

TESIS DE LICENCIATURA

# **Detección de Contornos y Wavelets de Enteros a Enteros para la Compresión de Imágenes de Resonancia Magnética (MRI)**

Sebastián Diego Masi

[sebam@fibertel.com.ar](mailto:sebam@fibertel.com.ar)

Directora: Dra. Ana Ruedin  
Codirector: Daniel Acevedo



Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires



Noviembre de 2012

# Resumen

La compresión de imágenes de resonancia magnética es sumamente importante para disminuir costos de almacenamiento y tiempos de transmisión, facilitando así las tareas de teledetección.

Debido al ruido existente en este tipo de imágenes, no se han conseguido buenos resultados utilizando compresores basados en entropía.

Los mejores resultados hasta el momento se han obtenido separando el fondo de la figura. Esta última contiene la información que es relevante para el diagnóstico. Luego se descarta el fondo y se aplican técnicas de compresión sin pérdida a la figura. Para ello es indispensable contar con un método de segmentación automática que sea robusto.

En este trabajo se utilizó una variante de la segmentación por conjuntos de nivel. A partir de un contorno inicial arbitrario en los tres ejes dimensionales de la imagen, se obtiene la máscara en forma iterativa. Este método ha demostrado ser robusto frente al ruido de la imagen.

Posteriormente, se realizó la compresión sin pérdida de la figura detectada. Para ello, se utilizó primero un algoritmo de reconocimiento de bloques similares usado en compresores de video. Con este algoritmo se identificaron bloques similares entre un corte de la imagen y su predecesor y se calcularon las diferencias entre los mismos. Luego se transformaron las diferencias de bloques obtenidas con la versión de dos dimensiones de la transformada wavelet de enteros a enteros CDF (2,2).

Para la codificación final se utilizó un codificador aritmético adaptativo. Sobre un conjunto de 15 secuencias de imágenes de resonancia magnética se logró en promedio un tasa de compresión de 5.15 a 1.

# Agradecimientos

A mi directora Ana Ruedin, por la paciencia y por abrirme la puerta al mundo de la compresión de datos.

A mis compañeros de facultad, Pablo F, Soledad, Pablo S, Guillermo, Ariel, Claudio, Hernán y Marcelo, por acompañarme y ayudarme, en mayor o menor medida, durante esta carrera.

A mis profesores del secundario Javier Cachenet, Sergio Ióppolo, Jorge Bottaro y Pablo Della Paolera, quienes me enseñaron a programar y me iniciaron en lo que es sin dudas mi vocación. Y a la profesora María del Carmen Palermo, por la excelente educación que me brindó en mi adolescencia.

A Diego, Juampi, Vale, Vir, Iru, Fer, Juancho y Puchito, mis amigos de la infancia a quienes volví a encontrar después de tanto tiempo.

A mis amigos Emilio, Matías, Demián, Cecilia y Fefo (que siempre estará en mi recuerdo), por bancarme en los momentos más difíciles de mi vida y ayudarme a salir adelante.

A mi tío Darío y mis primos Gabriela, Mariano y Martín, por apoyarme siempre. A Valeria, por enseñarme a leer y por ser la hermana mayor que no tengo.

A mi abuela Chichí por criarme, a mi abuela Beba por malcriarme, y a mi abuelo Fresno por ser el mejor ejemplo de moral que jamás haya podido imaginar. Los extraño a los tres.

A mi hermana Jimena, por ser mi apoyo permanente y por demostrarme el valor de la perseverancia y de no rendirse jamás.

A mi vieja, por su fuerza, su coraje, su empuje y su determinación en todos los aspectos de la vida y por ser mi red de contención.

Y fundamentalmente a mi viejo, que dedicó toda su vida para que nunca nos faltara nada. Estarás siempre presente en mi corazón, te quiero.

# Índice

I.	Introducción .....	6
1.	Medidas de Información .....	8
2.	Características de los Estudios MRI .....	9
II.	Técnicas Utilizadas .....	11
1.	Conjuntos de Nivel .....	11
2.	Reconocimiento de Bloques para Estimación de Movimiento .....	13
3.	Transformada wavelet de enteros en enteros CDF (2,2).....	15
4.	Codificador aritmético adaptativo.....	17
III.	Método propuesto .....	18
1.	Esquema de Compresión.....	18
2.	Definiciones Previas .....	19
3.	Segmentación por Detección de Contornos.....	21
3.1.	Selección del Contorno Inicial .....	21
3.2.	Selección del Contorno Inicial .....	23
3.2.	Generación de la Máscara de Segmentación.....	25
3.2.	Robustización del Método.....	26
4.	Estimación de Movimiento .....	28
4.1	Búsqueda Exhaustiva de Bloques Coincidentes .....	28
4.2	Codificación de Bloques .....	28
4.3	Arreglos Resultantes .....	29
5.	Procesamiento de los Bloques I y P.....	30
5.1	Aplicación de la Transformada CDF(2,2).....	30
5.2	Codificación Aritmética .....	30
6.	Procesamiento de los Vectores de Desplazamiento.....	31
6.1	Cálculo de las Diferencias.....	31
6.2	Codificación Aritmética .....	31
7.	Datos auxiliares.....	32
8.	Descompresión.....	33
IV.	Resultados.....	34
1.	Implementación .....	34
2.	Datos de prueba.....	34

3. Tiempo de ejecución .....	35
4. Compresión .....	36
5. Comparación con otros métodos.....	38
6. Discusión .....	39
6.1 Parámetros de compresión .....	39
6.2 Suavizado del corte previo a la estimación de movimiento .....	41
7. Conclusiones .....	42
Referencias .....	43

# I. Introducción

Las imágenes de resonancia magnética son en la actualidad una herramienta fundamental en la detección y el diagnóstico de distintos tipos de patologías. Estas imágenes son el resultado de estudios muy costosos y que no pueden ser repetidos debido a los cambios que pueden haberse producido en la mencionada patología desde el momento de realización del estudio. Es por esto que la compresión de estas imágenes cobra vital importancia para facilitar tareas de teledetección y, al mismo tiempo, ahorrar costos de almacenamiento. En este contexto además la compresión debe realizarse sin pérdida de información. Las MRI presentan como característica la existencia de una figura bien definida, en la cual se encuentra la totalidad de la información relevante para realizar el diagnóstico. Dicha figura se encuentra habitualmente rodeada por un fondo totalmente ruidoso.

En general, el ruido existente en este tipo de imágenes impide que la obtención de buenos resultados utilizando compresores basados en entropía [1]. En la actualidad existen diversas técnicas alternativas que han sido propuestas con el objetivo de lograr mayores tasas de compresión sin pérdidas de información.

Una de estas técnicas [2] considera la similitud entre cortes sucesivos de un estudio MRI como información redundante y propone la preservación sólo de algunos de los cortes del estudio y el uso estadístico del método de Análisis de Componente Principal (o Principal Component Analysis, PCA) para crear un espacio de proyección de la imagen. Esta técnica presenta la posibilidad de señalar cortes específicos del estudio como regiones de interés, forzando de esta forma que dichos cortes sean preservados e incluidos en el PCA. La principales desventajas de tal técnica son que requiere la intervención de un especialista para señalar los cortes que se desea preservar como regiones de interés y que se trata de un método que es sólo parcialmente sin pérdida de información, ya que ciertos cortes del estudio, si no son señalados, son completamente descartados.

Hasta el momento los mejores resultados de compresión sin pérdida de información se han obtenido aplicando un proceso que permite separar el fondo, inherentemente ruidoso, de la figura principal de la imagen, la cual contiene toda la información relevante de la misma. Una vez realizada esta separación, se descarta el fondo y se aplican técnicas de compresión sin pérdida a la figura.

Las imágenes de resonancia magnética poseen, en su gran mayoría, un alto nivel de ruido, el cual dificulta los procesos de compresión basados en entropía [1] que se usan habitualmente para comprimir otros tipos de imágenes. En busca de obtener mejores resultados de compresión, se han propuesto diversos mecanismos para segmentar la figura.

Como se ha mencionado, la información relevante de la imagen suele encontrarse concentrada en una figura rodeada por un fondo mayormente ruidoso. Los procesos de segmentación intentan utilizar dicha característica para separar la figura del fondo. Esto permite luego descartar totalmente el fondo y centrar los métodos de compresión, que deben ser sin pérdida, en la figura.

Para realizar la segmentación se han presentado diversos métodos entre los que se incluyen crecimiento de regiones [3], [4] y técnicas estadísticas basadas en histogramas [5], [6] complementados con operaciones morfológicas [7]. Recientemente, Corvetto y otros autores [8], [9], basando su trabajo en el de Brummer y Srikanth [7] y otros, han propuesto un algoritmo que utiliza histogramas y operaciones morfológicas, a los cuales agregaron una estrategia de votación después de segmentar la imagen en cada una de sus dimensiones.

En el presente documento, se propone la utilización de una variante más robusta del método mencionado, utilizando la segmentación por conjuntos de nivel para realizar la separación mencionada anteriormente. Se presentará un método basado en la detección de contornos mediante la utilización de técnicas de conjuntos de nivel. Partiendo de un contorno inicial arbitrario, se utilizan conjuntos de nivel para ir refinando iterativamente dicho contorno hasta que el mismo se adapte a la figura, diferenciándose así del método de segmentación basado en histogramas propuesto por Corvetto [9] en otra tesis de esta misma facultad.

Luego de realizar la segmentación de antes descripta, se propone comprimir la figura utilizando la técnica de estimación de movimiento mediante reconocimiento de bloques, muy utilizada en estándares de compresión de secuencias de video como MPEG1 / H.261 o MPEG4 / H.263. [10]

Por último, se propone la aplicación de una transformada wavelet bidimensional de enteros en enteros [11] a cada uno de los bloques obtenidos en el paso anterior y codificar los resultados utilizando un codificador aritmético adaptativo.

Este proceso de compresión y codificación también difiere del propuesto por Corvetto [9] en su tesis, en el cual se utiliza el algoritmo de cuantización k-medias para luego vectorizar los arreglos resultantes y aplicar una transformada wavelet unidimensional antes de la codificación final.

En la presente sección se plantearán los objetivos del trabajo, se indicarán las características principales de los estudios de resonancia magnética (MRI), y se enunciarán algunas nociones básicas relativas a la compresión de datos. En la sección II se presentarán brevemente las técnicas utilizadas durante el desarrollo del trabajo. En la sección III se explicará paso a paso el método de compresión propuesto y, por último, en la sección IV se expondrán los resultados obtenidos durante las pruebas del método, junto a las conclusiones, comparaciones con otros métodos e ideas para poner en práctica a futuro.

# 1. Medidas de Información

A continuación se definirán algunas de las convenciones sobre las medidas de información y compresión utilizadas para estimar el desempeño del método propuesto.

**Fuente de memoria nula:** Una fuente de memoria nula es un par ordenado  $F = (S, P)$ , donde  $S = s_1, s_2, \dots, s_n$  es un conjunto finito de símbolos (denominado alfabeto de la fuente), y  $P : S \rightarrow [0, 1]$  es la distribución de probabilidades de los elementos de  $S$ . La probabilidad de un determinado símbolo  $s_i$  se escribe  $P(s_i)$  o  $p_i$ .

**Entropía:** Se define la entropía  $h_b(F)$  de una fuente  $F = (S, P)$  como

$$h_b(S) = - \sum_{i=1}^n p_i \log_b p_i = \sum_{i=1}^n p_i \log_b \frac{1}{p_i}$$

donde el valor de  $b$  determina la unidad de información. Por ejemplo, si  $b = 2$ , entonces la entropía es medida en bits. La entropía sirve para estimar la cantidad de bits por símbolo, en promedio, que llevará la compresión de una fuente.

**Tasa de compresión:** La tasa de compresión  $TC$  de una determinada fuente finita de información se define como:

$$TC = \frac{o}{c}$$

donde  $o$  es el tamaño de la fuente en su formato original y  $c$  es el tamaño de la fuente una vez comprimida. De esta manera, una compresión a la mitad se nota "2: 1" y una compresión a una cuarta parte "4: 1".



## 2. Características de los Estudios MRI

Como se ha mencionado en la introducción, los estudios de resonancia magnética (MRI) suelen presentar altos niveles de ruido, lo cual dificulta su comprensión a través de codificadores basados en entropía.

Como también se ha mencionado, otra característica fundamental de los estudios MRI es la existencia siempre de una figura perfectamente definida rodeada de por un fondo que, si bien a primera vista puede parecer uniforme, es en realidad extremadamente ruidoso. Esto se puede observar si tomamos un corte cualquiera de un estudio MRI y procedemos a aislar el fondo.

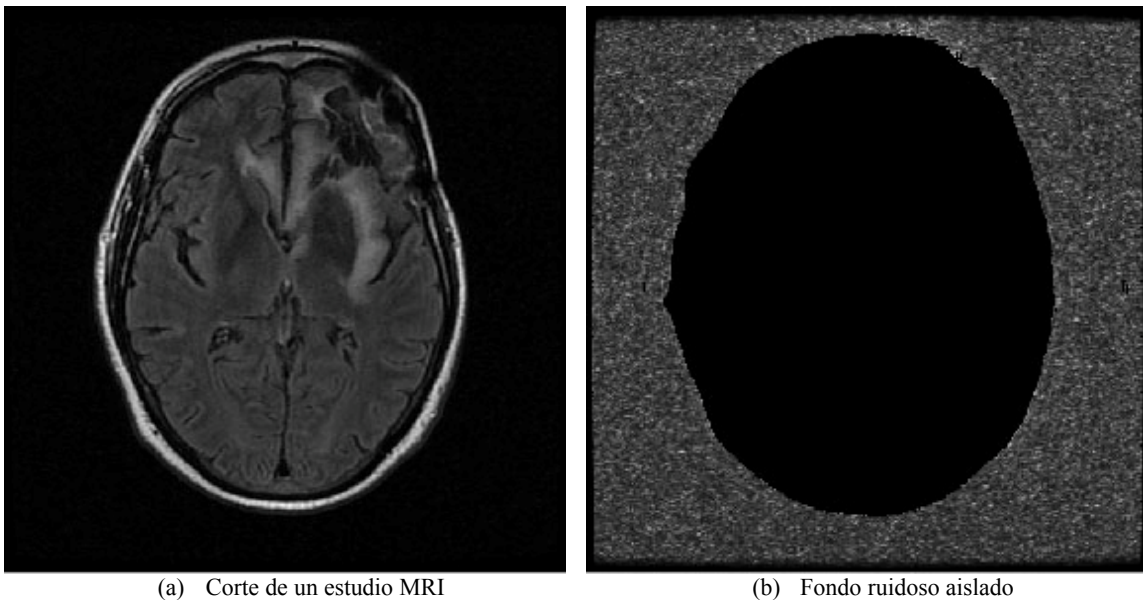


Fig. 1. Aislación de fondo en un estudio MRI

Se ha estudiado la distribución del ruido en las imágenes de resonancia magnética y las formas en que puede ser filtrado [12]. En general, el ruido en dichas imágenes responde a una distribución Rician, que es dependiente de la señal, haciendo muy difícil el trabajo de filtrado. En el caso del fondo, al no haber señal, se puede modelar el ruido como una distribución Rayleigh [5].

$$p_X(x) = \frac{x}{\sigma^2} e^{-(x^2/2\sigma^2)}$$

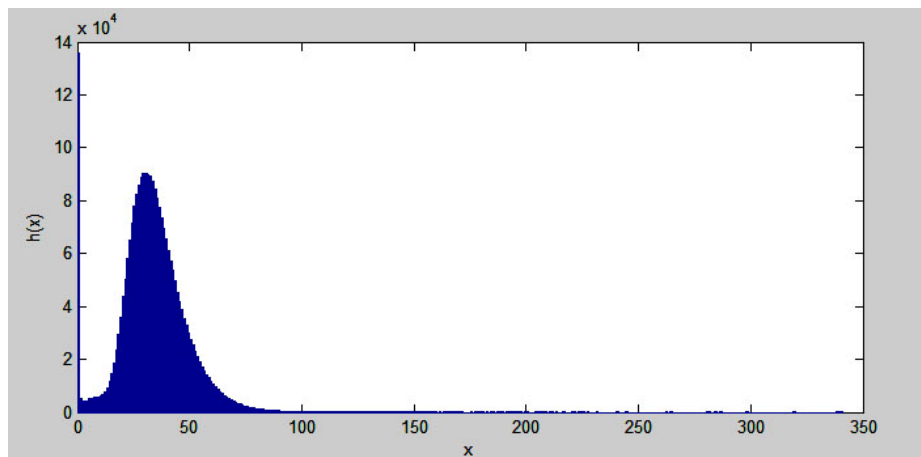


Fig. 2 Distribución empírica del fondo ruidoso

## II. Técnicas Utilizadas

### 1. Conjuntos de Nivel

Los métodos de Conjuntos de Nivel (Level-Sets) son técnicas numéricas inicialmente utilizadas para el seguimiento de formas y objetos en movimiento o cuya topología es variable [13]. Posteriormente se comenzó a utilizar estos métodos para la segmentación de imágenes facilitando la detección de contornos geoméricamente activos [14], [15], [16], representados como conjuntos de nivel de una función bidimensional que evoluciona dentro de un sistema Euleriano.

La idea básica es representar los contornos como el conjunto de nivel cero de una función implícita definida en una dimensión superior, y hacer evolucionar dicha función de acuerdo a una ecuación diferencial parcial [17].

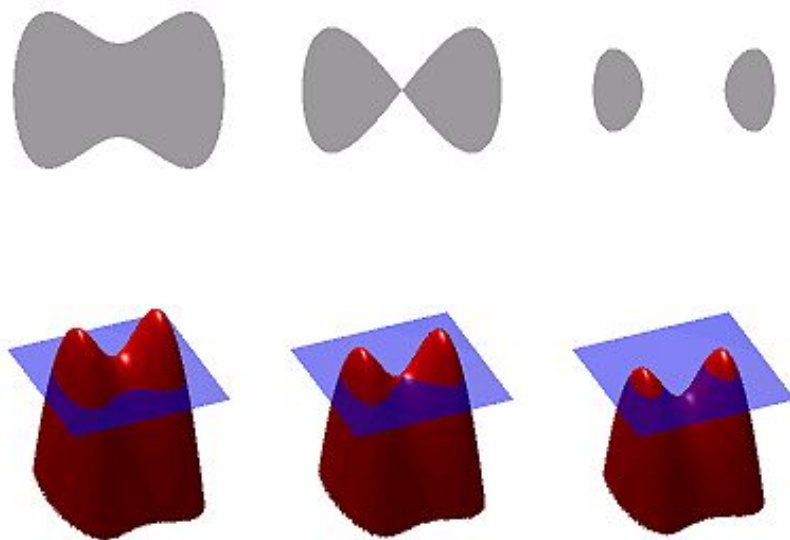


Fig. 3 Ilustración en la cual se puede apreciar gráficamente como el conjunto de nivel cero de la función de conjuntos de nivel (en rojo) aproxima los contornos de las formas dibujadas en gris.

En la formulación original de conjuntos de nivel para contornos activos, las formas, denotadas por  $C$ , son representadas por el conjunto de nivel cero  $C(t) = \{(x, y) \mid \varphi(t, x, y) = 0\}$  de una función de conjuntos de nivel  $\varphi(t, x, y)$ . Según el modelo, esta función es positiva en el interior de la curva, y negativa fuera de ella. La ecuación de evolución de la función  $\varphi$  puede ser escrita en la siguiente forma general:

$$\frac{\partial \phi}{\partial t} + F|\nabla \phi| = 0$$

la cual es llamada *ecuación de conjuntos de nivel* [13]. La función  $F$  es llamada función velocidad. Para la segmentación de imágenes, la función velocidad depende de la información de la imagen y de la función de conjuntos de nivel  $\phi$ .

En los métodos de conjuntos de nivel tradicionales [14], [15], [16], [18], la función  $\phi$  puede desarrollar cambios bruscos y formas demasiado agudas o demasiado planas durante la evolución. Lo cual hace altamente imprecisa la computación posterior. Para evitar estos problemas se inicializa la función  $\phi$  como una función de distancia con signo antes de la evolución, y luego se “reforma” (o “reinicializa”) la función  $\phi$  durante la evolución encontrando una nueva función que posea el mismo nivel cero, pero en la cual  $|\nabla \phi| = 1$ .

El método de reinicialización que habitualmente se utiliza consiste en resolver la ecuación

$$\frac{\partial \phi}{\partial t} = \text{signo}(\phi_0)(1 - |\nabla \phi|)$$

donde  $\phi_0$  es la función a reinicializar.

## 2. Reconocimiento de Bloques para Estimación de Movimiento

En la compresión de video son comúnmente utilizados diversos algoritmos de estimación de movimiento a través del reconocimiento de bloques similares entre cuadros de video consecutivos.

Esta técnica se basa en la suposición de que los objetos, y el fondo detrás de los mismos, existentes en un cuadro de una secuencia de video son los mismos que aparecen en el cuadro subsiguiente, sólo que se han movido y/o desplazado ligeramente. La idea detrás del reconocimiento de bloques es dividir el cuadro que se desea procesar en “macro bloques” que son entonces comparados con su bloque correspondiente y con un entorno del mismo en el cuadro anterior, de manera de crear un vector de movimiento que estima el movimiento de una ubicación a otra en cada cuadro. Este movimiento calculado para todos los macro bloques que componen un cuadro constituyen la estimación de movimiento del cuadro en cuestión. El área de búsqueda dentro de la cual se intenta detectar una buena coincidencia entre macro bloques está limitado a  $p$  pixels hacia cada uno de los cuatro lados del macro bloque correspondiente en el cuadro anterior. Este valor  $p$  se conoce como parámetro de búsqueda. Usualmente se utilizan macro bloques de 16 pixels de lado y se utiliza un parámetro de búsqueda  $p$  de 7 pixels.

La similitud o no de un macro bloque con otro de igual tamaño en el cuadro anterior está basada en la medida de error entre ambos. El macro bloque que minimiza el error es aquel que se asemeja más al bloque que se está procesando. Existen varias funciones para calcular la medida de error mencionada, de las cuales la más popular y menos costosa en términos de computabilidad es la Diferencia Absoluta Promedio (Mean Absolute Difference. MAD) dada por la siguiente ecuación:

$$MAD = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |A_{ij} - R_{ij}|$$

en donde  $N$  es el lado del macro bloque, mientras que  $A_{ij}$  y  $R_{ij}$  son los pixels que están siendo comparados entre el bloque actual y el bloque de referencia, respectivamente.

Si bien existen diversos algoritmos de reconocimiento de bloques, en el presente trabajo se ha utilizado el conocido como Búsqueda Exhaustiva (o también Búsqueda Completa). Se trata del algoritmo más costoso computacionalmente de todos los existentes, ya que calcula la medida de error para cada posible ubicación dentro de la ventana de búsqueda. Como resultado de esto, este algoritmo asegura encontrar la mejor coincidencia posible. [10]

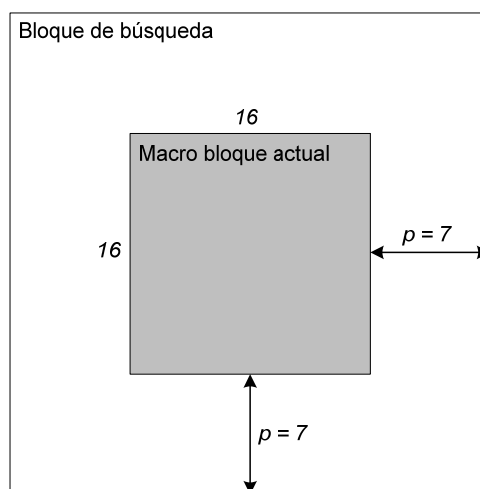


Fig. 4 Reconocimiento de bloques con un macro bloque de 16 pixels y un parámetro de búsqueda de 7 pixels

### 3. Transformada wavelet de enteros en enteros CDF (2,2)

La transformada wavelet de enteros en enteros CDF (2,2) biortogonal, también conocida como CDF(2,2), CDF(5,3) o LeGall (5/3), pertenece a la histórica primera familia de Wavelets Biortogonales, que fue presentada por Ingrid Daubechies. Esta transformada es utilizada por el método de compresión JPEG2000 para realizar compresión sin pérdida de información.

La wavelet Le Gall 5/3 utiliza dos filtros pasa bajos  $h'$  (para la transformada) y  $h$  (para la transformada inversa) y dos filtros pasa altos  $g'$  (para la transformada) y  $g$  (para la transformada inversa). Estos filtros son definidos como:

$$h' = \frac{1}{4\sqrt{2}} [-1 \quad 2 \quad 6 \quad 2 \quad -1]$$

$$h = \frac{1}{2\sqrt{2}} [-1 \quad 2 \quad 1]$$

$$g' = \frac{1}{2\sqrt{2}} [1 \quad -2 \quad 1]$$

$$g = \frac{1}{4\sqrt{2}} [1 \quad 2 \quad -6 \quad 2 \quad 1]$$

Durante la etapa de análisis se realiza una transformación mediante la utilización del filtro pasa altos  $h'$  y el filtro pasa bajos  $g'$  sobre la imagen a comprimir y la posterior aplicación de un sub-muestreo a los resultados obtenidos. La etapa de síntesis, por su parte, implica la realización primero de un sobre-muestreo y la utilización luego del filtro pasa altos  $h$  y el filtro pasa bajos  $g$  para obtener nuevamente la señal original.

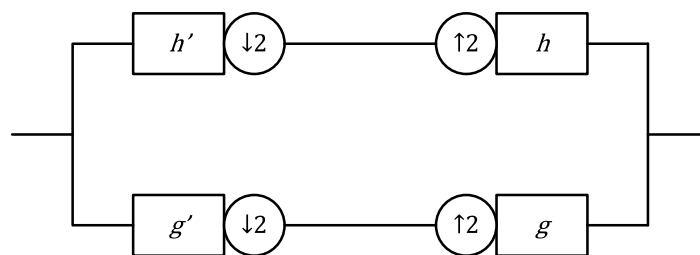


Fig. 5 Esquema de transformada y transformada inversa.

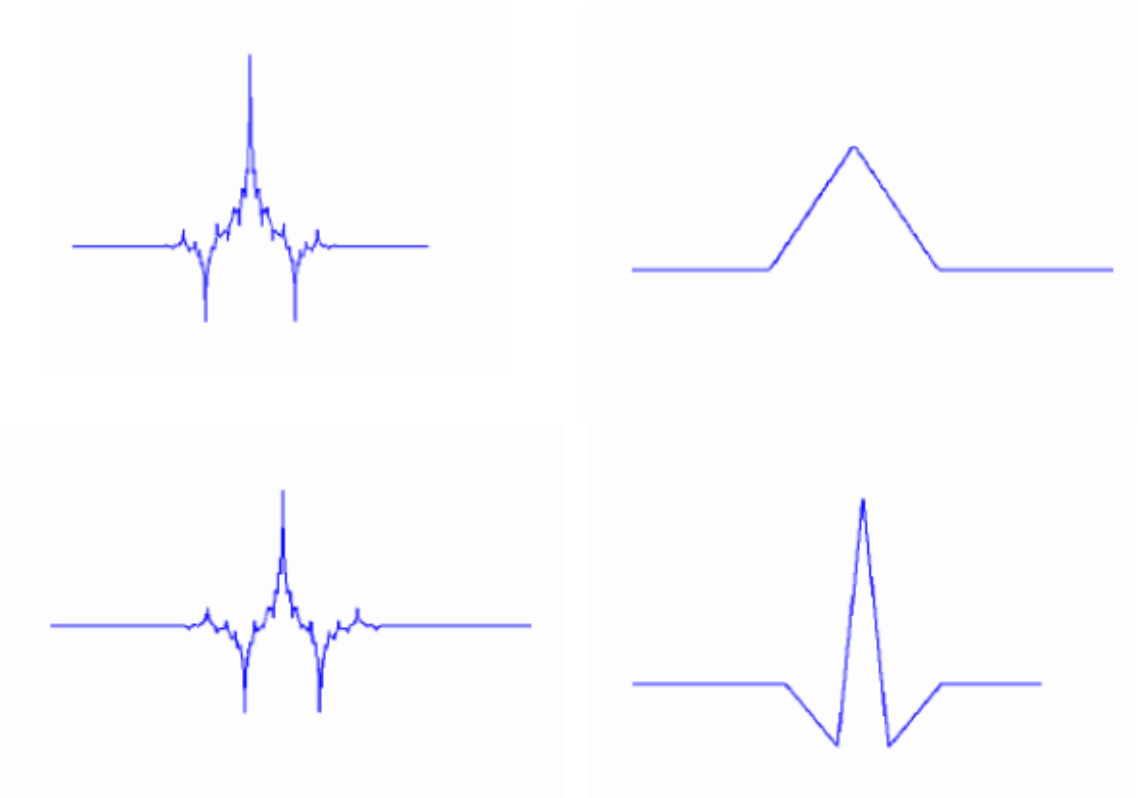


Fig. 6 Representación gráfica de los filtros utilizados



## 4. Codificador aritmético adaptativo

Se utilizará un codificador aritmético adaptativo [19]. Este tipo de codificador representa la información de forma más compacta que una codificación por Huffman en la mayoría de los casos [20], representando las cadenas de símbolos como números reales. Cada símbolo es representado por un intervalo real. La longitud del intervalo para cada símbolo depende de su frecuencia en la fuente.

Es posible utilizarlo con un modelo estático, definiendo las frecuencias de los símbolos antes de comenzar la codificación, o utilizando un modelo adaptativo, en el que las frecuencias se actualizan a medida que se lee la fuente. En este último caso, el tamaño final de la cadena comprimida depende del orden en que se encuentran los símbolos en la fuente.

# III. Método propuesto

## 1. Esquema de Compresión

La siguiente figura presenta los pasos del compresor propuesto. En las secciones subsiguientes se detalla cada una de las etapas del mismo.

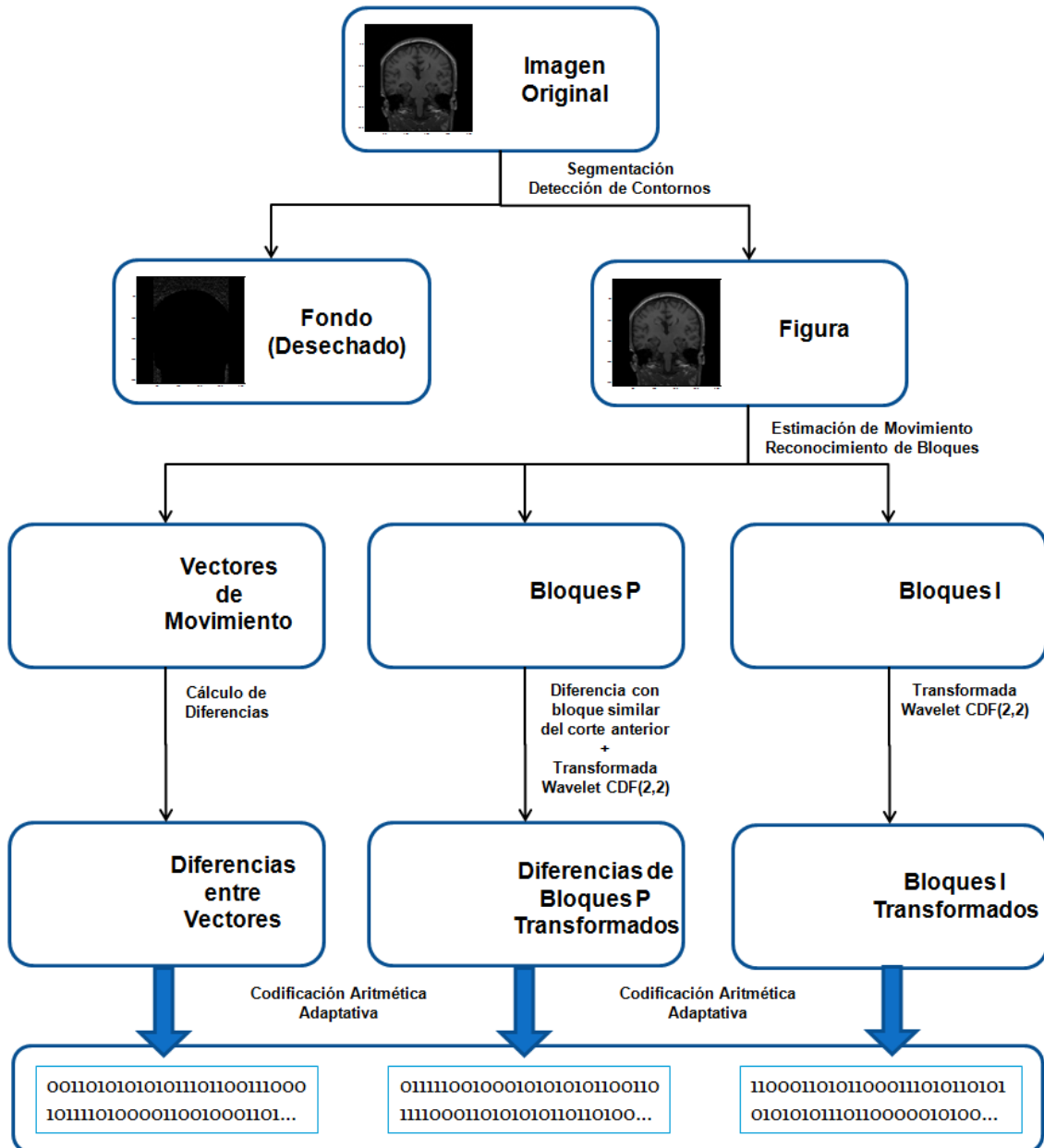


Fig. 7 Esquema del método de compresión propuesto

## 2. Definiciones Previas

Podemos definir un estudio de MRI de dimensiones  $N_X \times N_Y \times N_Z$  en los ejes de coordenadas  $x, y, z$  como una matriz tridimensional que denotamos como:

$$E_{i,j,k} \text{ con } \begin{cases} 1 \leq i \leq N_X \\ 1 \leq j \leq N_Y \\ 1 \leq k \leq N_Z \end{cases}$$

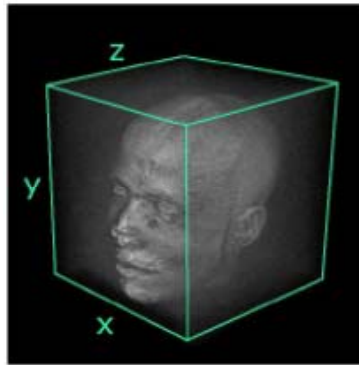


Fig. 8 Representación 3D de un estudio MRI

Cada posición de la matriz tridimensional mencionada es denominada voxel. En particular, podemos definir cada corte del estudio en cada uno de sus ejes de coordenadas de la siguiente manera:

$$E_{i,\blacksquare,\blacksquare} = E_{i,j,k} \text{ con } 1 \leq j \leq N_Y \wedge 1 \leq k \leq N_Z$$

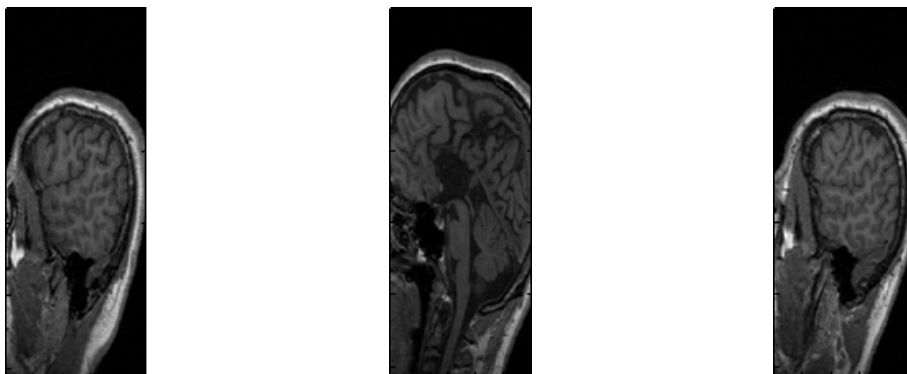


Fig. 9 Diferentes cortes de un mismo estudio tomados a lo largo del eje x

$$E_{\blacksquare, j, \blacksquare} = E_{i, j, k} \text{ con } 1 \leq i \leq N_X \wedge 1 \leq k \leq N_Z$$

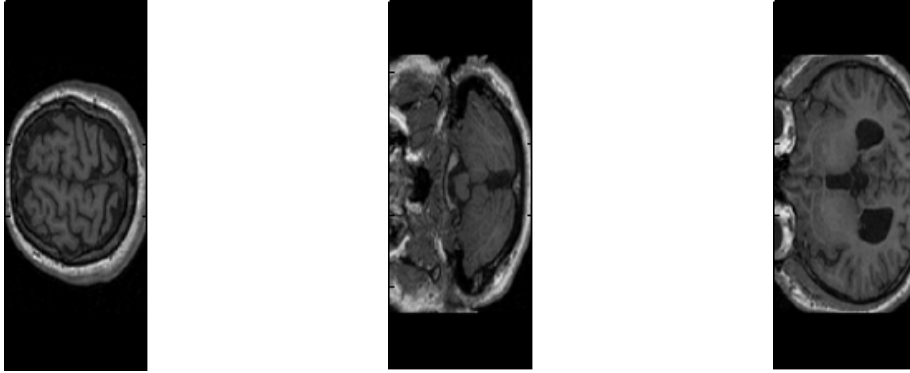


Fig. 10 Diferentes cortes de un mismo estudio tomados a lo largo del eje y

$$E_{\blacksquare, \blacksquare, k} = E_{i, j, k} \text{ con } 1 \leq i \leq N_X \wedge 1 \leq j \leq N_Y$$

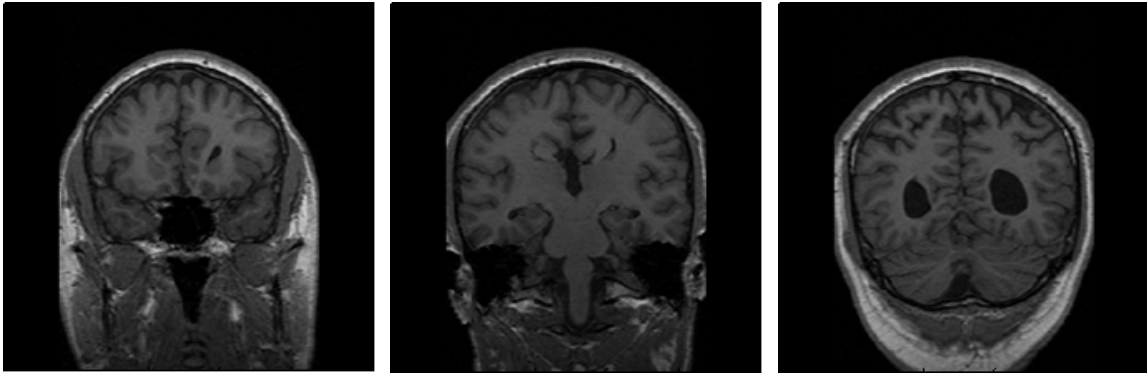


Fig. 11 Diferentes cortes de un mismo estudio tomados a lo largo del eje z

## 3. Segmentación por Detección de Contornos

### 3.1. Selección del Contorno Inicial

Para la propuesta de aplicación de las técnicas expuestas anteriormente a la segmentación (y posterior compresión) de imágenes de resonancia magnética, se ha utilizado una variante del método de conjuntos de nivel propuesta por Chumming, Chenyang, Changfeng y Fox [21]. Esta variante no necesita la mencionada “re inicialización” periódica de la función  $\varphi$  durante el proceso de evolución de la misma y, adicionalmente, no requiere que la función  $\varphi$  sea inicializada como una función de distancia con signo. Esto último permite iniciar el proceso utilizando un contorno arbitrario.

Para aplicar el método se toma, entonces, un corte bidimensional de una imagen de resonancia magnética, y se genera una máscara conteniendo el contorno inicial arbitrario a utilizar para la misma. En esta máscara  $M$  se determina el contorno inicial asignando los siguientes valores a los pixeles:

$$M(x, y) = \begin{cases} -4 & \text{si } \in \text{a la figura} \\ 0 & \text{si } \in \text{al contorno} \\ 4 & \text{si } \in \text{al fondo} \end{cases}$$

Por simplicidad se decidió utilizar como contorno inicial un recuadro que contenga la imagen (corte bidimensional) completa a procesar. Para ello se extiende la imagen original agregándole un recuadro exterior negro de 32 pixeles de ancho.

De esta forma, si la imagen original poseía 256x256 pixeles, al extenderla toma un nuevo tamaño de 320x320 pixeles (64 por cada lado). Ya con la imagen extendida se genera la máscara utilizando como contorno inicial un recuadro que contiene toda la imagen original más 16 pixeles por lado, contando como figura el interior del recuadro y como fondo al exterior del mismo.

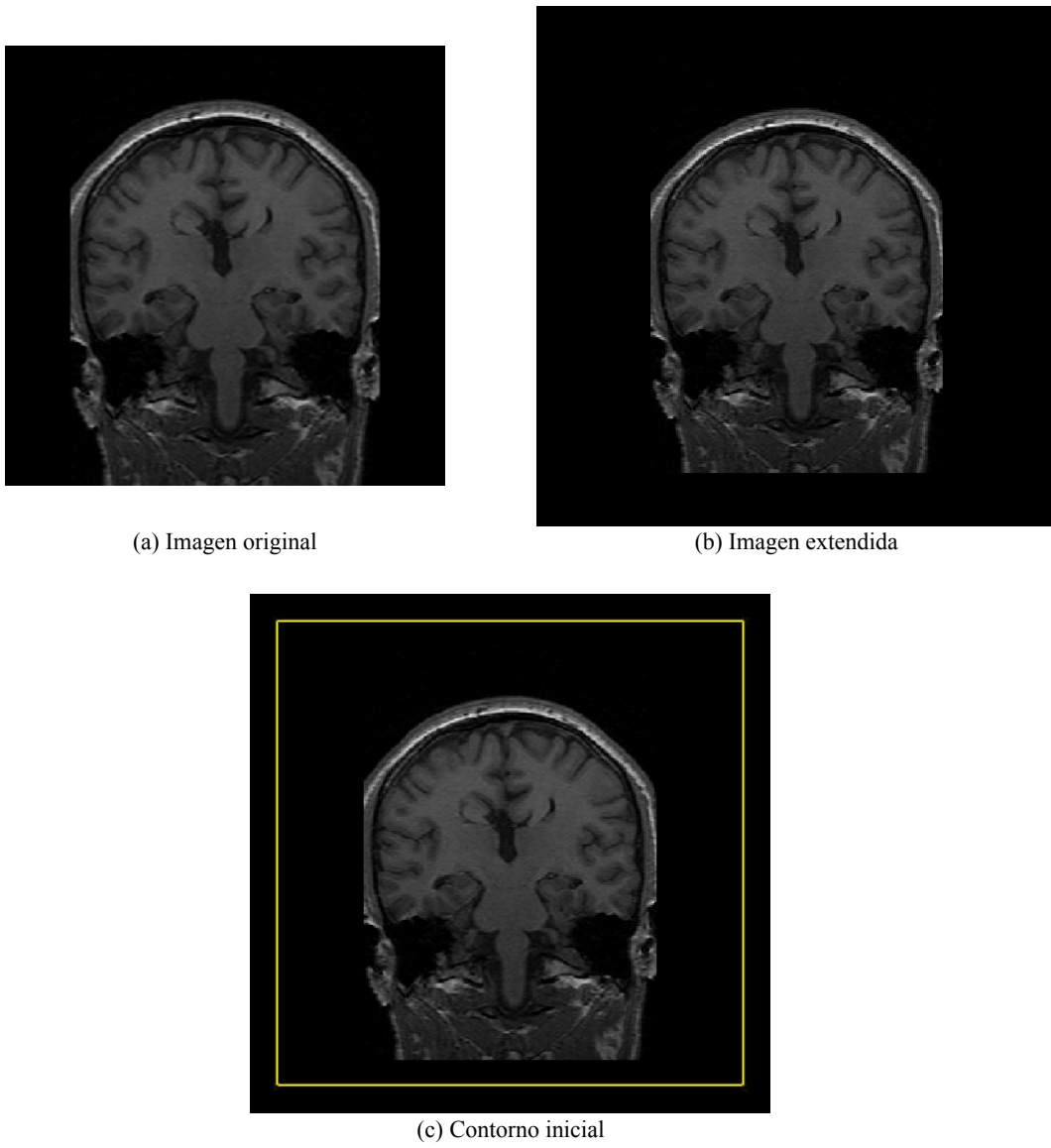


Fig. 12 Ejemplo de la construcción del contorno inicial de para un corte de una imagen de resonancia magnética

### 3.2. Selección del Contorno Inicial

En la segmentación de la imagen, los contornos activos son curvas dinámicas que se mueven acercándose hacia los límites de la figura. Para alcanzar este objetivo, el método de conjuntos de nivel utiliza una energía externa que mueve la curva de nivel cero hacia los límites de los objetos o figuras existentes en la imagen. Para ello se utiliza una *función indicadora de bordes*  $g$  definida como:

$$g = \frac{1}{1 + |\nabla G_{\sigma} * I|^2}$$

donde  $I$  es la imagen y  $\nabla G_{\sigma}$  es el núcleo Gaussiano con desviación estándar  $\sigma$ .

Con el objetivo de adaptar la función  $g$  a las características de la imagen que se está procesando se ha calculado  $\sigma$  en base a una estimación de la distribución del ruido presente en la imagen. Para ello, antes de comenzar la evolución de la función de conjuntos de nivel, se localiza el primer pixel de la imagen cuyo valor es diferente de cero y a partir del mismo se toma un recuadro pequeño (cuyo lado equivale a 1/20 del lado de la imagen) al cual se le calcula la desviación estándar, que será el valor asignado a  $\sigma$ .

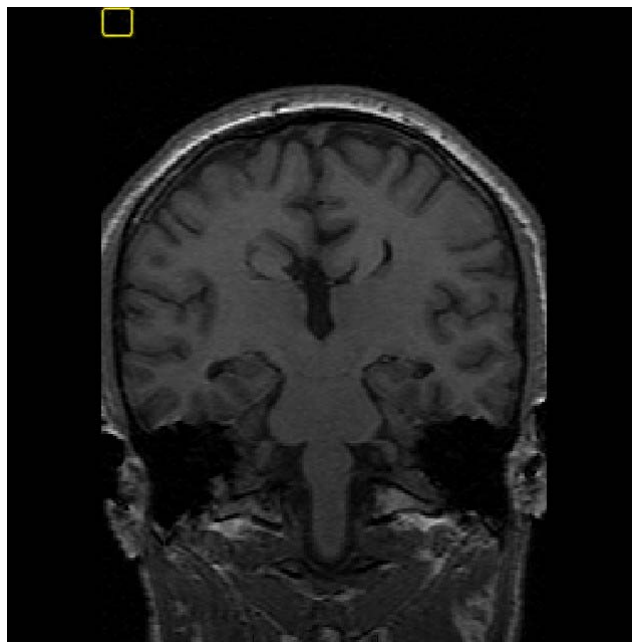


Fig. 13 Ejemplo del recuadro utilizado para estimar el nivel de ruido de la imagen. Se establece el valor de  $\sigma$  como la desviación estándar calculada sobre este recuadro.

Una vez establecidos el contorno inicial y el valor  $\sigma$  para la función indicadora de bordes, se procede a calcular la evolución de la función de conjuntos de nivel de manera iterativa. En cada iteración el método va modificando el contorno inicial y lo aproxima al contorno de la figura presente en la imagen.

La cantidad de iteraciones necesarias para obtener buenos resultados depende en buena parte del tamaño de la imagen y de los niveles de ruido presentes en la misma. Sin embargo, al seleccionar el  $\sigma$  de la función indicadora de bordes utilizando la estimación del nivel de ruido, se han conseguido buenos resultados determinando la

cantidad de iteraciones sólo en función del tamaño de la imagen. En particular, se ha optado por establecer el número de iteraciones igual a la cantidad de píxeles por lado de la imagen que se procesa.

Al finalizar las iteraciones, se obtiene en el contorno final detectado por el método para la imagen. A partir de aquí, se conforma la máscara correspondiente al corte procesado considerando como figura todos los píxeles con valores menores o iguales a cero, y como fondo los restantes.

Por último se elimina de la máscara el marco negro con el cual se había extendido a la imagen original a fines de mantener una coherencia entre el tamaño de la imagen y su máscara.

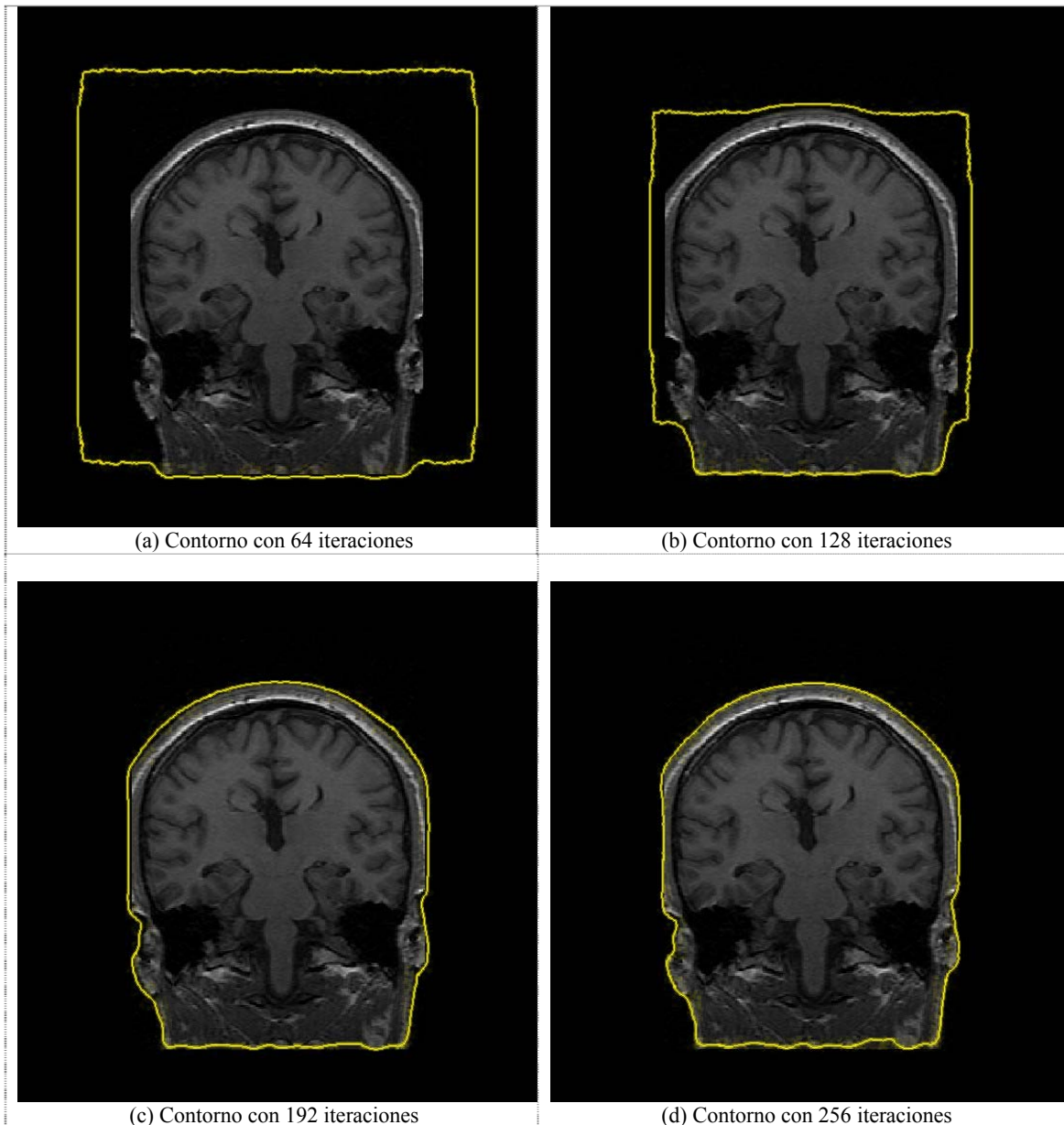


Fig. 14 Evolución del contorno inicial a medida que pasan las iteraciones de la función de conjuntos de nivel.



### 3.2. Generación de la Máscara de Segmentación

El método presentado hasta aquí permite realizar la segmentación de cada corte de la imagen de resonancia magnética por separado. Aplicándolo entonces a todos los cortes se puede obtener un conjunto de máscaras que permitirá segmentar la imagen completa.

Debido a la necesidad de que todo el proceso de compresión de este tipo de imágenes se realice sin pérdida de información, es extremadamente importante que la segmentación sea lo más precisa posible y minimizar la posibilidad de que un punto de la imagen perteneciente a la figura sea detectado como fondo durante este proceso.

Para reforzar en este sentido el método propuesto se decidió realizar todo el proceso recorriendo la imagen en sus tres ejes dimensionales y procesando individualmente de esta forma todos los cortes posibles de la imagen. Como cada voxel (pixel tridimensional) de la imagen forma parte de tres cortes diferentes (uno por cada eje dimensional), con esta metodología, el mismo es procesado tres veces.

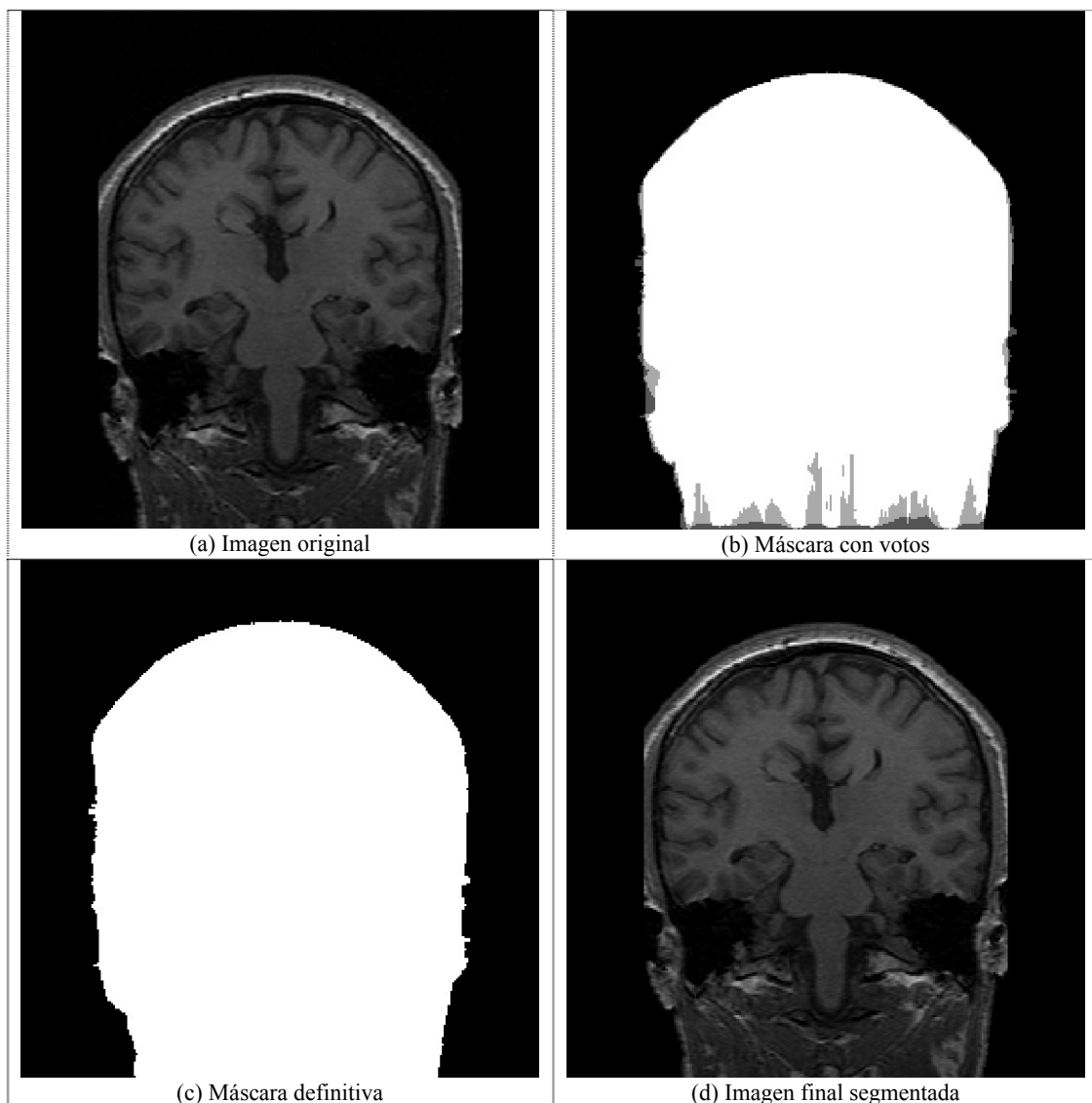


Fig. 15 Ejemplo del proceso de segmentación completo desde la imagen original hasta la imagen segmentada.

Aprovechando este procesamiento triple de la imagen se ha implementado un

mecanismo de votación, propuesto por Corvetto [8], [9] mediante el cual se considera que si un voxel ha sido detectado como parte de la figura en al menos de las tres oportunidades en que fue procesado, el mismo es considerado será considerado como parte integrante de la figura en la máscara final. De esta manera, sólo se consideran como parte del fondo, y por lo tanto podrán ser descartados, aquellos voxels que hayan sido detectados como fondo en las tres oportunidades en que han sido procesados.

### 3.2. Robustización del Método

El método presentado fue comparado con el método basado en histogramas utilizado por Corvetto [8], [9] sobre las mismas secuencias. En dicho método se realiza la segmentación de fondo y figura mediante la aplicación de un umbral a la imagen. El mismo es calculado mediante la estimación de la distribución del ruido presente en la imagen utilizando para ello una función de Rayleigh, minimizando la probabilidad de un error de clasificación.

Una vez calculado el umbral se obtiene una máscara inicial considerando como fondo los valores inferiores al mismo y como figura los valores superiores. Sobre ésta máscara se aplican sobre la misma operaciones morfológicas de rellenado y erosión con el fin de recuperar aquellas zonas relevantes de la imagen que pudieran haber quedado por debajo del umbral.

Al comparar el método propuesto con el recientemente descrito se advierte que, dependiendo de la imagen sobre la cual son aplicados, se obtienen mejores resultados con uno u otro método. En particular, es muy común que en zonas de una imagen el método de umbrales supere al de conjuntos de nivel, pero en otras zonas de la misma imagen se den resultados opuestos. Por estas razones, se considera que ambas técnicas pueden complementarse muy bien, obteniendo así un mecanismo de segmentación combinado más robusto que cualquiera de los dos métodos por separado.

Para efectuar la combinación mencionada de ambos métodos basta con ampliar el mecanismo de votación propuesto anteriormente. Obteniendo entonces un total de 6 “votos” (3 por la técnica de conjuntos de nivel y 3 por el método de Corvetto) por cada voxel presente en la imagen. Al igual que antes, para minimizar la posibilidad de pérdida de información, sólo serán considerados como fondo aquellos voxels que no hayan sido detectados como figura en ninguna de las 6 veces en que se los ha procesado.

Se ha observado que el método final combinado reduce la posibilidad de pérdida de información durante la segmentación pero que, en algunos casos podría llevar a resultados no deseados si alguno de los dos métodos detectara como figura demasiados puntos pertenecientes al fondo. Un ejemplo leve de esto puede observarse en Fig. 16(a), en donde han quedado sectores del fondo dentro del contorno obtenido por el método de umbrales. Una posible mejora para estos casos podría alcanzarse mediante modificaciones al sistema de votación, por ejemplo exigiendo al menos dos votos para considerar un voxel como parte de la figura. Aunque obviamente este cambio produciría también un aumento en la probabilidad de pérdida de información.

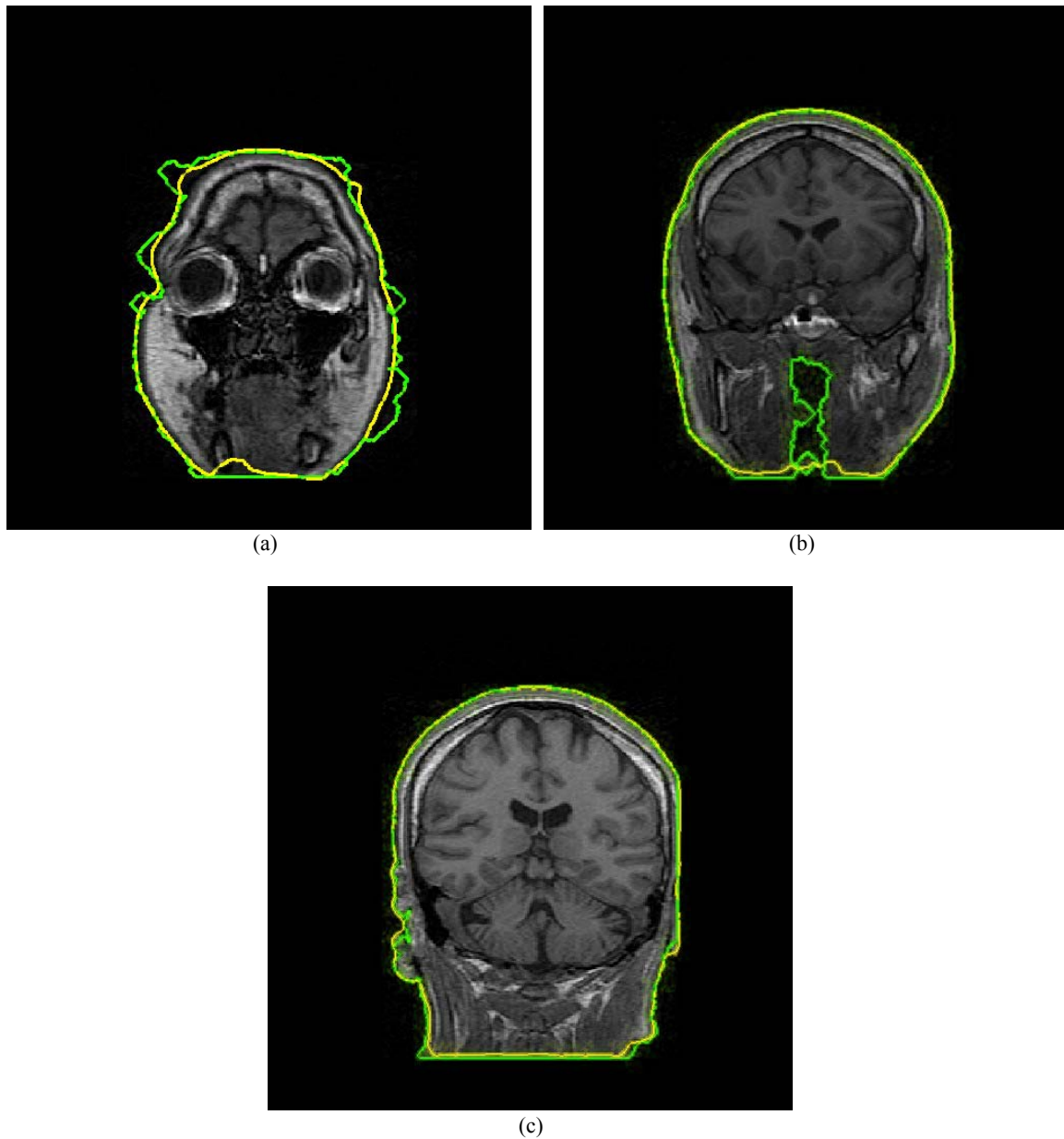


Fig. 16 Comparación de la segmentación por conjuntos de nivel (línea amarilla) con la segmentación por umbral (línea verde) en distintos cortes de una misma resonancia magnética.

## 4. Estimación de Movimiento

Una vez efectuado el proceso de segmentación, y habiendo separado la figura del fondo, se debe proceder a codificar la figura resultante. Dada la naturaleza tridimensional de los estudios MRI y la similitud existente entre un corte y el siguiente, podemos imaginar cada uno de los cortes como un cuadro dentro de una secuencia de video. A partir de esto, surge la idea de aplicar la técnica de estimación de movimientos a través del reconocimiento de bloques coincidentes (o similares) entre un corte y del estudio y su corte inmediato anterior.

### 4.1 Búsqueda Exhaustiva de Bloques Coincidentes

Se elige entonces, uno de los tres ejes dimensionales del estudio a lo largo del cual se realizará la comparación entre cortes sucesivos. La única restricción existente sobre el eje dimensional a elegir está dada por la necesidad de que los cortes deberán ser cuadrados con una cantidad de voxels por lado que sea potencia de 2. Si esto no se cumple, el algoritmo puede ser igualmente aplicado, pero se deberán tener ciertas precauciones en los bordes, buscando por ejemplo bloques más pequeños en la frontera, y rellenando con ceros para aplicar la wavelet.

Se divide cada corte en macro bloques de  $K$  voxels por lado, con  $K$  siendo una potencia de 2 menor a la cantidad de voxels por lado del corte completo. Se elige un parámetro de búsqueda  $p$  y un umbral  $u$  de tolerancia para la coincidencia entre bloques.

Por cada bloque perteneciente al corte que se está procesando, se realiza una búsqueda exhaustiva de bloques coincidentes o similares en el corte inmediato anterior utilizando el parámetro de búsqueda  $p$  para limitar el desplazamiento máximo permitido. De aquí se obtiene, por cada bloque  $B_p$  que se está procesando, el bloque  $B_A$  que más se le asemeja en el corte anterior dentro del desplazamiento permitido. Al mismo tiempo, se obtiene el vector de desplazamiento  $v = (v_x, v_y)$ , que indica la cantidad de voxels en los ejes  $x$  e  $y$  que se ha desplazar el bloque  $B_A$  para llegar a la posición ocupada por  $B_p$ , y el la medida del error de la diferencia entre bloques  $c(B_A, B_p)$ .

### 4.2 Codificación de Bloques

Una vez obtenidos los datos mencionados anteriormente, se obtiene la codificación  $f(B_p)$  del bloque  $B_p$  según el esquema:

$$f(B_P) = \begin{cases} B_P - B_A & \text{si } c(B_A, B_P) < u \\ B_P & \text{en otro caso} \end{cases}$$

Dependiendo de la forma en que se haya codificado, cada bloque  $B_P$  podrá ser clasificado entonces ahora como Bloque P (o bloque predicho), si fue codificado como la diferencia con  $B_P - B_A$ , o como Bloque I (o bloque intra), en el otro caso.

Para poder realizar la decodificación posteriormente, será necesario guardar también el vector de desplazamiento  $v$ . En el caso de los bloques I, se descartará el valor del vector obtenido previamente y establecerá  $v = (p + 1, p + 1)$ . Recordemos que el máximo desplazamiento permitido es  $p$  (el parámetro de búsqueda), por lo tanto un vector de desplazamiento con el valor especificado indicará que se trata de un bloque intra, y por lo tanto ha sido codificado en forma directa.

Dado que el primer corte del estudio no puede ser comparado con ningún corte anterior, todos los bloques del mismo son codificados como bloques I.

### 4.3 Arreglos Resultantes

Como resultado del procedimiento descrito, se obtienen finalmente tres arreglos tridimensionales, a saber:

- Un arreglo ABP que contiene todos los Bloques P obtenidos durante la codificación, cuyas dimensiones son  $K \times K \times N_P$ , donde  $N_P$  es la cantidad de Bloques P.
- Un arreglo ABI que contiene todos los Bloques I obtenidos durante la codificación, cuyas dimensiones son  $K \times K \times N_I$ , donde  $N_I$  es la cantidad de Bloques I.
- Un arreglo AVD que contiene todos los vectores de desplazamientos  $v$  obtenidos durante la codificación, cuyas dimensiones son  $2 \times 1 \times (N_P + N_I)$ .

Dado que todos los bloques y vectores se van incorporando al arreglo correspondiente en el orden en que son procesados, durante el proceso de decodificación o descompresión, el arreglo AVD nos servirá de índice para reconstruir, bloque a bloque, cada corte del estudio original, indicando si se en cada caso si se debe tomar un del arreglo ABP o uno del arreglo ABI.

## 5. Procesamiento de los Bloques I y P

A continuación se describen los pasos seguidos para realizar la codificación final de los arreglos de bloques ABP y ABI obtenidos hasta el momento.

### 5.1 Aplicación de la Transformada CDF(2,2)

A los arreglos ABP y ABI obtenidos en la sección 4.3 se los recorre bloque a bloque y por cada uno se aplica la versión bidimensional de la transformada wavelet de enteros en enteros CDF (2,2) en tantos niveles como las dimensiones del bloque lo permita. De esta forma, se obtienen dos nuevos arreglos ABPT y ABIT, los cuales contienen todos los bloques transformados, al igual que los anteriores, pero poseen una entropía menor.

Dado que el lado (en voxels) de los bloques es una potencia de 2 (al igual que el lado de cada uno de los cortes procesados), no es necesario efectuar ninguna operación de rellenado previa a la aplicación de la transformada en cuestión.

### 5.2 Codificación Aritmética

Para finalizar, los dos arreglos ABPT y ABIT resultantes de la aplicación de la transformada wavelet son codificados utilizando un codificador aritmético adaptativo. Ambos arreglos son almacenados, con distintas codificaciones, en un mismo archivo, en el cual se guardará posteriormente el arreglo ADV que contiene los vectores de desplazamiento.

## 6. Procesamiento de los Vectores de Desplazamiento

A continuación se describen los pasos seguidos para realizar la codificación final del arreglo ADV de vectores de desplazamiento obtenido en la sección 4.3

### 6.1 Cálculo de las Diferencias

Una vez obtenido el arreglo ADV como se ha descrito anteriormente, se procede a generar un nuevo arreglo ADVD, que contenga las diferencias entre dos vectores de desplazamientos consecutivos. Si continuamos llamando  $N_C$  a la cantidad de cortes procesados del estudio, podemos definir formalmente el mecanismo de cálculo de estas diferencias como:

$$ADVD_{\mathbf{n},\mathbf{n},1} = ADV_{\mathbf{n},\mathbf{n},1}$$

$$ADVD_{\mathbf{n},\mathbf{n},k} = ADV_{\mathbf{n},\mathbf{n},k} - ADV_{\mathbf{n},\mathbf{n},k-1} \text{ para } 2 \leq k \leq N_C$$

Este nuevo arreglo obtenido tiene una entropía menor a la del arreglo original y, por lo tanto, favorecerá en el paso siguiente, el aumento de la tasa de compresión.

### 6.2 Codificación Aritmética

El arreglo ADVD finalmente es procesado con un codificador aritmético adaptativo y almacenado en el mismo archivo en que se han guardado anteriormente los arreglos ABPT y ABIT.

## 7. Datos auxiliares

Para poder descomprimir los datos que han sido codificados en el archivo antes mencionado, será necesario guardar también, dentro de dicho archivo, algunos pocos datos auxiliares, a saber:

- Los diccionarios de símbolos para cada una de las tres codificaciones aritméticas utilizadas.
- El tamaño original de un corte del estudio
- El tamaño de los bloques utilizados.
- El valor del parámetro de búqueda  $p$ .

El primer ítem es necesarios, obviamente, para poder realizar la decodificación de la codificación aritmética aplicada y recuperar los arreglos guardados, mientras que los siguientes dos ítems servirán para calcular la cantidad de bloques existentes en cada cuadro, de forma de poder reconstruir los cuadros a medida que se recorren los arreglos recuperados. El último ítem es necesario para identificar dentro vector de desplazamiento los valores que indican un bloque I.



## 8. Descompresión

Como puede suponerse, el proceso de compresión ( o decodificación ), implica invertir cada uno de los procesos realizados durante la etapa de compresión, invirtiendo también el orden de en que son aplicados. Es decir, el inverso del último proceso aplicado durante la compresión, será el primero aplicado en la descompresión.

Los pasos a seguir son los siguientes:

- 1) A partir del archivo comprimido, se reconstruyen los arreglos ABPT, ABIT y ADVD (denominados  $ABPT_R$ ,  $ABIT_R$  y  $ADVD_R$  respectivamente).
- 2) Se aplica la anti-transformada wavelet CDF(2,2) a cada uno de los bloques existentes en los arreglos  $ABPT_R$  y  $ABIT_R$  para obtener los arreglos  $ABP_R$  y  $ABI_R$ .
- 3) Utilizando un proceso inverso al cálculo de diferencias, se obtiene el arreglo  $ADV_R$  a partir de  $ADVD_R$ .
- 4) Se reconstruye el primer corte del estudio a partir de los primeros bloques I existentes en el arreglo  $ABI_R$ .
- 5) Utilizando como guía los valores contenidos en el arreglo  $ADV_R$ , se reconstruye la codificación de cada uno de los cortes siguientes de la imagen, tomando bloques de los arreglos  $ABI_R$  o  $ABP_R$ , según corresponda, y decodificando los bloques P en base al corte inmediato anterior que ya fue decodificado.

# IV. Resultados

## 1. Implementación

El algoritmo fue implementado utilizando la aplicación MATLAB R2008a. Para ello se utilizaron varias rutinas implementadas previamente por otros autores, como es el caso de las rutinas de conjuntos de nivel, wavelets, búsqueda exhaustiva de bloques coincidentes y codificador aritmético.

## 2. Datos de prueba

El algoritmo fue testeado utilizando diversos estudios MRI obtenidos previamente. Estos estudios se encontraban originalmente almacenados en archivos binarios con formato DICOM utilizando 16 bits por voxel.

Las estimaciones de tasa de compresión se ha realizado utilizando como comparación el tamaño de los archivos DICOM originales, que es el formato en el que suelen presentarse este tipo de estudios.

A continuación se presenta una tabla con el detalle de los estudios utilizados y sus características:

<b>Estudio</b>	<b>Dimensiones (en voxels)</b>	<b>Tamaño en bits</b>	<b>Entropía</b>
sk1	256 x 256 x 99	103.809.024	6,0197
sk2	256 x 256 x 109	114.294.784	6,0941
sk3	256 x 256 x 94	98.566.144	4,8087
sk4	256 x 256 x 119	124.780.544	5,6902
sk5	256 x 256 x 119	124.780.544	5,7730
sk6	256 x 256 x 119	124.780.544	6,0537
sk7	256 x 256 x 119	124.780.544	5,8845
sk8	256 x 256 x 109	114.294.784	4,4882
sk9	256 x 256 x 129	135.266.304	5,8065
fl114	512 x 512 x 18	75.497.472	7,9431
fl115	512 x 512 x 18	75.497.472	8,1040
fl414	512 x 512 x 17	71.303.168	6,4616
fl512	512 x 512 x 20	83.886.080	7,3433
fl513	512 x 512 x 20	83.886.080	6,9066
fl514	512 x 512 x 20	83.886.080	7,5059

Tabla 1 Detalle de estudios de prueba

### **3. Tiempo de ejecución**

Es importante mencionar que durante el desarrollo del presente trabajo se ha priorizado fundamentalmente la robustez del método de segmentación, la cual implica un menor riesgo de pérdida de información relevante, por sobre los tiempos de ejecución. Las pruebas fueron ejecutadas en MATLAB R2008a utilizando diferentes equipos y sistemas operativos. La etapa más costosa en términos de tiempo de ejecución ha sido, sin dudas, la creación de la máscara de segmentación a través de la detección de contornos en cada uno de los cortes del estudio.

En menor medida, también la búsqueda exhaustiva utilizada para el reconocimiento de bloques durante la estimación de movimiento puede convertirse en un proceso con alto tiempo de ejecución dependiendo del tamaño de bloque y parámetros de búsqueda que se utilicen.

Afortunadamente, los dos procesos mencionados sólo son necesarios durante la compresión, por lo que, una vez comprimido, la descompresión del estudio puede realizarse en un tiempo mucho menor.

## 4. Compresión

A continuación se muestran los resultados de compresión obtenidos para los estudios utilizados en las pruebas del método presentado.

Se han realizado pruebas con diferentes valores para los parámetros del algoritmo, pero los mejores resultados han sido arrojados con los siguientes valores:

- $K = 16$  (voxels por lado para cada macro bloque)
- $p = 16$  (parámetro de búsqueda que limita el desplazamiento)
- $u = 3$  (umbral de tolerancia para codificar como bloque P)
- Se ha utilizado el eje z para recorrer los cortes del estudio

Estudio	Tamaño original (en bits)	Voxels c/Info	Entropía original	Resultados de la compresión		
				Tamaño final (en bits)	Bits p/voxel (bpv)	Tasa de Compresión
sk1	103.809.024	50,46%	6,0197	24.698.992	3,807	4,20:1
sk2	114.294.784	47,06%	6,0941	27.097.600	3,793	4,21:1
sk3	98.566.144	49,28%	4,8087	16.937.952	2,749	5,81:1
sk4	124.780.544	39,55%	5,6902	24.529.248	3,145	5,08:1
sk5	124.780.544	40,34%	5,7730	25.194.576	3,231	4,95:1
sk6	124.780.544	44,69%	6,0537	27.964.816	3,586	4,46:1
sk7	124.780.544	42,53%	5,8845	26.344.920	3,378	4,73:1
sk8	114.294.784	42,13%	4,4882	17.737.648	2,483	6,44:1
sk9	135.266.304	41,18%	5,8065	28.253.664	3,342	4,78:1
fl114	75.497.472	48,54%	7,9431	20.492.936	4,343	3,68:1
fl115	75.497.472	44,78%	8,1040	16.346.528	3,464	4,61:1
fl414	71.303.168	33,28%	6,4616	11.077.664	2,486	6,43:1
fl512	83.886.080	36,40%	7,3433	14.781.152	2,819	5,67:1
fl513	83.886.080	43,23%	6,9066	17.526.032	3,343	4,78:1
fl514	83.886.080	24,14%	7,5059	11.224.064	2,141	7,47:1
<b>Promedios</b>	<b>102.620.638</b>	<b>41,84%</b>	<b>6,3255</b>	<b>20.680.519</b>	<b>3,207</b>	<b>5,15:1</b>

Tabla 2. Resultados de la compresión de los estudios de prueba

Cabe destacar que, como era de esperarse, la cantidad de voxels detectados como parte de la figura durante el proceso de segmentación influye en gran medida en las tasas de compresión que pueden obtenerse.

Con el objetivo visualizar mejor el funcionamiento del método se han analizado los archivos comprimidos resultantes y se ha determinado con mayor precisión como ha quedado compuestos los mismos. A continuación se presentarán los porcentajes de Bloques I y Bloques P que conforman cada estudio, así también como una distribución (también mediante porcentajes) del tamaño total comprimido entre los tres arreglos que lo componen (ABPT, ABIT y AVDT).

Estudio	Porcentaje Bloques I	Porcentaje Bloques P	Distribución de bits entre los arreglos comprimidos		
			Bloques I	Bloques P	Vector de Desplazamiento
sk1	54,72 %	45,28 %	98,50 %	1,39 %	0,12 %
sk2	52,42 %	47,58 %	98,31 %	1,54 %	0,14 %
sk3	49,42 %	50,58 %	89,45 %	10,31 %	0,23 %
sk4	44,00 %	56,00 %	97,87 %	1,94 %	0,18 %
sk5	44,65 %	55,35 %	97,61 %	2,22 %	0,18 %
sk6	49,16 %	50,84 %	98,24 %	1,61 %	0,15 %
sk7	47,26 %	52,74 %	97,96 %	1,88 %	0,15 %
sk8	42,76 %	57,24 %	89,16 %	10,57 %	0,27 %
sk9	45,23 %	54,77 %	97,75 %	2,08 %	0,16 %
fl114	54,18 %	45,82 %	99,04 %	0,80 %	0,17 %
fl115	50,59 %	49,41 %	98,73 %	1,05 %	0,22 %
fl414	39,03 %	60,97 %	97,37 %	2,30 %	0,33 %
fl512	42,64 %	57,36 %	97,68 %	2,01 %	0,32 %
fl513	50,50 %	49,50 %	98,91 %	0,87 %	0,22 %
fl514	32,40 %	67,60 %	95,86 %	3,56 %	0,57 %
<b>Promedios</b>	<b>46,60%</b>	<b>53,40%</b>	<b>96,83%</b>	<b>2,94%</b>	<b>0,23%</b>

Tabla 3. Composición de los archivos comprimidos resultantes de las pruebas de compresión

Como puede observarse en la tabla presentada, en la mayoría de los estudios se ha obtenido una distribución similar de bloques P y bloques I (aproximadamente un 50% de cada tipo).

Por otra parte, también es interesante marcar que, en todos los casos, entre el 90 y el 99% del los bits del archivo comprimido resultante son utilizados para la codificación de Bloques I, dejando tan sólo entre el 0,8 y el 10% los bits utilizados en la codificación de Bloques P. El vector de desplazamientos por su parte, requiere la utilización de un número menor al 1% de los bits totales del archivo.

## 5. Comparación con otros métodos

Para poder evaluar la efectividad del método, se ha comparado el mismo con otros métodos de compresión conocidos, aplicándolos a los mismos estudios utilizados para las pruebas.

Los métodos seleccionados para realizar la comparación son:

- PNG [22]
- JPEG2000 [23]
- JPEG-LS (LOCO-I) [24]
- SPIHT [25]
- ZIP [26]

En los primeros cuatro casos, se ha realizado la compresión de cada cuadro de la imagen y luego se calculó la sumatoria de los tamaños de los archivos obtenidos. En el caso de la compresión ZIP simplemente se tomaron los archivos originales y se los comprimió con el compresor en cuestión.

Es importante aclarar que los métodos mencionados fueron aplicados sobre los estudios originales, sin la aplicación previa de ningún proceso de segmentación.

Estudio	Voxels c/Info	PNG	JPEG 2000	JPEG LS	SPIHT	ZIP	Método Prop.
sk1	50,46%	2,75:1	3,52:1	3,73:1	3,85:1	2,46:1	<b>4,20:1</b>
sk2	47,06%	2,63:1	3,26:1	3,63:1	3,74:1	2,44:1	<b>4,21:1</b>
sk3	49,28%	3,87:1	5,23:1	5,20:1	5,64:1	3,29:1	<b>5,81:1</b>
sk4	39,55%	2,77:1	3,42:1	3,79:1	3,91:1	2,61:1	<b>5,08:1</b>
sk5	40,34%	2,72:1	3,37:1	3,72:1	3,85:1	2,57:1	<b>4,95:1</b>
sk6	44,69%	2,65:1	3,23:1	3,65:1	3,76:1	2,47:1	<b>4,46:1</b>
sk7	42,53%	2,73:1	3,50:1	3,77:1	3,89:1	2,54:1	<b>4,73:1</b>
sk8	42,13%	3,93:1	5,33:1	5,35:1	5,85:1	3,51:1	<b>6,44:1</b>
sk9	41,18%	2,70:1	3,32:1	3,71:1	3,82:1	2,55:1	<b>4,78:1</b>
fl114	48,54%	2,33:1	2,54:1	3,33:1	3,47:1	1,99:1	<b>3,68:1</b>
fl115	44,78%	2,11:1	2,52:1	2,96:1	3,14:1	1,79:1	<b>4,61:1</b>
fl414	33,28%	2,90:1	3,61:1	4,16:1	4,46:1	2,50:1	<b>6,43:1</b>
fl512	36,40%	2,22:1	2,77:1	3,08:1	3,36:1	1,89:1	<b>5,67:1</b>
fl513	43,23%	2,61:1	3,09:1	3,66:1	3,80:1	2,28:1	<b>4,78:1</b>
fl514	24,14%	1,94:1	2,39:1	2,53:1	2,63:1	1,79:1	<b>7,47:1</b>
<b>Promedios</b>	<b>41,84%</b>	<b>2,72:1</b>	<b>3,41:1</b>	<b>3,75:1</b>	<b>3,94:1</b>	<b>2,45:1</b>	<b>5,15:1</b>

Tabla 4. Comparación de los resultados obtenidos con otros métodos

## 6. Discusión

### 6.1 Parámetros de compresión

Como se ha mencionado anteriormente, se han realizado diversas pruebas para evaluar los parámetros de compresión que producen los mejores resultados.

En particular, tanto para  $K$  como para  $p$  se han utilizados entre 2 y 16 inclusive (cabe recordar que  $K$  debe ser una potencia de 2). En cada caso se ha procurado utilizado siempre la restricción  $p \leq K$  para restringir la búsqueda exhaustiva a bloques que sean adyacentes al que se desea procesar y, también, para evitar que los tiempos de ejecución se eleven demasiado.

Respecto al umbral, se han utilizado múltiples valores de diferentes magnitudes, detectando que los mejores comportamientos se encuentran con valores pequeños ( $u < 5$ ).

Se presentan los resultados promedio obtenidos para diferentes combinaciones de parámetros, expresados en bpv (bits por voxel). En cada caso, se ha marcado el mejor resultado obtenido.

$K = 4$		$u$						
		1	2	3	4	5	10	15
$p$	2	3,4299	3,4352	3,4434	3,4514	3,5087	3,5959	3,6557
	4	3,429	3,4355	3,4446	3,4544	3,5191	3,6187	3,6804

Tabla 5. Entropías promedio utilizando bloques de 4 voxels de lado

$K = 8$		$u$						
		1	2	3	4	5	10	15
$p$	2	3,2667	3,2586	3,26	3,2666	3,3287	3,3909	3,4521
	4	3,2618	3,2536	3,2548	3,2625	3,3248	3,3906	3,4577
	8	3,259	3,2496	3,2517	3,2601	3,3244	3,3916	3,464

Tabla 6. Entropías promedio utilizando bloques de 8 voxels de lado

$K = 16$		$u$						
		1	2	3	4	5	10	15
$p$	2	3,2451	3,2242	3,2181	3,2232	3,2824	3,3349	3,3902
	4	3,2402	3,2194	3,2135	3,2176	3,2816	3,331	3,3883
	8	3,236	3,215	3,2097	3,2134	3,2803	3,33	3,388
	16	3,2336	3,2125	3,2074	3,211	3,3788	3,329	3,3873

Tabla 7. Entropías promedio utilizando bloques de 16 voxels de lado

Dado que en todos los casos los mejores resultados se han dado con el mayor valor de  $p$  utilizado, se decidió realizar nuevas pruebas utilizando valores mayores, omitiendo la restricción original  $p \leq K$  y manteniendo  $u < 5$ . Sin embargo, los resultados obtenidos en estas nuevas pruebas fueron peores que los obtenidos anteriormente.

Se presentan a continuación los resultados de las nuevas pruebas mencionadas. A modo comparativo, se incluye también el mejor resultado obtenido durante las pruebas originales.

$K = 16$		$u$			
		1	2	3	4
$p$	16	3,2336	3,2125	<b>3,2074</b>	3,211
	18	3,2518	3,2467	3,254	3,264
	20	3,2515	3,2465	3,2537	3,2637
	22	3,2513	3,2403	3,2529	3,263

Tabla 8. Entropías promedio utilizando bloques de 16 voxels de lado y  $p \geq K$ .



## 6.2 Suavizado del corte previo a la estimación de movimiento

Durante el desarrollo del método presentado, se ha contemplado la posibilidad de que, dado que el estudio posee altos niveles de ruido, este ruido estuviera interfiriendo en el óptimo rendimiento del cálculo de la medida de error utilizada para determinar el bloque coincidente en el corte anterior.

Como tanto el corte que se está procesando como el anterior contienen ruido, ambos cortes contienen ruido, en caso de que ambos ruidos no se cancelen mutuamente, el cálculo podría considerar dos veces el ruido (en magnitud) al calcular la diferencia entre bloques.

Para evitar esto, se procedió a suavizar primero el corte anterior (sobre el que se está buscando el bloque coincidente) utilizando un filtro pasa bajos definido como:

$$f = \begin{bmatrix} 0 & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\ 0 & \frac{1}{8} & 0 \end{bmatrix}$$

y recién luego realizar la búsqueda exhaustiva del bloque coincidente al actual.

Al realizar pruebas con esta técnica, no se observaron diferencias de rendimiento. Por el contrario, en la mayoría de los casos, los resultados fueron levemente peores que los obtenidos sin realizar el suavizado. Es por esto que finalmente decidió descartarse este paso.

## 7. Conclusiones

Los altos niveles de ruido presentes en las imágenes de resonancia magnética dificultan la compresión eficiente y sin pérdida de información de las mismas. Los métodos de segmentación para separar el fondo ruidoso de la figura relevante han demostrado ser muy útiles en los mencionados procesos de compresión.

Se ha presentado un método que realiza la segmentación a través de la detección de contornos en la imagen, aplicando para ello técnicas de conjuntos de nivel. La comparación de los resultados obtenidos con los arrojados por otros métodos como el de segmentación por umbrales nos ha permitido otorgarle mayor robustez al procedimiento mediante la combinación de ambos mecanismos.

Como debilidad del método presentado debe mencionarse el tiempo que demora la creación de una máscara tridimensional completa (aproximadamente 45 minutos para una imagen con 100 cortes de 256x256 píxeles cada uno), debido a la cantidad de iteraciones necesarias para la detección de contornos en cada corte de la imagen mediante el mecanismo de conjuntos de nivel utilizado. En contraposición al método de umbrales, que demora apenas algo más de 1 minuto en calcular la máscara para la misma imagen. Afortunadamente esta demora sólo afecta al proceso de compresión, permitiendo que la descompresión de la imagen sea mucho más ágil.

Se ha presentado también un método para codificar y comprimir la imagen segmentada obtenida que aplica reconocidas técnicas de compresión de video que permiten explotar la similitud existente entre cortes consecutivos de los estudios de resonancia magnética. La comparación realizada con otros métodos de compresión ha mostrado resultados prometedores.

Debido a la utilización de estas técnicas de compresión de video mencionadas, el método presenta un mejor comportamiento en imágenes que poseen un alto grado de similitud entre cortes. Lo cual es de esperar que ocurra en imágenes con muchos cortes, ya que los mismos se encuentran menos espaciados.

Si bien en las pruebas realizadas se han obtenido resultados muy altos (en cuanto a tasa de compresión) en imágenes con pocos cortes (fl414 y fl514), esto está relacionado con la baja cantidad de voxels con información existentes en dichos estudios, provocando que ya durante el proceso de segmentación se reduzca notablemente la cantidad de bits utilizados.

Comparando las entropías originales de los estudios con los resultados obtenidos se puede observar que, previsiblemente, aquellas imágenes con una menor entropía original suelen presentar también las mejores tasas de compresión (siempre comparando entre estudios de características similares en cuanto a tamaño y cantidad de cuadros).

# Referencias

- [1] P. Roos, M. Viergever, M. Van Dijke y J. Peters, *Reversible intraframe compression of medical images*, IEEE Transactions on Medical Imaging, Pattern Recognition, 7(4):328-336, 1988.
- [2] S. Gangl y B. Žalik, *Partially Lossless Compression of DICOM Image Sets*, Congreso Argentino de Informática y Salud (CAIS), 40 Jornadas JAIIO, 2011.
- [3] L. Shen y R. M. Rangayyan, *A Segmentation-Based Lossless Image Coding Method for High-Resolution Medical Image Compression*, IEEE Transactions On Medical Imaging, Vol. 16, No. 3, 1997.
- [4] E. Ferrante, M. del Fresno, P. Escobar y J. Massa, *Segmentación de MRI aplicada al Análisis de la Evolución de Tumores en Series Temporalmente Espaciadas*, Congreso Argentino de Informática y Salud (CAIS), 40 Jornadas JAIIO, 2011.
- [5] M. E. Brummer, R. M. Mersereau, R. L. Eisner y R. R. Lewine, *Automatic detection of brain contours in MRI data sets*, IEEE Transactions on medical Imaging 12(2) 153-166, 1993.
- [6] J. Hu y N. C. Beaulieu, *Accurate Simple Closed-Form Approximations to Rayleigh Sum Distributions and Densities*, IEEE Communications Letters Vol. 9, No. 2 1089-7798, 2005.
- [7] R. Srikanth y A. G. Ramakrishnan, *Contextual encoding in uniform and adaptive mesh-based lossless compression of MR images*, IEEE Trans. Medical Imaging, 24(9), 2005.
- [8] A. Corvetto y A. Ruedin, *Compresión de imágenes de resonancia magnética sin pérdida en la región de interés*, XII RPIC Reunión de Procesamiento de la Información y Control, 2007.
- [9] A. Corvetto, *Compresión de Estudios de Resonancia Magnética Basada en Segmentación, Transformaciones Enteras y Codificación Aritmética*, Departamento de Computación, FCEyN, UBA, 2007.
- [10] A. Barjatya, *Block Matching Algorithms For Motion Estimation*, IEEE. Preprint: <http://image.ing.bth.se/ipl-bth/Siamak.khatibi/computerVisionBTH10Summer/papers/BlockMatchingAlgorithmsForMotionEstimation.pdf>.
- [11] M. Antonini, M. Barlaud, P. Mathieu y I. Daubechies, *Image Coding Using Wavelet Transform*, IEEE Transactions on Image Processing, Vol. 1 No. 2, 1992.
- [12] R. D. Nowak, *Wavelet-Based Rician Noise Removal for Magnetic Resonance Imaging*, IEEE Transactions on Image Processing, 1997.
- [13] S. Osher y J. Sethian, *Fronts propagating with curvaturedependent speed: algorithms based on Hamilton-Jacobi formulations*, J. Comp. Phys., vol. 79, pp. 12-49, 1988.
- [14] V. Caselles, F. Catte, T. Coll y F. Dibos, *A geometric model for active contours in image*

- processing*, Numer. Math., vol. 66, pp. 1-31, 1993.
- [15] V. Caselles, R. Kimmel y G. Sapiro, *Geodesic active contours*, Int'l J. Comp. Vis., vol. 22, pp. 61-79, 1997.
- [16] R. Malladi, J. A. Sethian y B. C. Vemuri, *Shape modeling with front propagation: a level set approach*, IEEE Trans. Patt. Anal. Mach. Intell., vol. 17, pp. 158-175, 1995.
- [17] S. Kim y H. Lim, *A Hybrid Level Set Segmentation for Medical Imagery*, IEEE Nuclear Science Symposium Conference Record 0-7803-9221-3, 2005.
- [18] S. Osher y R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, New York, 2002.
- [19] I. H. Witten, R. M. Neal y J. G. Cleary, *Arithmetic Coding for Data Compression*, Communications of the ACM, Vol. 30, No. 6, 1987.
- [20] A. Moat, R. M. Neal y I. H. Witten, *Arithmetic Coding Revisited*, ACM Transactions on Information Systems Vol 16, Issue 3 256-294, 1988.
- [21] L. Chunming, X. Chenyang, G. Changfeng y M. D. Fox, *Level Set Evolution Without Re-initialization: A New Variational Formulation*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) 1063-6919, 2005.
- [22] G. Randers-Pehrson, *PNG (Portable Network Graphics) Specification, Version 1.2*, PNG Development Group, 1999.
- [23] M. W. Marcellin, M. J. Gormish, A. Bilgin y M. P. Boliek, *An Overview of JPEG-2000*, IEEE Data Compression Conference, pp. 523-541, 2000.
- [24] M. Weinberger, G. Seroussi y G. Sapiro, *The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS*, Hewlett Packard Computer Systems Laboratory HPL-98-193, 1998.
- [25] A. Said y W. A. Pearlman, *A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, pp. 243-250, 1996.
- [26] PKWARE Inc., *.ZIP File Format Specification Version 6.3.2*, <http://web.archive.org/web/20070928174718/http://www.pkware.com/documents/casestudies/APPNOTE.TXT>, 2007.
- [27] S. Masi, A. Ruedin y D. Acevedo, *Detección de Contornos para Segmentar Fondo y Figura en Imágenes de Resonancia Magnética (MRI)*, Congreso Argentino de Informática y Salud (CAIS), 40 Jornadas JAIIO, 2011.