PRÁCTICA 1: FILTRADO Y MEJORA DE IMAGEN MÉDICA

El objetivo de la práctica es evaluar tres algoritmos de filtrado de imagen médica, y la creación de un mapa paramétrico a partir de una serie de imágenes. El objetivo de este tipo de técnicas es mejorar la calidad de la imagen, o conseguir una nueva imagen a partir de varias imágenes de partida

1. Filtrado anisotrópico por difusión

Como se vio en clase, el objetivo de los filtros anisotrópicos es eliminar el ruido en zonas homogéneas de la imagen (objetos), sin suavizar los contornos entre las regiones. En el documento DifusionIntro.pdf se puede leer una introducción más detallada a estos métodos.

La función dif_aniso permite realizar un filtrado de Perona-Malik (modelo más empleado de filtrado) con Octave/Matlab. El objetivo de la práctica es probar los filtrados, así como entender el concepto y la implementación del mismo.

Se deben cargar las imágenes T1.png y T2.png, y realizar los siguientes pasos:

1) Encontrar los parámetros del filtro que permiten un filtrado correcto de cada una de las imágenes. Esto se puede hacer de forma experimental, o realizando medidas sobre el valor de los gradientes en los contornos de la imagen. Comentar el efecto de cada parámetro sobre el resultado.

En el artículo Gerig92.pdf, además de realizar una revisión de la metodología de filtrado, en el apartado 2.7.2 se explica como ampliar el filtrado al caso de dos imágenes (información multicanal).

2) Modificar el código para poder realizar el filtrado conjunto de las dos imágenes T1 y T2, y comparar el resultado del filtrado conjunto con el del filtrado de un solo canal.

2. Filtrado no local de media

EL filtrado non-local means consiste en sustituir el valor de cada píxel por la media de los k píxeles con un vecindario más parecido al vecindario del píxel inicial.

1) Implementar una función de filtrado non-local means con el tamaño de vecindario como parámetro variable. Para comparar vecindarios, se puede emplear la suma de diferencias cuadradas.
2) Comparar el resultado de filtrar la imagen T1.png con este filtrado, un filtrado gaussiano, y el filtro de Perona-Malik.

3. Corrección de inhomogeneidad

Las imágenes de Resonancia Magnética, debido al campo de radiación de la antena con la que se adquieren, pueden presentar una iluminación no homogénea a lo largo del volumen.

El objetivo de este apartado es la corrección del mismo, empleando la versión del algoritmo N4, proporcionado por el software 3D Slicer.

Para ello:

1) Cargar la imagen brain.nii en 3D slicer
2) Ir al módulo MRI Bias Field Correction (dentro del grupo de módulos Filtering)
3) Corregir el artefacto de inhomogeneidad con el algoritmo N4 analizando los distintos parámetros. Comentar el efecto de los distintos parámetros sobre el resultado

4. Cálculo de mapas de hierro del cerebro.

Recientemente, se han encontrado evidencias de la presencia anormal de hierro en el cerebro de pacientes con Alzheimer y otras patologías neurodegenerativas. Por ello, sería interesante poder medir la presencia de hierro de forma no invasiva.

Usando un equipo de resonancia magnética, mediante la adquisición con una secuencia especifica en distintos tiempos y el análisis de la imagen podemos obtener la velocidad con la que se produce la relajación en T2 de los protones de hidrógeno en forma de una exponencial. Debido a las características de las partículas de hierro, su presencia lleva a que esta exponencial decaiga más rápido de lo normal.

Para obtener el valor de T2, necesitamos conocer la ecuación exponencial (formula 1) que describe la caída de la intensidad de esta señal y ajustar mediante los puntos de las imágenes otra exponencial con las mismas características en cuanto a la velocidad de decaimiento.

$$p \cdot e^{\left(\frac{-TR}{T_2}\right)} \cdot \left(1 - e^{\left(\frac{-TE_i}{T_2}\right)}\right) \quad [1]$$

Obteniendo las imágenes con un tiempo de repetición (TR) cinco veces mayor que el valor de T1, podremos simplificar la formula que pasa a ser una exponencial con exponente negativo como describe la formula 2.

$$S_i(t) = S_0 \cdot e^{\left(\frac{-TE_i}{T_2}\right)} \quad [2]$$

Obteniendo de cada voxel de un mismo corte en distintos tiempos la tupla intensidad y tiempo, podemos representar una gráfica con las mismas características, de modo que usando mínimos cuadrados obtengamos la exponencial (formula 3) que mejor se ajusta a nuestros puntos.

$$y = A \cdot e^{(B \cdot x)} \quad [3]$$

De este modo, aplicando logaritmos, minimizando los errores y despejando podemos obtener A y B, donde A sería igual a S_0 y B a –R2 o lo que es lo mismo, la inversa de T2.

El objetivo de este apartado consiste en calcular un mapa de T2 a partir de una serie de imágenes de resonancia magnética adquiridas con distintos tiempos de eco.

1) Las imágenes slice_TE*.tif son el mismo corte de un cerebro adquirido con los valores de TE que indica el nombre del archivo. Usando Octave/Matlab, el objetivo es ajustar a cada píxel de la imagen una exponencial, y obtener una imagen que represente el parámetro B en cada píxel.

2) *Desarrollo opcional*: Los volúmenes vol_TE*.nii son los volúmenes completos adquiridos con distintos tiempos de eco. El objetivo es cargarlos en el 3D slicer, y programando en python, realizar el ajuste de la exponencial en cada vóxel.

En el caso de tener Matlab disponible, se puede realizar el mismo proceso. En el enlace http://www.mathworks.com/matlabcentral/fileexchange/8797 se puede bajar un toolbox que permite leer, visualizar y escribir imágenes en formato NIFTI.