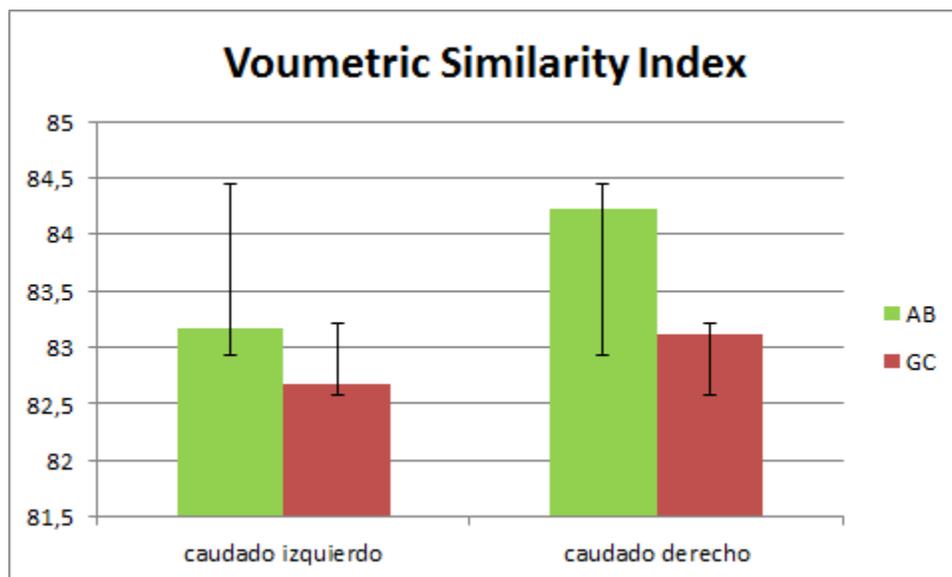


Figura 19: Gráfico comparativo del Volumetric Similarity Index (SI).

4.4. Tiempo de ejecución

Respecto al tiempo de ejecución, la Tabla 10 muestra los tiempos transcurridos para cada fase, ejecutando 1 IRM y segmentando 1 sola estructura. El tiempo total asciende a 1520 segundos, que equivale a 25 minutos aproximadamente. La fase que abarca la gran parte del tiempo de CPU es la Fase 2 (segmentación basada en atlas), con unos 23 minutos. Estos tiempos han sido tomados en un ordenador con 4 GB de RAM y un procesador de doble núcleo.

<i>Fase</i>	<i>Tiempo (segundos)</i>
1. Transformaciones preliminares	1,997
2. Segmentación basada en atlas	1.390,718
3. Preparación Graph Cut	13,447
4. Graph Cut	38,314
5. Extracción de resultados	59,780
6. Evaluación	14,929
Total	1.519,185

Tabla 10: Tiempo de ejecución de cada fase.

5. Planificación

En este capítulo hay una explicación detallada de cómo se ha planificado todo el proyecto. Se listan todas las tareas a realizar, y cuanto tiempo se emplea para cada una de ellas. Seguidamente se muestran estas tareas en un diagrama de Gantt, y por último se evalúa económicamente el coste de la propuesta.

5.1. Tareas

En este apartado se listan todas las tareas realizadas divididas en distintas fases: preparación, diseño, implementación, pruebas y resultados y documentación. Y las semanas que han sido necesarias para llevarlas a cabo. Este proyecto ha durado desde mediados de Diciembre de 2012 a mediados de Junio de 2013, contando que cada mes tiene 4 semanas, se ha necesitado para este proyecto un total de 24 semanas más o menos. En la Tabla 11 se listan todas las tareas mostrando la semana que se ha iniciado la tarea y la duración de estas.

5.2. Diagrama de Gantt

En la Figura 20 se muestra el diagrama de Gantt con todas las tareas listadas previamente. En el gráfico se puede apreciar cuales son más costosas y cuales menos. Hay tareas que se han realizado en paralelo y otras que se han planificado de forma secuencial ya que hasta no terminar una no se podría continuar con la siguiente.

La primera fase ha sido la de formación la cual ha sido para instalar el nuevo software y leerse y entender todo el proyecto previo del método Caudate Cut [2], tarea que ha necesitado 4 semanas para realizarse, ya que había que entender bien como se había llevado a cabo la investigación para poder hacer este proyecto. Después se ha realizado el diseño, en un mes se han diseñado los casos de uso y las fases del proceso a desarrollar. La fase que más tiempo ha requerido ha sido la de implementación, ya que ha sido un entorno bastante largo de programar y aunque se haya reutilizado código ha sido bastante costoso. En penúltimo lugar está la fase de pruebas y resultados la cual ha precisado tiempo a causa de que se han hecho pruebas con muchos sujetos, y esto ha sido un proceso largo. Por último el diagrama muestra como se ha planificado la parte de la documentación.

<i>Tarea</i>	<i>Inicio</i>	<i>Duración</i>
FASE I - PREPARACIÓN	0	4
Instalación y aprendizaje de SPM8	0	1
Estudio del proyecto previo CaudateCut	0	4
Instalación OpenCV	0	1
Instalación Visual Studio 2010	0	1
Aprendizaje herramienta M2HTML	0	1
FASE II - DISEÑO	4	4
Definición de requisitos y casos de uso	4	2
Diseño de fases del nuevo programa	6	2
FASE III - IMPLEMENTACIÓN	8	8
Implementación fase 1: transformaciones preliminares	8	1
Implementación fase 2: método basado en atlas	9	1
Implementación fase 3: preparación para GC	10	1
Implementación fase 4: método basado el Graph Cut	11	2
Implementación fase 5: extracción de resultados	13	1
Implementación fase 6: evaluación de resultados	14	2
FASE IV - PRUEBAS Y RESULTADOS	16	5
Pruebas de ejecución fase 1: transformaciones preliminares	16	1
Pruebas de ejecución fase 2: método basado en atlas	16	1
Pruebas de ejecución fase 3: preparación para GC	16	1
Pruebas de ejecución fase 4: método basado el Graph Cut	16	1
Pruebas de ejecución fase 5: extracción de resultados	16	1
Pruebas de ejecución fase 6: evaluación de resultados	17	1
Pruebas de ejecución de todo el proceso	18	2
Estudio de resultados obtenidos	20	1
FASE V - DOCUMENTACIÓN	21	3
Índice	21	1
Introducción	21	1
Descripción del problema	22	1
Desarrollo detallado	22	2
Planificación	22	1
Conclusiones	23	1

Tabla 11: Tareas planificadas en el proyecto.

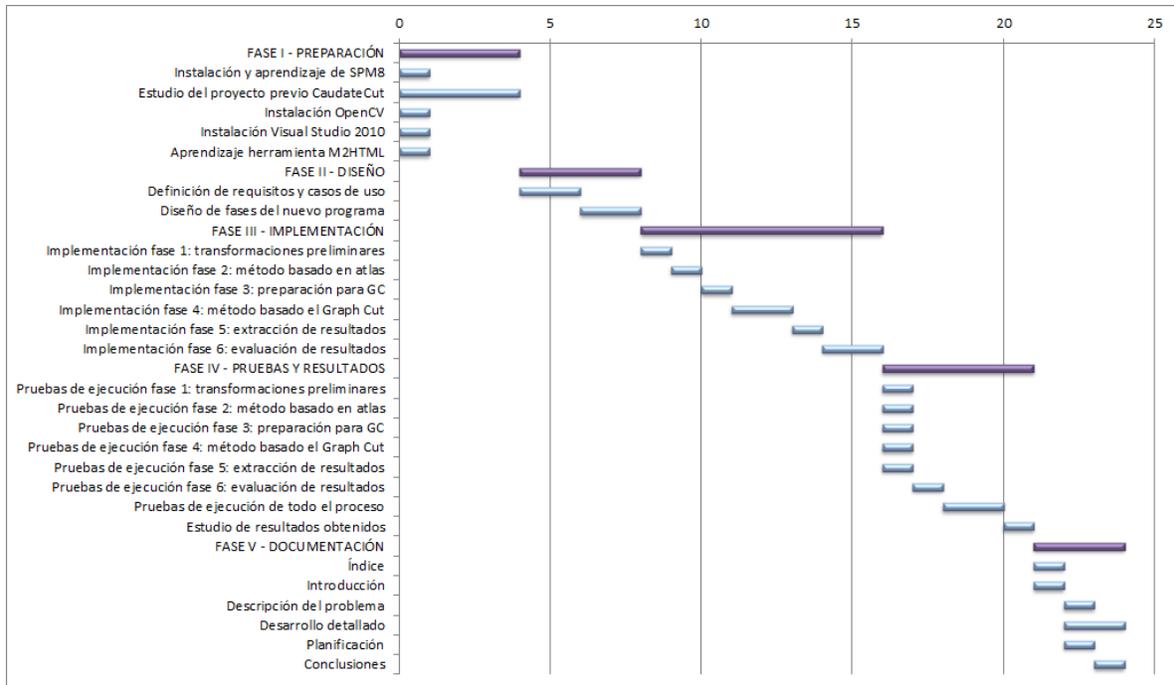


Figura 20: Diagrama de Gantt con todas las tareas planificadas.

5.3. Evaluación económica

Para poder hacer una evaluación económica y saber cuál sería el presupuesto del proyecto se deben definir los componentes necesarios para llevar a cabo esta propuesta. En cuanto a material necesario sería:

- Ordenador con RAM de 12 GB y CPU de mínimo 4 núcleos
- Licencia Matlab
- Licencia OpenCV
- Licencia Visual Studio 2010
- Licencia SPM8 En cuanto a la mano de obra
- Ingeniero del software
- Programador

En la Tabla 12 se muestra el coste económico que tendría el material, lo más caro sería el ordenador, teniendo en cuenta que cuanto más potentes sean la RAM y la CPU, mejor se trabajará, ya que hay que hacer muchos cálculos y de muchos sujetos. Es necesaria también la licencia para el Visual Studio para poder utilizar el programa incluido de Graph Cut en C++. En cuanto al gasto de la mano de obra viene reflejado en la Tabla 13 donde lo más caro es el pagar al programador a causa de las horas que requiere trabajar. Se ha hecho una estimación de trabajo de 4 horas diarias (en una jornada laboral de lunes a viernes). El Ingeniero de software necesitaría 1 mes (4 semanas) para hacer su trabajo, mientras que el programador necesitaría 13 semanas para poder llevar a cabo la implementación del software. En total, como se puede observar en la Tabla 14, el presupuesto total sería de 4.050,00 €.

<i>Material</i>	<i>Precio</i>
Ordenador con RAM 12GB y CPU 4 núcleos	800,00 €
Matlab	- €
OpenCV	- €
Visual Studio 2010	210,00 €
SPM8	- €
Total	1.010,00 €

Tabla 12: Presupuesto del material.

<i>Cargo</i>	<i>Horas</i>	<i>€ / Hora</i>	<i>Precio</i>
Ingeniero del software	80	12,00 €	960,00 €
Programador	260	8,00 €	2.080,00 €
Total			3.040,00 €

Tabla 13: Presupuesto mano de obra.

<i>Tipo de presupuesto</i>	<i>Precio</i>
Presupuesto material	1.010,00 €
Presupuesto mano de obra	3.040,00 €
Presupuesto total	4.050,00 €

Tabla 14: Presupuesto final.

6. Conclusiones

Al inicio de este trabajo las expectativas eran el poder crear un entorno que segmentara automáticamente estructuras subcorticales en IRMs. Esas expectativas se han cumplido, ya que se ha logrado alcanzar este objetivo utilizando técnicas punteras en visión por computador. En este capítulo se explicará que partes hemos conseguido realizar con éxito y cuáles pueden ser mejoradas.

6.1. Objetivos alcanzados

En cuanto al diseño de la aplicación los resultados obtenidos han sido los deseados, ya que se ha logrado llegar a el diseño modular necesario para que una vez implementado fuese fácil de mejorar y reutilizar por futuros investigadores.

En cuanto a la implementación de la aplicación, el fin principal era que segmentara las estructuras automáticamente de forma correcta. Como se puede apreciar en los resultados obtenidos, se ha logrado hacer esta segmentación con éxito. Los resultados se aproximan a la segmentación manual hecha por los médicos, obteniendo medidas de overlap mayores al 70%. Estos resultados llevan a contemplar la posibilidad de aplicar este método para el soporte al diagnóstico de TDAH en el futuro.

6.2. Mejoras posibles

Hay varias mejoras posibles, ya que este proyecto está enfocado en parte a facilitar la investigación, donde lo que se pretende es que este entorno no pare de evolucionar y mejorarse.

Una de las mejoras más importantes que se deben realizar es el de implementar nuevos métodos de segmentación que hagan esta tarea más eficaz, llegando a segmentar exactamente igual o mejor que puede hacerlo un médico. Uno de estos métodos sería el Graph Cut supervisado [2], el cual se podría

incluir en el entorno.

Otra mejora posible sería reducir el tiempo de CPU. El método basado en atlas es un método muy costoso que tarda bastante en ejecutarse. Lograr reducir el tiempo de CPU de este método sería una mejora muy importante. Por último sería interesante desarrollar una interfaz gráfica para esta aplicación, de modo que los médicos pudieran usarla para segmentar.

7. Anexos

7.1. Anexo A: Guía de usuario

7.1.1. Distribución de carpetas

Para una buena organización se requiere una buena distribución de carpetas, de este modo el usuario solo deberá acceder a algunas de ellas, haciendo todo mucho mas intuitivo y sencillo, y sin correr el riesgo de que se modifique el código. La distribución es la siguiente:

- **Doc**
Esta carpeta contiene toda la documentación en HTML de la aplicación.
- **Data**
Esta carpeta contiene todos los archivos y datos, tanto iniciales, temporales como finales. Divididos en:
 - **GT**
Carpeta contenedora de los Ground Truth. Estas son las IRMs segmentadas manualmente por médicos, y con las que se hará la comparación para ver cuán efectivo es el método.
 - **Input**
Esta es la carpeta donde el usuario colocará las IRMs a segmenta. En cualquier formato, tanto en formato .nii como .hdr.
 - **Output**
Esta carpeta contiene todos los resultados finales divididos en:
 - *Slices results*: Contiene todos los slices en .png de las IRMs con la segmentación dibujada.
 - *Volum results*: Contiene los volúmenes .nii con máscara de la segmentación final.
 - *Values results*: Contiene los resultados numéricos de todos los sujetos, comparando con el Ground Truth para ver cuánto se acerca a la realidad.
 - *Percent Gain*: Contiene el porcentaje de ganancia respecto al método basado en atlas únicamente (sin GC).
 - **Temporal _Output**
Contiene todos los resultados intermedios para la segmentación, divididos por fases. Los resultados temporales que crea cada fase se guardan en su carpeta correspondiente, estas son:
 - AB Segmentation results
 - Preparation GC Application results
 - Graph Cut unsupervised results
- **Code**
Contiene todos los scripts y funciones del código, con un prefijo que indica a que fase pertenece, siendo estas:
 - *C*: Copy
 - *ABS*: AB Segmentation
 - *PGC*: Preparation for GC Application
 - *GCu*: Graph Cut unsupervised
 - *ER*: Extract results
 - *E*: Evaluate

Tanto Doc como Data son las únicas carpetas que el usuario deberá y podrá manipular, en la carpeta Code solo deberá acceder el desarrollador

7.1.2. Fases del proceso

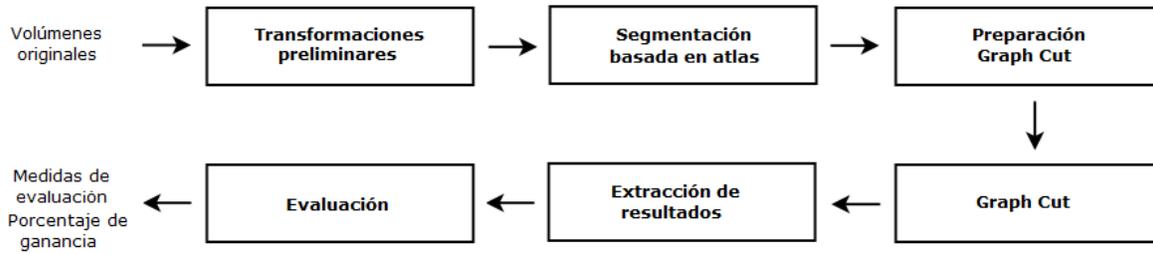


Figura 21: Fases del proceso.

En la figura 21 se puede observar como el proceso está dividido en seis fases explicadas a continuación:

■ 1. *Transformaciones preliminares*

Esta es la primera fase, lo que hace es coger todas las IRMs a segmentar de la carpeta “*Input*” y copiarlas en “*Temporal_Output*” con un seudónimo para la protección de datos, este seudónimo se guarda junto a su nombre original en un .xls. En caso de que las IRMs estén en formato .hdr las convierte en .nii ya que es en el formato que trabajamos. Estos nuevos volúmenes .nii almacenados en “*Temporal_Output*” son los que se emplearán para la segmentación.

■ 2. *ABSegmentation*

Aquí se ejecuta el método basado en atlas siguiendo los siguientes pasos:

- Segmenta la materia gris
- Pasa al espacio normalizado
- Segmenta la estructura
- Pasa la estructura segmentada al espacio original Todos estos resultados se guardan en la carpeta “*AB Segmentation Results*” de “*Temporal Output*”. Y estos resultados son de donde extraeremos las semillas en la siguiente fase.

■ 3. *Preparation for GC Application*

Esta fase está creada para prepararlo todo para aplicar el Graph Cut con la segmentación del AB que ya tenemos. Se divide con las siguientes partes:

- Extrae todas los cortes de los volúmenes, tanto los originales, los del Ground Truth y los resultantes de la segmentación basada en atlas.
- Corta la imagen rectangularmente conteniendo la estructura a segmentar (bounding-box) según los resultados la segmentación basada en atlas ya que será el trozo de imagen con el que se operará en el GC.
- Inicializa las semillas para el GC, haciendo erosión para las semillas de la estructura (foreground) y dilatación para las del resto del cerebro (background).

Todos estos resultados se almacenan en “*Preparation GC Application results*” de “*Temporal Output*” y son los que se emplearán en la siguiente fase.

■ 4. *Graph Cut unsupervised*

Una vez tenemos inicializadas las semillas, ya podemos crear nuestro grafo, y hacer la segmentación con Graph Cut no supervisado. Este Graph Cut se ejecuta con un ejecutable aparte, y para ello debemos definir todos los potenciales tanto unarios como de frontera y guardarlos en disco para poder ejecutar la aplicación. Así que los pasos de esta fase son los siguientes:

- Cálculo de los potenciales unarios
- Cálculo de los potenciales de frontera o vecindad
- Ejecución del ejecutable Graph Cut
- Extracción de los cortes resultantes

Los resultados de los potenciales serán almacenados en la carpeta “*Preparation GC Application results*” de la carpeta “*Temporal output*”, y los resultados finales en “*Graph Cut unsupervised Results*”. Con el Graph Cut finalizado ya podemos pasar a la siguiente fase, donde extraeremos y evaluaremos los resultados.

■ 5. *Extract results*

Aquí se extraen todos los resultados finales para guardarlos en “*Output*”

- Almacena los volúmenes con el resultado de la segmentación.
- Almacena los cortes originales con la segmentación dibujada.

■ 6. *Evaluate*

Esta es la última fase, lo que se quiere es evaluar los resultados obtenidos tanto con AB como con GC. Para ello se crea un .xls con las siguientes medidas de evaluación:

- **Volumetric overlap (en %):** Número de voxels en la intersección de la segmentación automática y de la segmentación manual, dividido por el número de voxels en la unión de ambas segmentaciones, el resultado se multiplica por 100. Este valor es 100 para una segmentación perfecta y tiene 0 como el valor más bajo posible, cuando no hay superposición en absoluto entre ambas segmentaciones.
- **Volumetric similarity index (en %):** Intersección de las segmentaciones partido el volumen medio de ambas. 100 para una segmentación perfecta y tiene 0 como el valor más bajo posible.
- **Relative absolute volume difference (en %):** El volumen total de la segmentación automática se divide por el volumen total de la segmentación manual. A partir de este número se resta 1, el valor absoluto será tomado, y el resultado se multiplica por 100. Este valor es 0 para una segmentación perfecta y mayor que cero en caso contrario. El valor perfecto de 0 también se puede obtener para una segmentación no perfecta, siempre y cuando el volumen de que la segmentación es igual al volumen de la referencia.
- **Average symmetric absolute surface distance (en mm):** Se determinan los voxels fronterizos en ambas segmentaciones (auto y manual). Para cada voxel en estos conjuntos, se determina el voxel más cercano en el otro conjunto (utilizando la distancia euclídea). Todas estas distancias se almacenan, para los voxels fronterizos de ambas segmentaciones. El promedio de todas estas distancias da los promedios distancia de la superficie absoluta simétrica. Este valor es 0 para una segmentación perfecta.
- **Symmetric RMS (Root Mean Square) surface distance (en mm):** Esta medida es similar a la medida anterior, pero guarda las distancias al cuadrado entre los dos grupos de voxels fronterizos. Se extrae la raíz del promedio de los valores al cuadrado y obtenemos el valor final. Este valor es 0 para una segmentación perfecta.
- **Maximum symmetric absolute surface distance (en mm):** Se toma el máximo valor de las distancias calculadas en las medidas anteriores. Este valor es 0 para una segmentación perfecta.

Se guardan estas medidas en el .xls para cada uno de los sujetos, junto a la desviación estándar y la media de todos ellos. Se crea también un .xls con el porcentaje de ganancia de volumetric overlap y volumetric similarity index respecto el método basado en atlas, para cada uno de los sujetos y para la media. De este modo podemos ver si hemos logrado mejorar la segmentación

7.1.3. Información general para el uso de la aplicación

Para poder ejecutar la aplicación hay que hacer unos pasos preliminares.

- Hay que añadir la librería OpenCV a las variables de entorno, para ello desde Windows hay que ir a: Mi PC → botón derecho → Propiedades → Configuración avanzada del sistema → Variables de entorno → Buscar variable 'Path' → Editar. Y se añade al final de la línea (sin espacios):

```
“;<Path OpenCV>\Release;<Path OpenCV>\Debug”
```

Donde <Path OpenCV> es la ruta donde está la carpeta “*OpenCV 2.1 bin*”, de la carpeta “*Code*” del código. Por ejemplo si el código está en: “C:” se debería añadir al final de la línea:
“;C:\BrainCut\Code\OpenCV 2.1 bin\Release;C:\BrainCut\Code\OpenCV 2.1 bin\Debug”

- Se añaden todas las imagines “.nii o “.hdr” que se quieran segmentar dentro de la carpeta “BrainCut\Data\Input”
- Se añade (si es preciso) todas las imágenes segmentadas manualmente dentro de la carpeta “BrainCut\Data\GT”
- Por último para indicar las estructuras que se quieren segmentar, hay que abrir el archivo “BrainCut\Code\create_parameters.m” y en el campo “parameters.structures” poner todas las estructuras a segmentar, por ejemplo por defecto están ambos caudados:

```
parameters.structures =['Caudate_L'; 'Caudate_R'];
```

Referencias

- [1] Wikipedia: <http://es.wikipediHernandez-Velaa.org/wiki/Wikipedia:Portada>
- [2] L. Igual, J. C. Soliva, A. Hernandez-Vela, S. Escalera, X. Jimenez, O. Vilarroya, and P. Radeva. A fully-automatic caudate nucleus segmentation of brain MRI: application in volumetric analysis of pediatric attention-deficit/hyperactivity disorder. Biomed Eng Online, 2011.
- [3] http://kidshealth.org/parent/en_espanol/medicos/mri_brain_esp.html
- [4] <http://queaprendemos hoy.com/como-funciona-una-resonancia-magnetica>
- [5] <http://ocw.um.es/cc.-sociales/neuropsicologia/material-de-clase-1/tema-2.-organizacion-y-funcion-cerebral.pdf>
- [6] <http://lasaludi.info/del-nucleo-caudado.html>
- [7] http://www.psicologia-online.com/ebooks/general/ganglios_basales.htm
- [8] <http://www.mathworks.es/products/matlab>
- [9] <http://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatismatlab.htm>
- [10] <http://www.artefact.tk/software/matlab/m2html>
- [11] <http://lts08.bigr.nl/about.php>