

TESIS DE LICENCIATURA

COMPRESIÓN DE ESTUDIOS DE RESONANCIA  
MAGNÉTICA BASADA EN SEGMENTACIÓN,  
TRANSFORMACIONES ENTERAS Y  
CODIFICACIÓN ARITMÉTICA

Andrés L. Corvetto  
acorvett@dc.uba.ar

Directora: Dra. Ana Ruedin

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

Diciembre de 2007

# Resumen

En este trabajo se presentará un método de compresión orientado a estudios de resonancia magnética. El método está basado en la detección y segmentación del fondo ruidoso circundante a la figura de interés, seguido por una compresión sin pérdida de los datos relevantes.

Los estudios de resonancia magnética contienen áreas con información de diagnóstico muy sensible, por lo que su compresión debe realizarse sin pérdida. Por lo general, constan de una figura, que contiene los datos relevantes, rodeada por un fondo ruidoso.

El método propuesto consiste de dos etapas. La primera se ocupa de segmentar la figura relevante del fondo ruidoso. Para ello, se aplica un umbral, obteniéndose una máscara binaria inicial. Se aplican luego operaciones morfológicas sobre los cortes, para obtener la máscara final. La operación se repite en los tres ejes para mayor robustez.

La etapa siguiente comprime la figura segmentada sin pérdida. Para ello, primero se cuantiza utilizando el algoritmo k-medias. Se divide el estudio en dos arreglos, uno que contiene los datos cuantizados y otro que contiene los datos residuales de la cuantización, necesarios para reconstruir el original. Se calculan las diferencias entre los cortes cuantizados. Los datos obtenidos se reordenan antes de codificarlos con un codificador aritmético adaptativo. Los datos residuales, por su parte, son vectorizados, transformados con la transformada S+P y codificados con un codificador aritmético adaptativo.

Se aplicará el método a un conjunto de estudios reales y se exhibirán los resultados obtenidos.

# Agradecimientos

A la Dra. Silvia Kochen del Centro Municipal de Epilepsia de la División Neurología del Hospital Ramos Mejía, y a Jorge Calvar de FLENI, por facilitarnos estudios de resonancias magnéticas.

A mi directora Ana Ruedin, por la paciencia y el apoyo.

Al departamento de computación, por haberme brindado una educación gratuita y de excelencia.

A Agus, Ale I, Ana Laura, Bauna, Alita, Colton, Daniel G, Deniel, Diego L, Dieguito, Gaby, Herno, Ingru, Jaka, Jimena, Lean, Nat K, Nat P, Nicokei, Marjan, Mati, Pablenque, Ramak, Ro, Saša, Sil, Vale, Vero, Vic, y todos mis amigos con los que compartí tan buenos momentos estos años, dentro y fuera de la facultad.

A Mariana, por acompañarme durante este año.

A mi hermano Alejandro, mis primas Ana y Ceci, mi tío Paco, mis tías Elsa, Marisa y Silvia, y mis abuelos Alberto y Matilde, que me han apoyado siempre.

A mi mamá, por bancarme en todo, y a mi papá, que está siempre en mi recuerdo.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Organización del trabajo . . . . .	1
1.2. Objetivos . . . . .	1
1.3. Características de las imágenes de resonancia magnética . . . . .	3
1.4. Medidas de información y compresión . . . . .	4
<b>2. Técnicas utilizadas</b>	<b>5</b>
2.1. Ajuste a distribución Rayleigh . . . . .	5
2.2. Operaciones sobre imágenes binarias . . . . .	8
2.2.1. Rellenado . . . . .	8
2.2.2. Operaciones morfológicas . . . . .	8
2.3. Algoritmo K-medias . . . . .	9
2.4. Transformada entera S+P . . . . .	10
2.5. Codificador aritmético adaptativo . . . . .	11
<b>3. Método propuesto</b>	<b>12</b>
3.1. Esquema del compresor propuesto . . . . .	12
3.2. Definiciones previas . . . . .	13
3.3. Detección de fondo . . . . .	14
3.4. Segmentación y cuantización . . . . .	17
3.5. Procesamiento de los datos cuantizados . . . . .	18
3.5.1. Obtención de las diferencias . . . . .	18
3.5.2. Reordenamiento . . . . .	20
3.5.3. Codificación . . . . .	20
3.6. Procesamiento de los residuos . . . . .	22
3.6.1. Clasificación por clases y vectorización . . . . .	22
3.6.2. Aplicación de la transformada S+P . . . . .	22
3.6.3. Codificación . . . . .	23
3.7. Datos auxiliares . . . . .	24
3.8. Descompresión . . . . .	25
<b>4. Resultados</b>	<b>26</b>
4.1. Implementación . . . . .	26
4.2. Datos de prueba . . . . .	26
4.3. Tiempo de ejecución . . . . .	27

4.4. Detección de fondo . . . . .	27
4.5. Compresión . . . . .	29
4.6. Comparación con otros métodos . . . . .	30
4.7. Discusión . . . . .	33
4.7.1. Alternativas consideradas . . . . .	33
4.7.2. Eje de las diferencias . . . . .	33
4.7.3. Cantidad de clases utilizadas para segmentación . . . . .	34
4.7.4. Reordenamiento . . . . .	34
4.7.5. Codificación alternativa del arreglo <i>ESQ</i> . . . . .	35
4.8. Conclusiones . . . . .	37
4.9. Posible trabajo futuro . . . . .	38
<b>A. Anexos</b>	<b>39</b>
A.1. Método alternativo de detección de fondo . . . . .	39
A.1.1. Ventanas deslizantes . . . . .	39
A.1.2. Identificación del fondo . . . . .	40
A.1.3. Algoritmo . . . . .	41
A.1.4. Resultados . . . . .	42
A.2. Agregado de región de interés a la figura . . . . .	44

# Capítulo 1

## Introducción

### 1.1. Organización del trabajo

En este capítulo, se enumerarán los objetivos, se mostrarán algunas características de los estudios de resonancia magnética y se definirán algunas nociones básicas. En el capítulo 2 se presentarán brevemente las principales técnicas utilizadas en el método de compresión propuesto. En el capítulo 3 se detallarán los pasos del método. Finalmente, en el capítulo 4 se presentarán los resultados obtenidos, junto con las conclusiones e ideas para trabajo futuro.

### 1.2. Objetivos

Las imágenes de aplicación médica usadas para diagnóstico, como las tomografías computadas (*CT*) o las resonancias magnéticas (*MR*), contienen datos de importancia potencialmente vital. Generalmente, se encuentran compuestas por grandes volúmenes de datos, por lo que es imprescindible encontrar compresores eficientes para ahorrar espacio de almacenamiento y ancho de banda para su transferencia. Dada la importancia de los datos contenidos, el proceso de compresión y descompresión debe hacerse sin pérdida de información.

La información de un estudio de resonancia magnética consiste en una sucesión de rebanadas o cortes bidimensionales del objeto a ser analizado. Por lo general, las imágenes están fuertemente contaminadas con ruido. Cada corte contiene además sectores con información relevante para el diagnóstico, generalmente rodeados por un fondo no relevante.

Se han propuesto varios métodos de compresión sin pérdida utilizando diferencias entre cortes (Roos et al., 1988) seguido de un codificador de entropía, pero los autores concluyen que no es eficiente en imágenes de resonancia magnética debido al ruido que poseen. Asimismo, se utilizaron wavelets tridimensionales (Schiavi et al., 2004), segmentación de la imagen en grupos de tiempo y frecuencia mediante el equilibrado de los árboles generados por

paquetes de wavelets, seguidos de una entropía de orden mayor (Chiu et al., 2000). En general, estos métodos obtienen buenas tasas de compresión sacrificando bits de información en la parte de interés de la figura. Otros métodos propuestos más modernos involucran el uso de S-trees (Chan et al., 1987) o esquemas basados en mallas (Srikanth et al., 2005).

Se propone como primer paso un método de separación del fondo y la figura, que permite trabajar sólo con la parte relevante de cada corte, reemplazando el fondo por un valor sintético que lo representa. Es decir, se asigna una jerarquía mayor a la figura, y se la comprime sin pérdida. El fondo, por el contrario, es comprimido con pérdida de información. De esta manera, se obtiene una compresión eficiente de la imagen médica, sin perder información en la parte de interés. La detección de fondo en estudios de resonancia magnética ha sido también analizada y usada anteriormente. Se han estudiado métodos por crecimiento de regiones sin semilla (unseeded region growing) (Bai et al., 2004), operaciones morfológicas (Srikanth et al., 2005) y wavelets (Wu, 2003).

Para comprimir la figura, se propone un método basado en segmentación, cuantización, transformadas enteras y codificación aritmética adaptativa, que provee compresión sin pérdida de información relevante.

### 1.3. Características de las imágenes de resonancia magnética

La Fig. 1.1 muestra un corte de un estudio de resonancia magnética de una cabeza. En el centro de la imagen se observa el objeto de estudio, rodeado por un fondo oscuro y a primera vista uniforme. Si se aísla el fondo y se incrementa el contraste de la imagen, puede verse que no es uniforme, sino que contiene ruido.

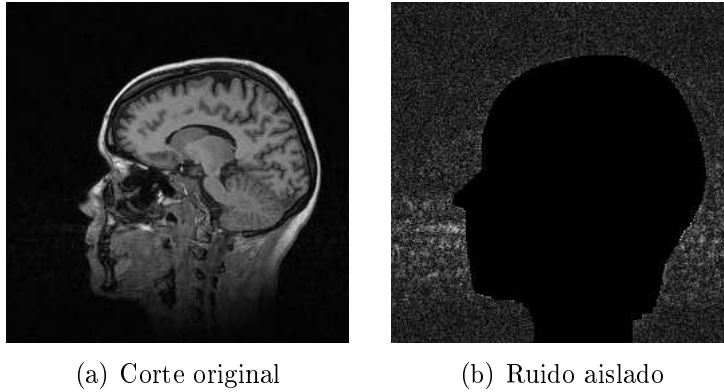


Figura 1.1: Un corte de un estudio de resonancia magnética de cabeza

Se ha estudiado la distribución del ruido en las imágenes de resonancia magnética y las formas en que puede ser filtrado (Nowak, 1997). En general, el ruido en dichas imágenes responde a una distribución Rician, que es dependiente de la señal, haciendo muy difícil el trabajo de filtrarlo. En el caso del fondo, al no haber señal se puede modelar el ruido como una distribución Rayleigh (Brummer et al., 1993):

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

En la sección 2.1 se presentarán algunas características de esta distribución.



## 1.4. Medidas de información y compresión

En esta sección se definen algunas convenciones de medición de información y compresión que serán utilizadas para medir la efectividad del método propuesto.

**Fuente** Una fuente 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** La entropía de una fuente  $F = (S, P)$ ,  $h_b(F)$  se define como

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

El valor de  $b$  determina la unidad de información. Si  $b = 2$ , la entropía es medida en bits.

Como medida, la entropía sirve para estimar la cantidad de bits por símbolo promedio que llevará la compresión de una fuente (por ej, un archivo).

**Tasa de compresión** La tasa de compresión  $TC$  de un determinado archivo o fuente finita de información, se define como:

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

donde  $o$  es el tamaño del archivo original y  $c$  es el tamaño del archivo comprimido.

Utilizando esta medida, una compresión a la mitad se nota “2:1” y una compresión a un quinto “5:1”.

# Capítulo 2

## Técnicas utilizadas

### 2.1. Ajuste a distribución Rayleigh

Dado un conjunto de datos experimentales que responden a una distribución estadística, se propone un método para encontrar los parámetros de dicha distribución. Sea  $X$  una variable aleatoria entera que responde, en teoría a una distribución Rayleigh (Papoulis , 1984)

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

de la que se tiene una muestra experimental.

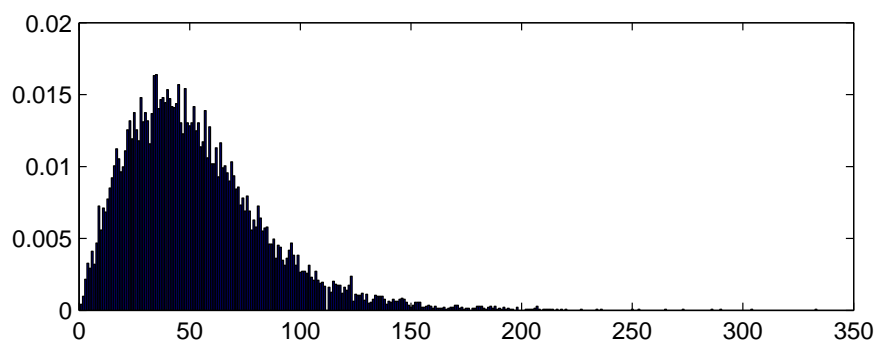


Figura 2.1: Una distribución experimental Rayleigh

Sea  $hist_X$  el histograma de dicha muestra experimental. La figura 2.1 muestra un ejemplo posible de  $hist_X$ . Sea  $hist_X(x)$  la frecuencia del valor  $x$  en  $X$ .

Para obtener el parámetro  $\sigma$  que más aproxima los datos experimentales, se propone encontrar un  $\sigma'$  que minimice la distancia:

$$d(\sigma') = \sum_{x \in M} (hist_X(x) - p'_R(x))^2$$

donde

$$p'_R(x) = \frac{x}{\sigma'^2} e^{-(x^2/2\sigma'^2)}$$

La figura 2.2 muestra el resultado de una aproximación con este método.

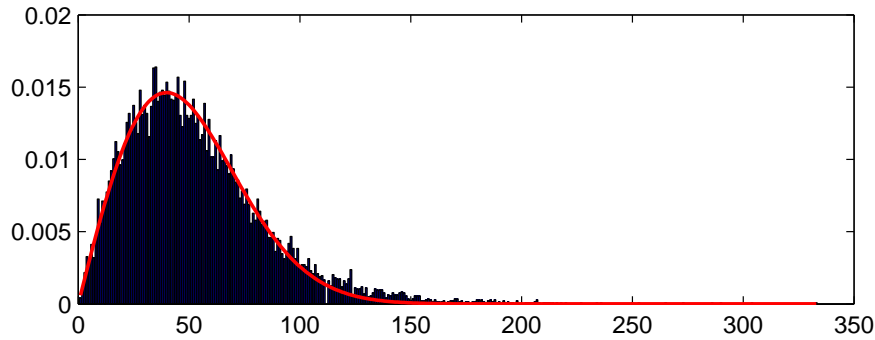


Figura 2.2: Aproximación a la distribución mostrada en la figura 2.1.  
( $\sigma = 39,8241$ )

Una posibilidad que puede presentarse es que la muestra no sea pura, es decir que la muestra contenga sea la suma de dos variables aleatorias, una de las cuales sea de distribución conocida (Rayleigh en este caso). En tal caso se tiene una variable  $X'$  tal que:

$$X' = X + Y$$

donde  $X$  responde a una distribución Rayleigh, e  $Y$  es de una distribución desconocida. La figura 2.3 muestra un histograma ejemplo.

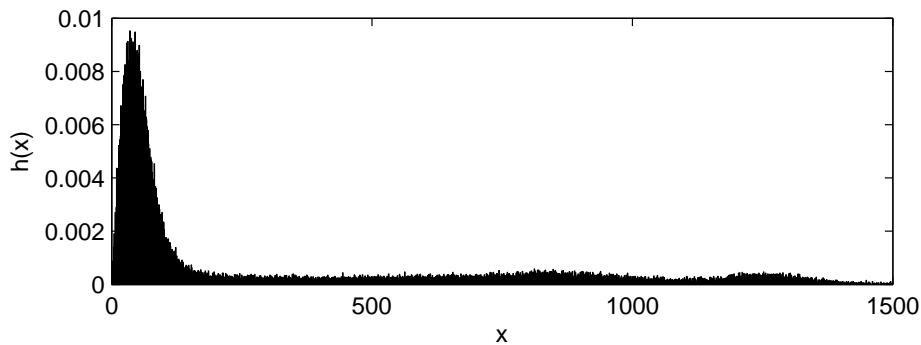


Figura 2.3: Una distribución mixta  $X'$ . Rayleigh ( $X$ ) + distribución desconocida de menor intensidad ( $Y$ )

Si  $X$  aporta un porcentaje importante de valores y los valores de  $Y$  se concentran en un rango distinto que los de  $X$ , es posible ajustar  $X$  a una distribución Rayleigh. Sin embargo, no alcanza con minimizar la  $d(\sigma')$  propuesta antes, pues está diseñada para ajustarse a distribuciones Rayleigh puras (como  $X$ ).

Sea  $hist_X(x)$  el valor de  $x$  en el histograma de  $X$  y  $hist_Y(x)$  el valor de  $x$  en el histograma de  $Y$ . Para cada uno de los histogramas, vale

$$\sum_{x \in X} hist_X(x) = 1$$

$$\sum_{x \in Y} hist_Y(x) = 1$$

Al combinarlas en  $X'$ , cada variable aporta datos en una proporción determinada. Si  $s$  es la proporción de datos aportados por  $X$ , para que la suma total de  $hist_{X'}$  sea 1 debe valer

$$\forall x \in X' : hist_{X'}(x) = hist_X(x) \times s + hist_Y(x) \times (1 - s)$$

Se agrega entonces una variable nueva  $s$  a la función distancia, que escala la distribución Rayleigh, permitiendo que la suma de sus valores sea menor a 1. Si se minimiza la nueva función distancia  $d'$ :

$$d'(\sigma', s) = \sum_{x \in X} (hist_X(x) - p'_R(x) \times s)^2$$

se obtienen parámetros  $\sigma'$  y  $s$  que aproximan la distribución de  $X$  como

$$p'_X(x) = s \times \frac{x}{\sigma'^2} e^{-(x^2/2\sigma'^2)}$$

La figura 2.4 muestra la aproximación del histograma de la figura 2.3.

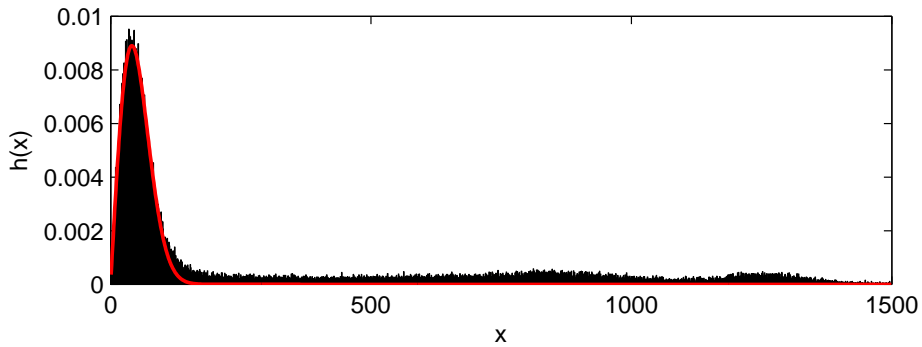


Figura 2.4: Aproximación a la distribución mostrada en la figura 2.3.  
( $\sigma = 40,8574$   $s = 0,6$ )

## 2.2. Operaciones sobre imágenes binarias

### 2.2.1. Rellenado

La operación de relleno consiste en rellenar todos los espacios vacíos no accesibles desde los bordes de una imagen. Para obtener dichos espacios, es posible hacer un “flood-fill” desde el borde en una copia de la imagen. Los espacios que quedan sin rellenar, son los que deben ser rellenados en la imagen original. Cada espacio se rellena utilizando el mismo algoritmo. La figura 2.5 muestra un ejemplo.

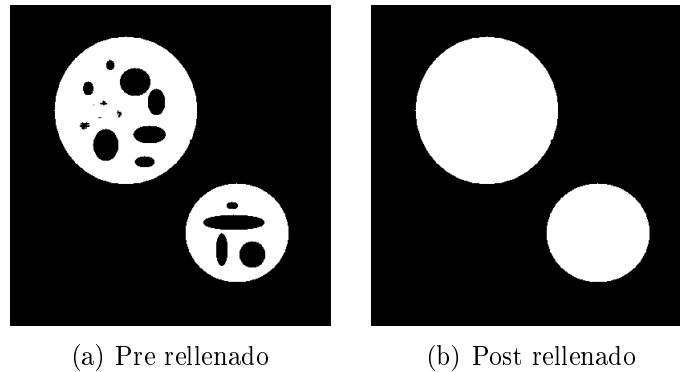


Figura 2.5: Operación de relleno

### 2.2.2. Operaciones morfológicas

La morfología matemática es una herramienta muy utilizada en el procesamiento de imágenes. Las operaciones morfológicas pueden simplificar los datos de una imagen, preservar las características esenciales y eliminar aspectos irrelevantes. Se utilizan sobre todo en imágenes binarias.

Las operaciones morfológicas básicas son la dilatación y la erosión. La erosión puede ser usada para la eliminación de elementos pequeños aislados, y es la que será usada en este trabajo. Consiste en desechar elementos que sean más pequeños que un determinado elemento estructural. La figura 2.6 muestra un ejemplo.

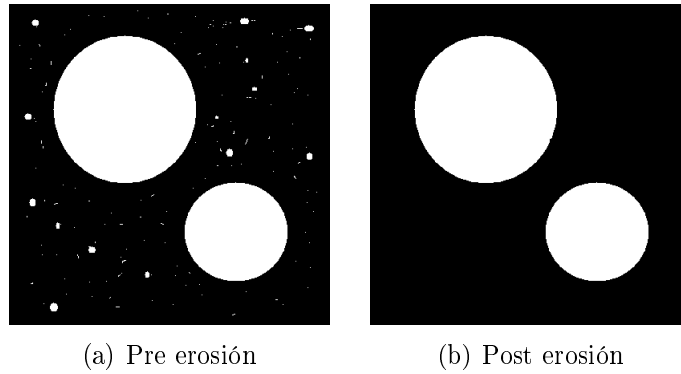


Figura 2.6: Operación de erosión

### 2.3. Algoritmo K-medias

El algoritmo K-medias (Hartigan et al., 1979) es un algoritmo para agrupar objetos según sus atributos en particiones. El objetivo del algoritmo es el de reducir la varianza inter-partición, o sea

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} \frac{|x_j - \mu_i|^2}{\#(S_i)}$$

donde hay  $k$  particiones  $S_i$  de tamaño  $\#(S_i)$ ,  $i = 1, 2, \dots, k$  y  $\mu_i$  es el centroide o media de todos los valores  $x_j \in S_i$ .

La implementación más común de K-medias es el algoritmo iterativo de Lloyd (Lloyd, 1982). Dicho algoritmo consiste en dividir inicialmente los valores en  $k$  particiones. Luego calcula la media o centroide de cada partición, y calcula nuevas particiones, asociando cada elemento con el centroide más cercano. Se itera nuevamente calculando centroides y nuevas particiones hasta que no hay cambios entre iteraciones.

## 2.4. Transformada entera S+P

La transformada entera más simple es una variación de la Wavelet Haar, llamada Transformada S (Ranganath et al., 1988), que se define de la siguiente forma: Sea  $s_n$  una señal con  $n = 0 \dots N - 1$ . Cada paso de la transformada S se define como las secuencias  $c_n$  (aproximación) y  $d_n$  (detalle):

$$c_n = \left\lfloor \frac{s_{2n} + s_{2n+1}}{2} \right\rfloor \quad (2.1)$$

$$d_n = s_{2n} - s_{2n+1} \quad (2.2)$$

Al aplicar sucesivamente la fórmula a las secuencias de aproximación, se obtienen más niveles de la transformada.

La transformada entera S+P fue propuesta por Said y Pearlman (Said et al., 1996). Dicha transformada consiste en calcular la transformada S y luego aplicar una codificación diferencial predictiva en la secuencia de detalle. Más formalmente:

Sean  $c_n$ ,  $d_n$  el resultado de aplicar la transformada S a  $s_n$ . Calcular

$$d'_n = d_n - \left\lfloor \hat{d}_n + \frac{1}{2} \right\rfloor, \quad n = 0, 1, \dots, \frac{N}{2} - 1 \quad (2.3)$$

donde

$$\Delta c_n = c_{n-1} - c_n \quad (2.4)$$

y

$$\hat{d}_0 = \frac{\Delta c_1}{4} \quad (2.5)$$

$$\hat{d}_n = \frac{1}{4} \Delta c_n + \frac{3}{8} \Delta c_{n+1} - \frac{1}{4} d_{n+1}, \quad n = 1, 2, 3, \dots, \frac{N}{2} - 2 \quad (2.6)$$

$$\hat{d}_{\frac{N}{2}-1} = \frac{\Delta c_{\frac{N}{2}-1}}{4} \quad (2.7)$$

La transformada S+P de  $s$  se define entonces como  $r = (c_n, d'_n)$ . Nuevamente, para obtener varios niveles, se aplica la transformada sucesivamente a las secuencias de aproximación.

La antitransformada S+P se obtiene aplicando las mismas operaciones en orden inverso. En primer lugar, se deben obtener los  $d_n$  de la transformada S a partir de los  $d'_n$ . Se calcula:

$$d_n = d'_n + \left\lfloor \hat{d}_n + \frac{1}{2} \right\rfloor, \quad n = \frac{N}{2} - 1, \frac{N}{2} - 2, \dots, 0 \quad (2.8)$$

donde los  $\hat{d}_n$  se calculan a partir de las ecuaciones (2.6), (2.5) y (2.7). Luego, se aplica la antitransformada S, obteniéndose la señal original  $s$ .

## 2.5. Codificador aritmético adaptativo

Se utilizará un codificador aritmético adaptativo (Witten et al., 1987). Este tipo de codificador representa la información de forma más compacta que una codificación por Huffman en la mayoría de los casos (Moffat et al., 1998), 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.



# Capítulo 3

## Método propuesto

### 3.1. Esquema del compresor propuesto

La figura 3.1 presenta los pasos del compresor propuesto. En las siguientes secciones se dan los detalles de cada etapa.

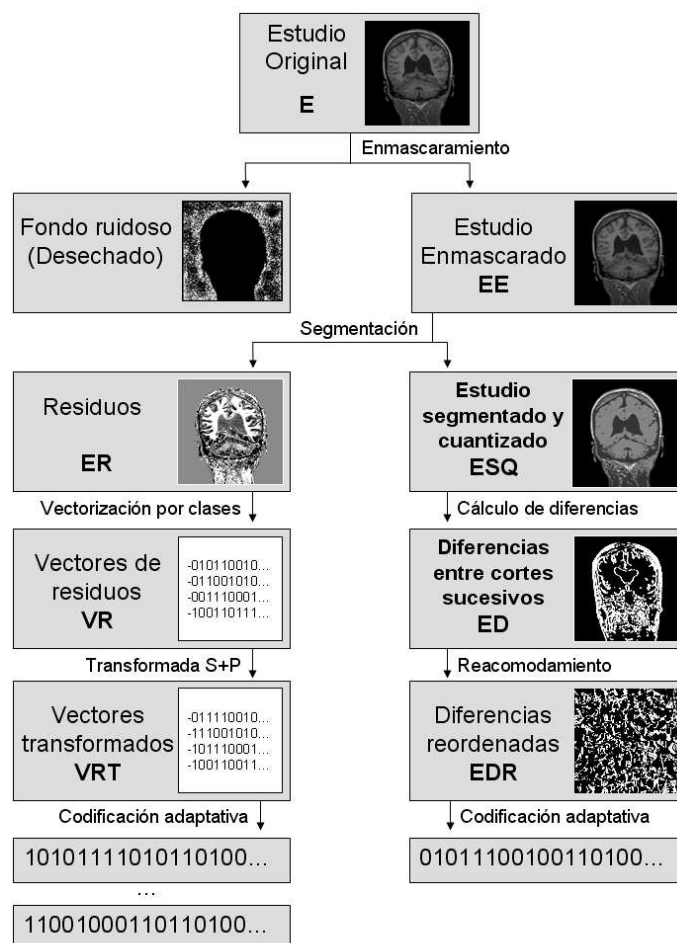


Figura 3.1: Esquema del compresor propuesto

### 3.2. Definiciones previas

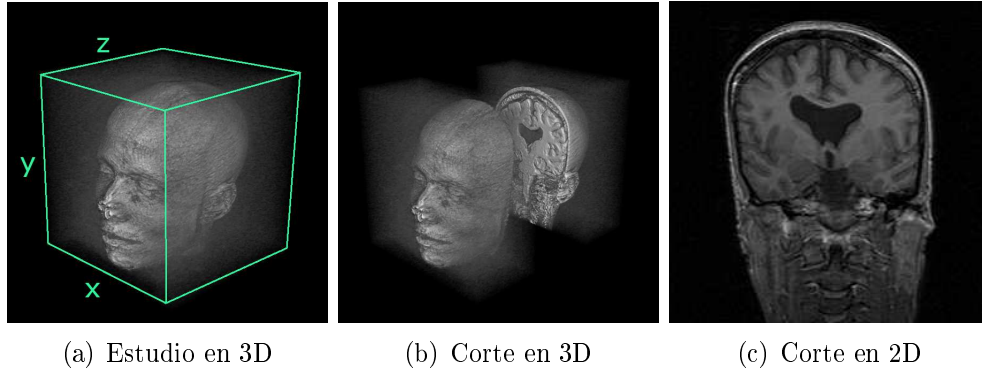


Figura 3.2: Un estudio MRI

Para formalizar los pasos se define un estudio MRI  $E$  de dimensiones  $N_X \times N_Y \times N_Z$ , en los ejes de coordenadas  $x$ ,  $y$  y  $z$  (Visualizado en la figura 3.2. Es decir, cada estudio es un arreglo tridimensional que escribimos

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

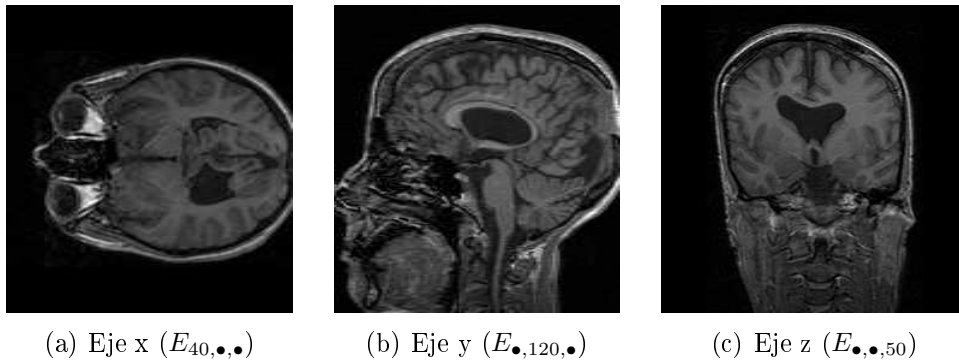
Particularmente,

$$E_{i,\bullet,\bullet} = E_{i,j,k} \text{ para } 1 \leq j \leq N_X, 1 \leq k \leq N_Z$$

$$E_{\bullet,j,\bullet} = E_{i,j,k} \text{ para } 1 \leq i \leq N_Y, 1 \leq k \leq N_Z$$

$$E_{\bullet,\bullet,k} = E_{i,j,k} \text{ para } 1 \leq i \leq N_X, 1 \leq j \leq N_Y$$

son cortes del estudio en los ejes  $x$ ,  $y$  y  $z$ , respectivamente. Cada posición del arreglo tridimensional (voxel) contiene un valor positivo o cero. La figura 3.3 muestra ejemplos de cortes en los 3 ejes.



(a) Eje x ( $E_{40,\bullet,\bullet}$ )

(b) Eje y ( $E_{\bullet,120,\bullet}$ )

(c) Eje z ( $E_{\bullet,\bullet,50}$ )

Figura 3.3: Cortes en los 3 ejes

### 3.3. Detección de fondo

La primera etapa del método consiste en la detección del fondo ruidoso de la figura. El resultado buscado es un arreglo tridimensional  $M$  del mismo tamaño que  $E$ , pero con sólo 2 valores posibles: 0 para indicar fondo y 1 para indicar figura. A este arreglo lo llamaremos máscara. Se propone un método similar al planteado en (Srikanth et al., 2005), que consiste en calcular y aplicar un umbral al estudio y luego procesar las máscaras obtenidas. El método aquí propuesto agrega una etapa final para sumar robustez al resultado.

El primer paso para obtener  $M$ , consiste en aplicar un umbral a los datos del estudio. Aquellos valores por debajo del umbral tendrán valor 0 en  $M$ . En (Brummer et al., 1993), se propone el siguiente método para obtener el umbral:

Sea  $h(f)$  el histograma de las intensidades de todos los voxels del estudio. Se aproxima una función de distribución Rayleigh  $r(f)$  para estimar la distribución del ruido de fondo en el estudio. Este procedimiento está detallado en la sección 2.1. Se define entonces una función  $g(f)$ , que representa la distribución de todos los voxels que no son fondo.

$$g(f) = |h(f) - r(f)|$$

El umbral de mínimo error  $\tau_n$  se puede calcular minimizando la expresión  $\varepsilon\tau$

$$\varepsilon\tau = \sum_{f=0}^{\tau_n-1} g(f) + \sum_{f=\tau_n}^{\infty} r(f)$$

La figura 3.4 muestra el histograma de un estudio, las funciones  $g(f)$  y  $r(f)$  y el valor del umbral calculado. La idea es encontrar el punto en el que los valores de la distribución Rayleigh pasan a ser menos significativos que los de la figura.

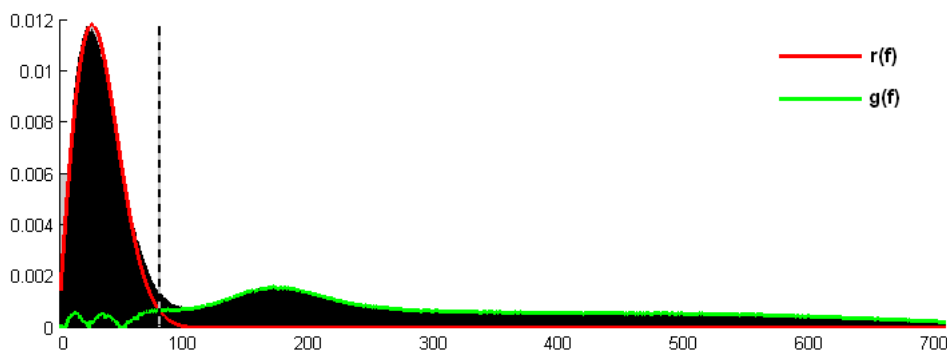


Figura 3.4: Cálculo de umbral utilizando ajuste a Rayleigh. La línea punteada marca el valor calculado.

En algunos casos el estudio puede haber sido pasado por un umbral previamente y no contener el histograma esperado. En ese caso, debe buscarse

el umbral de otra manera. Una posibilidad es analizar las intensidades más leves (que se supone aparecen con más frecuencia) y encontrar un mínimo local en el histograma donde converge la campana de los valores de fondo con los valores de la figura, y luego fijar el umbral en un valor ligeramente más bajo, para hacerlo más robusto. La figura 3.5 muestra un histograma con estas características, y el umbral correspondiente calculado con este método. Una opción final sería determinarlo manualmente a través de una interfaz apropiada.

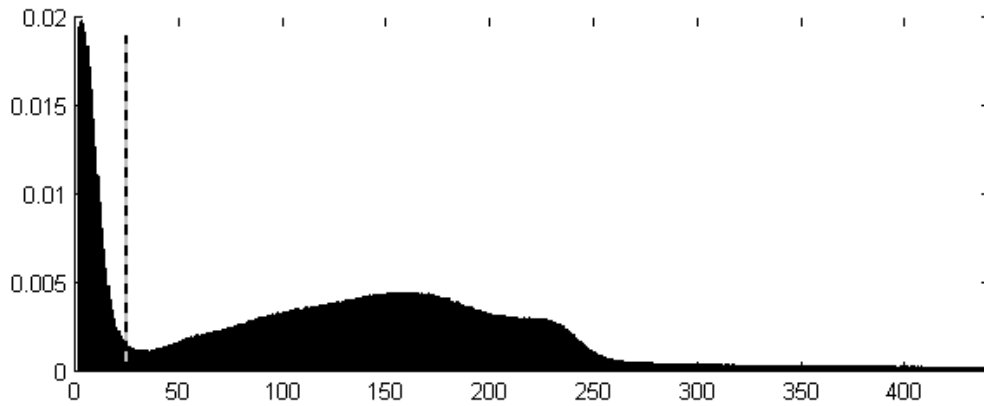


Figura 3.5: Cálculo de umbral encontrando un mínimo local. La línea punteada marca el valor calculado.

Al aplicar el umbral se obtiene una primera versión de la máscara, con 0 en los voxels que contenían valores por debajo y 1 en los que lo superaban. Esto no es suficiente, sin embargo, pues algunas porciones relevantes a la figura pueden quedar bajo el umbral. Se aplican entonces operaciones a la máscara binaria (ver sección 2.2) para solucionar este problema.

La primera operación es una operación de rellenado y la segunda de erosión. Los parámetros de la operación morfológica de erosión (elemento estructural, conectividad) deben ser obtenidos empíricamente. La figura 3.6 muestra la evolución de un corte a través de las operaciones. Las operaciones son aplicadas a cortes, es decir a secciones bidimensionales del estudio.

Una observación importante es que éste es el único paso del método en que es posible perder información importante (calculando mal la máscara), ya que se después se procede sin pérdida de información. Por lo tanto, es necesario hacer este paso lo más robusto posible.

Teniendo esto en cuenta, para obtener una detección más confiable del fondo y la figura, las operaciones son aplicadas a través de los cortes en los tres ejes del estudio ( $x$ ,  $y$  y  $z$ ). De esta manera, para cada voxel se tienen tres “votos” para considerarlo dentro de la máscara. La figura 3.7 muestra un corte con voxels con distinto grado de consenso sobre su relevancia. La cantidad de votos necesarios para considerar un voxel como parte de la figura (entre 1 y 3) es un parámetro del método. Para la parte experimental de esta tesis, se utilizó un valor de 1. De esta manera, llegado el caso de una

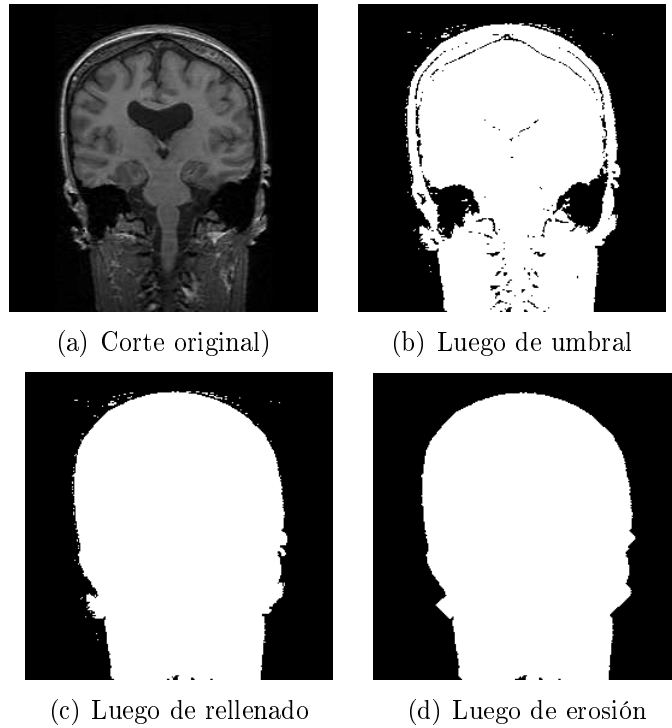


Figura 3.6: Etapas en la obtención de la máscara de un corte.

detección incorrecta, se cae en el error de considerar una parte del fondo como figura. Este error sólo implica una pérdida en la eficiencia de compresión, mientras que si se detecta erróneamente figura como fondo, se pierden datos potencialmente valiosos.

Todas las operaciones para obtener la máscara son efectuadas sobre una copia del estudio original, nunca alterando la información de la figura.

Una vez aplicadas las operaciones morfológicas se obtiene la versión final de la máscara  $M$ . Esta es aplicada al estudio para obtener el nuevo arreglo tridimensional  $EE$ : el estudio enmascarado. Es decir, todo aquél voxel de  $E$  que no esté contenido en la máscara es llevado a 0. Formalmente:

$$EE_{i,j,k} = \begin{cases} E_{i,j,k} & \text{si } M_{i,j,k} \neq 0 \\ 0 & \text{si } M_{i,j,k} = 0 \end{cases}$$

Es importante notar que no es necesario guardar la información de la máscara. Los datos contenidos en  $EE$  (la figura en un fondo homogéneo) son los únicos que van a ser codificados.

Finalmente, se deja abierta la posibilidad de reemplazar este algoritmo por otro con la misma función (en la sección A.1 se muestra una alternativa desarrollada durante esta investigación) y proseguir el procesamiento como se propone en las secciones siguientes.



Figura 3.7: Un corte con varios niveles de votación.

Los cuatro niveles de intensidad de cada pixel representan:

- negro: fondo.
- gris oscuro: 1 voto como figura, 2 como fondo.
- gris claro: 2 votos como figura, 1 como fondo.
- blanco: 3 votos como figura.

(La zona del interior del cuello no es detectada unánimemente como figura)

### 3.4. Segmentación y cuantización

Una vez aplicada la máscara al estudio, el paso siguiente es codificar la figura enmascarada. La idea es aprovechar la similitud entre voxels contiguos y codificar sólo una vez la información redundante que se pueda extraer de ellos. Se propone dividir el estudio en unas pocas zonas, de acuerdo con la intensidad de sus valores. Cada zona, al ser homogénea en su valor, puede ser representada de manera más compacta. La cantidad de clases elegida es pequeña (menos de 5), para maximizar la probabilidad de que vóxeles contiguos pertenezcan a la misma clase. Esto se analiza en la sección 4.7.3.

Se cuantiza entonces el arreglo obtenido en la sección anterior ( $EE$ ), que contiene a la figura en un fondo homogéneo. Para esto, se ejecuta el algoritmo  $k$ -medias descrito en la sección 2.3 sobre la intensidades de los datos de la figura completa. Se define una cantidad de clases  $c$ . Una vez obtenido el resultado, cada voxel es reemplazado por su elemento representativo. Se obtiene entonces un arreglo  $ESQ$  de las mismas dimensiones que  $EE$ , que representa al estudio segmentado y cuantizado. Se calcula también el arreglo tridimensional de residuos  $ER$  tal que

$$EE = ER + ESQ$$

Utilizando  $ER$  y  $ESQ$  se puede reconstruir el estudio enmascarado.

La figura 3.8 muestra cortes representativos de  $ER$  y  $ESQ$ . Cada uno de estos nuevos arreglos se procesará de una manera distinta. Las secciones siguientes detallan estos procesos.

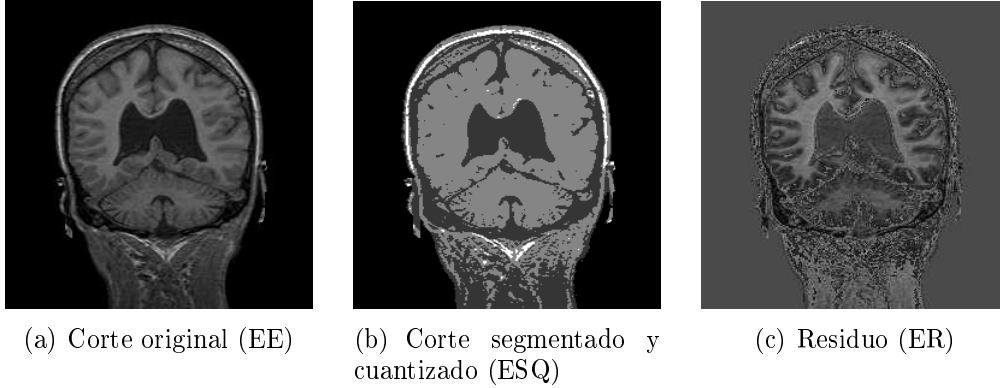


Figura 3.8: Un ejemplo de segmentación y cuantización de un corte de un estudio.  
(En este caso,  $c = 3$ )

## 3.5. Procesamiento de los datos cuantizados

En esta sección se describen los pasos a seguir para el procesamiento del arreglo tridimensional  $ESQ$  obtenido en 3.4.

### 3.5.1. Obtención de las diferencias

Dado el arreglo tridimensional  $ESQ$  (estudio segmentado y cuantizado), se procede a obtener un nuevo arreglo tridimensional  $ED$ , que alberga las diferencias entre los cortes sucesivos de  $ESQ$ , en algún eje en particular (ver sección 4.7.2). Formalmente (suponiendo que las diferencias se calculan en el eje  $z$ ):

$$ED_{\bullet,\bullet,1} = ESQ_{\bullet,\bullet,1}$$

$$ED_{\bullet,\bullet,k} = ESQ_{\bullet,\bullet,k} - ESQ_{\bullet,\bullet,k-1} \text{ para } 2 \leq k \leq N_Z$$

Las figuras 3.9 muestran una visualización del arreglo tridimensional  $ED$ . La principal característica de este nuevo arreglo es que muchos de sus valores son 0. Esto se debe al relativo parecido existente entre cortes sucesivos (ver figura 3.10). La consecuencia más importante es que la entropía de  $ED$  es menor que la de  $ESQ$ .

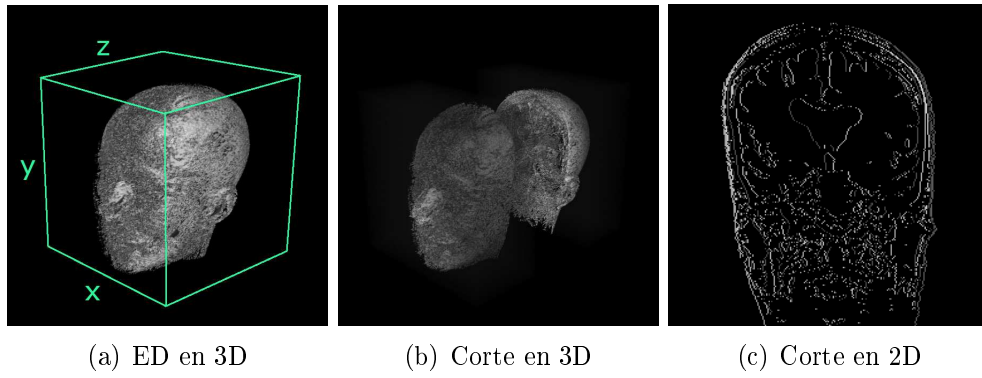


Figura 3.9: Arreglo *ED*

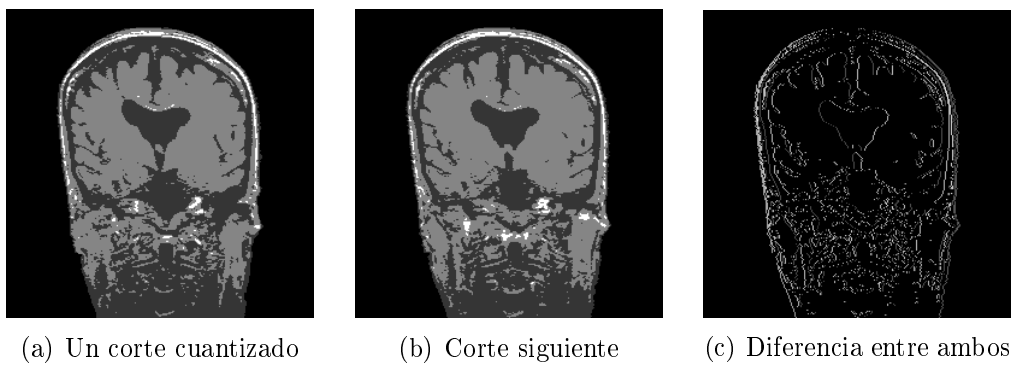


Figura 3.10: Diferencia entre 2 cortes cuantizados consecutivos.



### 3.5.2. Reordenamiento

Para codificar los datos contenidos en el arreglo tridimensional  $ED$ , se utilizará un codificador aritmético adaptativo. Para ello se requiere pasar los datos a un vector unidimensional. Esta vectorización se hace recorriendo secuencialmente todo el arreglo tridimensional, en un orden determinado (primero avanzando por un eje, luego por otro y finalmente por el restante). El orden que se elija cambia el tamaño final del archivo codificado.

Fijado un orden (por ejemplo,  $x - y - z$ ), para aprovechar la capacidad de adaptación del codificador, se propone una última transformación a  $ED$ . Dicha transformación consiste en un reordenamiento del arreglo para dejar la mayor cantidad de valores 0 iniciales en el vector a pasar al codificador. Esto mejora la eficiencia del codificador sensiblemente (ver sección 4.7.4).

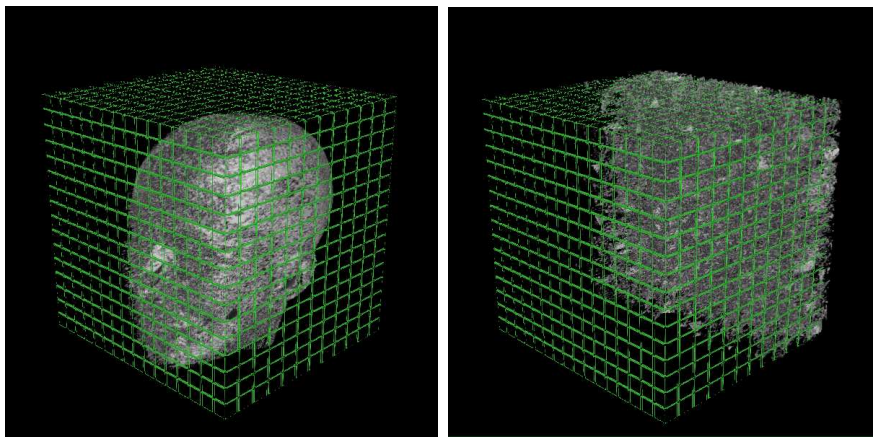
El reordenamiento se calcula de la siguiente manera:

- Se divide el arreglo  $ED$  en bloques tridimensionales de tamaño fijo definido ( $C_X \times C_Y \times C_Z$ ). Si es preciso, se modifica el tamaño del estudio para que sus dimensiones sean múltiplos de las dimensiones de los bloques.
- Se ordenan los bloques en función de la cantidad de valores 0 que contengan y se las reorganiza para obtener un nuevo arreglo tridimensional  $EDR$ , donde se concentran en un sector todas los bloques con valores nulos, de manera que al vectorizarlo, los valores nulos queden en general al principio del vector.

La figura 3.11 muestra una visualización del proceso y de reordenamiento.

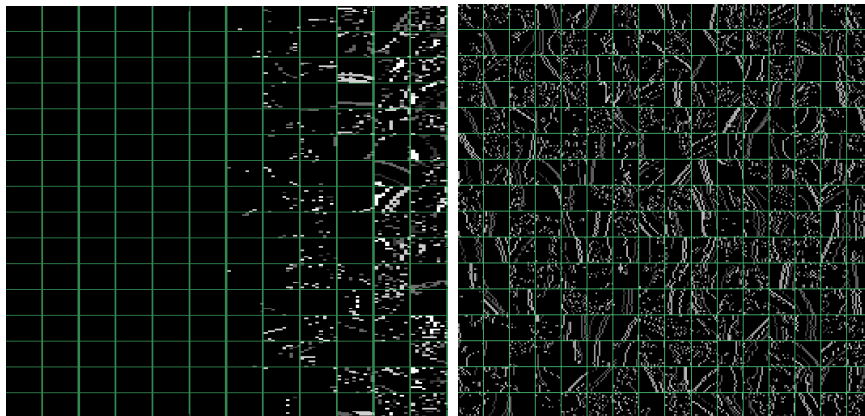
### 3.5.3. Codificación

El arreglo tridimensional  $EDR$  es vectorizado, dejando los valores 0 al principio del vector. Se aplica luego un codificador aritmético adaptativo, cuya eficiencia es mejorada por el reordenamiento (ver sección 4.7.4). Se obtiene así finalmente una tira de bits que se guarda en un archivo binario. Se guarda también el orden de los bloques para poder reconstruir el arreglo de diferencias (ver sección 3.7).



(a)  $ED$ , dividido en bloques

(b)  $EDR$ , mismos bloques reordenados



(c) Corte de  $EDR$  en el eje  $x$

(d) Corte de  $EDR$  en el eje  $z$

Figura 3.11: Reordenamiento

## 3.6. Procesamiento de los residuos

En esta sección se describen los pasos a seguir para el procesamiento del arreglo tridimensional  $ER$  obtenida en la sección 3.4.

### 3.6.1. Clasificación por clases y vectorización

El arreglo tridimensional  $ER$  es separado en  $c$  clases, correspondientes a la segmentación descrita en la sección 3.4. Cada una de esas clases es vectorizada, eliminando los intersticios. La figura 3.12 muestra una esquematización de este proceso.

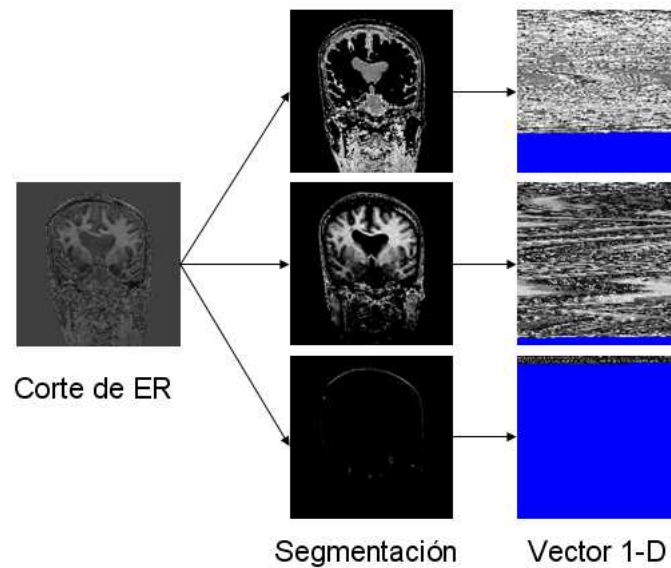


Figura 3.12: Clasificación por clases y vectorización

Se obtienen entonces  $c$  vectores unidimensionales  $VR_i$ .

Una consideración constante que se tuvo al procesar estos datos, es nunca codificar la misma información más de una vez. Se consideró en algún momento utilizar técnicas para transformar o codificar  $ER$  entero (ver sección 4.7.1), pero esto lleva a tener en cuenta nuevamente toda la zona circundante a la figura, que ya fue observada y codificada cuando se trató con el arreglo cuantizado. La vectorización en este caso, sirve entonces para tratar sólo con los valores dentro la figura.

### 3.6.2. Aplicación de la transformada S+P

A los vectores obtenidos en la sección 3.6.1 se les aplica la transformada S+P (sección 2.4) en varios niveles (la cantidad también se determinó experimentalmente), obteniéndose los vectores  $VRT_i$ . Esto reduce la entropía de los vectores.

Si es necesario, los vectores son rellenos con 0s para cumplir con las condiciones de aplicación de la transformada.

### 3.6.3. Codificación

Los vectores transformados  $VRT_i$  son finalmente codificados con un codificador aritmético adaptativo. Son añadidos al archivo binario donde se guardó la codificación de  $EDR$ .

### 3.7. Datos auxiliares

Para poder descomprimir exitosamente los datos codificados, se requieren datos auxiliares a los vectores codificados en las secciones 3.5.3 y 3.6.3. Estos datos auxiliares son (por orden de tamaño):

- El orden de reordenamiento de  $EDR$ .
- Los diccionarios de símbolos para las codificaciones.
- El tamaño original del estudio.
- El tamaño de los bloques utilizados para el reordenamiento.
- La cantidad de niveles de transformada S+P utilizada.

Un posible orden de codificación para un archivo único conteniendo todos los datos sería:

1. El tamaño original del estudio.
2. Los diccionarios de símbolos para las codificaciones.
3. El tamaño de los bloques utilizados para el reordenamiento.
4. El orden de reordenamiento de  $EDR$ .
5. Codificación de  $EDR$ .
6. La cantidad de niveles de transformada S+P utilizada.
7. Codificación de los vectores  $VRT_i$ .

Leyendo solo los items 1 al 5 puede reconstruirse el estudio cuantizado, que puede ser útil para una visualización rápida.

### 3.8. Descompresión

El proceso de descompresión se compone de los siguientes pasos:

1. A partir de la tira de bits, se reconstruye el arreglo tridimensional  $EDR_R$  ( $EDR$  reconstruido).
2. Se reordena el arreglo tridimensional  $EDR_R$  utilizando el orden guardado como dato auxiliar y se obtiene  $ED_R$ .
3. Utilizando un procedimiento inverso al cálculo de diferencias sobre  $ED_R$ , se obtiene  $ESQ_R$ . Este arreglo tridimensional se utilizará como "molde" para los datos de los residuos.
4. Cada vector de residuos es decodificado. Se obtienen los vectores  $VRT_{iR}$ .
5. Se aplica la antitransformada S+P 1D la cantidad de niveles correspondiente a los  $VRT_{iR}$ , y se obtienen los  $VR_{iR}$ .
6. Se crea un arreglo tridimensional vacío de  $N_X \times N_Y \times N_Z$ . Usando  $ESQ_R$  como guía, la información cada vector  $VR_{iR}$  es ubicada en el arreglo (Recordar que cada vector  $VR_i$  era obtenido utilizando las mismas clases que la cuantización de  $ESQ$ ). Cuando todos los vectores han sido acomodados, se tiene el arreglo tridimensional  $ER_R$ .
7. Se suman  $ESQ_R$  y  $ER_R$ , y se obtiene  $E_R$ , el estudio reconstruido.

# Capítulo 4

## Resultados

### 4.1. Implementación

El algoritmo fue implementado en MATLAB 7, utilizando algunas rutinas ya escritas. Se utilizaron las rutinas de operaciones morfológicas incluidas con la distribución de MATLAB y un codificador aritmético adaptativo ya implementado<sup>1</sup>.

### 4.2. Datos de prueba

Se probó el algoritmo con varios estudios de prueba. La tabla 4.1 muestra algunas de sus características. Los datos son por lo general guardados en archivos binarios en formato DICOM<sup>2</sup>, dedicando 16 bits para cada voxel. Este no es el valor óptimo en la mayoría de los casos. El valor de  $bpv_R$  de cada estudio  $E$  (sin comprimir) se calculó utilizando la fórmula:

$$bpv_R(E) = \lceil \log_2(\text{máx}(E)) \rceil$$

De esta manera,  $bpv_R$  representa el mínimo valor posible de bits por símbolo para el archivo sin comprimir. El  $tamaño_R$  se calcula utilizando  $bpv_R$ . Se da presenta también el  $tamaño_{16}$  calculado con 16 bits por símbolo:

$$tamaño_R = N_X \times N_Y \times N_Z \times bpv_R$$

$$tamaño_{16} = N_X \times N_Y \times N_Z \times 16$$

---

<sup>1</sup><http://www.diegm.uniud.it/bernardini/Software/Matlab/#aritm>

<sup>2</sup><http://dicom.nema.org/>

Estudio	Dimensiones en voxels	$bpv_R$ (bpv óptimo)	Tamaño en bits	
			calculado con $bpv_R$	calculado con 16 bpv
sk1	$256 \times 256 \times 99$	10	64,880,640	103,809,024
sk2	$256 \times 256 \times 109$	11	78,577,664	114,294,784
sk3	$256 \times 256 \times 94$	8	49,283,072	98,566,144
sk4	$256 \times 256 \times 119$	11	85,786,624	124,780,544
sk5	$256 \times 256 \times 119$	11	85,786,624	124,780,544
sk6	$256 \times 256 \times 119$	11	85,786,624	124,780,544
sk7	$256 \times 256 \times 119$	11	85,786,624	124,780,544
sk8	$256 \times 256 \times 109$	8	57,147,392	114,294,784
sk9	$256 \times 256 \times 129$	11	92,995,584	135,266,304
fl114	$512 \times 512 \times 18$	11	61,341,696	75,497,472
fl115	$512 \times 512 \times 18$	12	56,623,104	75,497,472
fl414	$512 \times 512 \times 17$	11	49,020,928	71,303,168
fl512	$512 \times 512 \times 20$	11	57,671,680	83,886,080
fl513	$512 \times 512 \times 20$	11	57,671,680	83,886,080
fl514	$512 \times 512 \times 20$	11	57,671,680	83,886,080

Tabla 4.1: Estudios de prueba

### 4.3. Tiempo de ejecución

El algoritmo fue implementado sin priorizar la velocidad de ejecución. Las pruebas fueron ejecutadas en Matlab 7, en una PC corriendo Windows XP con un procesador AMD Athlon 64 3000+, y 1 Gb de RAM. La etapa más costosa en tiempo resultó ser la identificación de fondo, ya que en ella se recorren todos los cortes del estudio en las 3 dimensiones, y por cada uno se aplican operaciones binarias, que insumen un alto costo en cuanto a tiempo. Las dimensiones del estudio tienen, como es esperable, una relación directa con el tiempo que se tarda en comprimirlo. La descompresión, por su parte, es un proceso más rápido, ya que no incluye ningún procesamiento realmente complejo. La tabla 4.2 muestra los tiempos de compresión y descompresión para algunos estudios.

### 4.4. Detección de fondo

Se verificó que la detección de fondo fuera correcta. Para los estudios elegidos, no se interfirió con el objeto de estudio. La figura 4.1 muestra algunos cortes de algunos estudios, con el fondo detectado resaltado. La figura 4.2 muestra el fondo detectado para 2 cortes. En estas imágenes se evidencia la naturaleza caótica del fondo ruidoso, y lo que se gana al evitar codificarlo.



Estudio	Compresión	Descompresión
sk1	256.1 s	101.4 s
sk3	222.0 s	69.4 s
sk7	295.2 s	110.0 s
sk9	322.7 s	123.0 s
f115	220.5 s	81.1 s
f513	214.4 s	61.1 s

Tabla 4.2: Tiempos de compresión / descompresión (medidos en segundos)

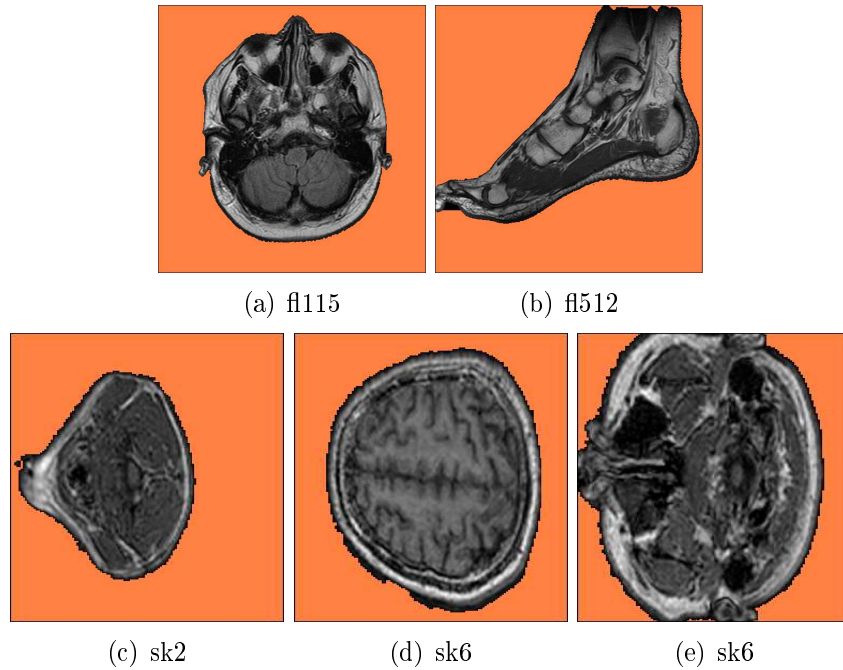


Figura 4.1: Muestras de resultados de detección de fondo

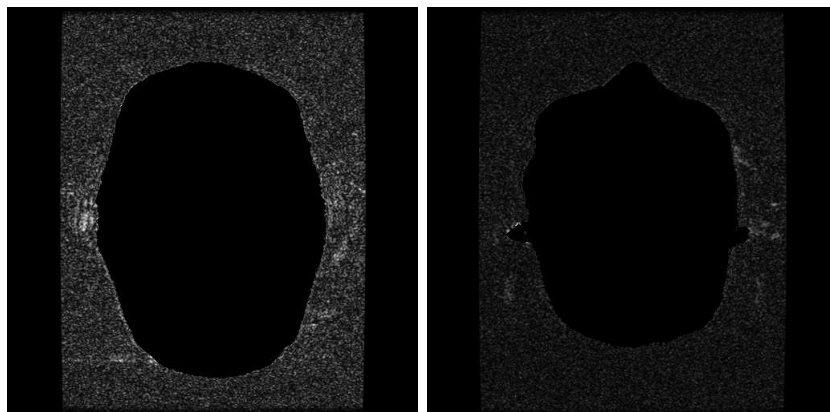


Figura 4.2: Fondo aislado para 2 cortes

## 4.5. Compresión

Las tablas 4.3 y 4.4 muestran los resultados obtenidos para la compresión de los estudios de prueba.

Los estudios fueron comprimidos con los siguientes parámetros (En la sección 4.7 se discute la elección de estos valores):

- $c = 3$  (Cantidad de clases).
- 3 niveles de transformada S+P.
- $C_X = 16$ ,  $C_Y = 16$ ,  $C_Z = 10$  (Dimensiones de los bloques para reordenamiento)
- Se utilizó el eje  $y$  para calcular las diferencias.

Estudio	Tamaño original en bits calculado con $bpv_R$	Voxels $c$ /info	Resultados de la compresión		
			Tamaño final en bits	Bits p/voxel (bpv)	Tasa de compresión
sk3	49,283,072	50.67 %	18,117,264	2.941	2.720:1
sk1	64,880,640	49.35 %	24,597,792	3.791	2.637:1
sk2	78,577,664	45.76 %	26,568,856	3.719	2.957:1
fl114	61,341,696	42.93 %	17,997,896	3.814	3.408:1
sk8	57,147,392	42.57 %	18,423,568	2.579	3.102:1
sk6	85,786,624	41.84 %	26,768,280	3.432	3.205:1
fl115	56,623,104	41.69 %	15,115,520	3.203	3.746:1
sk7	85,786,624	40.64 %	25,085,984	3.217	3.420:1
sk4	85,786,624	38.84 %	24,265,920	3.111	3.535:1
sk9	92,995,584	38.21 %	26,663,176	3.154	3.488:1
sk5	85,786,624	37.40 %	23,790,816	3.051	3.606:1
fl512	57,671,680	34.20 %	14,309,560	2.729	4.030:1
fl513	57,671,680	32.69 %	13,654,424	2.604	4.224:1
fl414	49,020,928	30.93 %	10,210,352	2.291	4.801:1
fl514	57,671,680	12.62 %	5,895,296	1.124	9.783:1

Tabla 4.3: Resultados de la compresión de los estudios, ordenados por la cantidad de voxels detectados como figura.

Las tasas de compresión fueron calculadas usando  $tamaño_R$  como tamaño original.

Es importante notar cómo el porcentaje de voxels con información relevante afecta al tamaño final del archivo comprimido, y por ende a la tasa de compresión.

Estudio	Tamaño original en bits calculado con 16 bpv	Voxels c/info	Tasa de compresión obtenida
sk3	98,566,144	50.67 %	5.440:1
sk1	103,809,024	49.35 %	4.220:1
sk2	114,294,784	45.76 %	4.302:1
fl114	75,497,472	42.93 %	4.195:1
sk8	114,294,784	42.57 %	6.204:1
sk6	124,780,544	41.84 %	4.662:1
fl115	75,497,472	41.69 %	4.995:1
sk7	124,780,544	40.64 %	4.974:1
sk4	124,780,544	38.84 %	5.142:1
sk9	135,266,304	38.21 %	5.073:1
sk5	124,780,544	37.40 %	5.245:1
fl512	83,886,080	34.20 %	5.862:1
fl513	83,886,080	32.69 %	6.143:1
fl414	71,303,168	30.93 %	6.983:1
fl514	83,886,080	12.62 %	14.229:1

Tabla 4.4: Tasas de compresión calculadas con  $tamaño_{16}$ 

## 4.6. Comparación con otros métodos

Para tener una referencia comparativa, se comprimieron los estudios con los siguientes métodos:

- ZIP: Cada estudio se escribió en un archivo binario, dedicando 16 bits a cada voxel. Luego se comprimió el archivo usando ZIP en máxima compresión.
- PNG: Se comprimió cada corte para cada estudio en formato PNG y se sumó el tamaño de los archivos resultantes.
- SPIHT: Idem PNG el compresor obtenido de <http://www.cipr.rpi.edu/research/SPIHT/spiht3.html>
- JPEG2000: Idem PNG, con el compresor obtenido de <http://jj2000.epfl.ch/>
- JPEG-LS (LOCO-I): Idem PNG, con compresor obtenido de <http://kt.ijs.si/aleks/jpeg-ls/mirror.htm>

Las tablas 4.5 y 4.6 muestran los resultados obtenidos:

En los resultados presentados, los otros métodos de compresión se aplicaron a los estudios completos, incluyendo el fondo ruidoso. Para tener una comparación del procesamiento de estudios enmascarados, se comprimieron

Estudio	Voxels c/info	ZIP	PNG	SPIHT	JPEG 2000	JPEG LS	Método prop.
sk3	50.67 %	4.860	4.134	<b>2.839</b>	3.061	3.078	2.941
sk1	49.35 %	6.492	5.826	4.157	4.540	4.289	<b>3.791</b>
sk2	45.76 %	6.545	6.079	4.276	4.907	4.410	<b>3.719</b>
fl114	42.93 %	8.041	6.877	4.614	6.293	4.807	<b>3.814</b>
sk8	42.57 %	4.561	4.074	2.733	3.002	2.992	<b>2.579</b>
sk6	41.84 %	6.478	6.025	4.255	4.953	4.386	<b>3.432</b>
fl115	41.69 %	8.941	7.600	5.093	6.356	5.404	<b>3.203</b>
sk7	40.64 %	6.307	5.854	4.110	4.574	4.249	<b>3.217</b>
sk4	38.84 %	6.125	5.781	4.094	4.677	4.221	<b>3.111</b>
sk9	38.21 %	6.274	5.934	4.184	4.822	4.312	<b>3.154</b>
sk5	37.40 %	6.224	5.892	4.160	4.751	4.297	<b>3.051</b>
fl512	34.20 %	8.451	7.221	4.762	5.780	5.189	<b>2.729</b>
fl513	32.69 %	7.003	6.118	4.211	5.176	4.380	<b>2.604</b>
fl414	30.93 %	6.402	5.513	3.589	4.437	3.847	<b>2.291</b>
fl514	12.62 %	8.940	8.260	6.089	6.692	6.318	<b>1.124</b>

Tabla 4.5: Comparación con otros métodos para estudios completos. (Resultados expresados en bpv)

Estudio	Voxels c/info	ZIP	PNG	SPIHT	JPEG 2000	JPEG LS	Método prop.
sk3	50.67 %	3.29:1	3.87:1	<b>5.64:1</b>	5.23:1	5.20:1	5.44:1
sk1	49.35 %	2.46:1	2.75:1	3.85:1	3.52:1	3.73:1	<b>4.22:1</b>
sk2	45.76 %	2.44:1	2.63:1	3.74:1	3.26:1	3.63:1	<b>4.30:1</b>
fl114	42.93 %	1.99:1	2.33:1	3.47:1	2.54:1	3.33:1	<b>4.19:1</b>
sk8	42.57 %	3.51:1	3.93:1	5.85:1	5.33:1	5.35:1	<b>6.20:1</b>
sk6	41.84 %	2.47:1	2.65:1	3.76:1	3.23:1	3.65:1	<b>4.66:1</b>
fl115	41.69 %	1.79:1	2.11:1	3.14:1	2.52:1	2.96:1	<b>4.99:1</b>
sk7	40.64 %	2.54:1	2.73:1	3.89:1	3.50:1	3.77:1	<b>4.97:1</b>
sk4	38.84 %	2.61:1	2.77:1	3.91:1	3.42:1	3.79:1	<b>5.14:1</b>
sk9	38.21 %	2.55:1	2.70:1	3.82:1	3.32:1	3.71:1	<b>5.07:1</b>
sk5	37.40 %	2.57:1	2.72:1	3.85:1	3.37:1	3.72:1	<b>5.24:1</b>
fl512	34.20 %	1.89:1	2.22:1	3.36:1	2.77:1	3.08:1	<b>5.86:1</b>
fl513	32.69 %	2.28:1	2.61:1	3.80:1	3.09:1	3.66:1	<b>6.14:1</b>
fl414	30.93 %	2.50:1	2.90:1	4.46:1	3.61:1	4.16:1	<b>6.98:1</b>
fl514	12.62 %	1.79:1	1.94:1	2.63:1	2.39:1	2.53:1	<b>14.23:1</b>

Tabla 4.6: Comparación con otros métodos para estudios completos (tasa de compresión calculada con  $tamaño_{16}$ )

también los estudios después de enmascararles y suprimirles el fondo (arreglo *EE*). Las tablas 4.7 y 4.8 muestran los resultados obtenidos. Tanto SPIHT como JPEG-LS muestran un mejor desempeño en este caso.

Estudio	Voxels c/info	ZIP	PNG	SPIHT	JPEG 2000	JPEG LS	Método prop.
sk3	50.67 %	4.079	3.366	<b>2.427</b>	2.555	2.473	2.941
sk1	49.35 %	5.195	4.498	3.333	3.588	<b>3.315</b>	3.791
sk2	45.76 %	4.983	4.471	3.290	3.795	<b>3.243</b>	3.719
fl114	42.93 %	5.308	4.625	<b>3.167</b>	4.730	3.1681	3.814
sk8	42.57 %	3.548	3.008	<b>2.175</b>	2.322	2.185	2.579
sk6	41.84 %	4.635	4.172	3.050	3.642	<b>3.006</b>	3.432
fl115	41.69 %	4.681	3.933	<b>2.622</b>	3.737	2.645	3.203
sk7	40.64 %	4.417	3.930	2.896	3.219	<b>2.841</b>	3.217
sk4	38.84 %	4.259	3.801	2.820	3.292	<b>2.765</b>	3.111
sk9	38.21 %	4.267	3.851	2.836	3.352	<b>2.778</b>	3.154
sk5	37.40 %	4.148	3.738	2.747	3.207	<b>2.691</b>	3.051
fl512	34.20 %	4.046	3.341	<b>2.188</b>	3.075	2.249	2.729
fl513	32.69 %	3.782	3.223	<b>2.179</b>	3.032	2.188	2.604
fl414	30.93 %	3.440	2.882	<b>1.835</b>	2.582	1.887	2.291
fl514	12.62 %	1.626	1.417	1.023	1.394	<b>0.969</b>	1.124

Tabla 4.7: Comparación con otros métodos para estudios enmascarados.  
(Resultados expresados en bpv)

En (Bai et al., 2004) se exhiben resultados para compresión sin pérdida de imágenes completas (con ruido de fondo), logrando 3.63 bpv para un corte. Hay que tener en cuenta, sin embargo, que las imágenes utilizadas son de 8 bits. En ese artículo se utiliza segmentación del fondo y la figura, y se define también una region de interés dentro de la figura, dándose la opción de comprimir con o sin pérdida. Se utiliza codificación Barrows-Wheeler y jpeg2000.

En (Srikanth et al., 2005) se exhiben resultados para un método que, como en el método propuesto, quita el ruido de fondo del estudio y luego lo comprime. En este caso, se utiliza un método que divide cada corte con una malla irregular. Luego se utiliza compensación de movimiento entre cortes y se codifica el diferencial. Se reportan resultados de 1.89 bpv para cada estudio, aunque nuevamente hay que tener en cuenta que se tratan de estudios de 8 bits. Por otra parte, el porcentaje de voxels que finalmente quedan dentro de la figura es crucial para el tamaño final del archivo comprimido.

Estudio	Voxels c/info	ZIP	PNG	SPIHT	JPEG 2000	JPEG LS	Método prop.
sk3	50.67 %	3.92:1	4.75:1	<b>6.59:1</b>	6.26:1	6.47:1	5.44:1
sk1	49.35 %	3.08:1	3.56:1	4.80:1	4.46:1	<b>4.83:1</b>	4.22:1
sk2	45.76 %	3.21:1	3.58:1	4.86:1	4.22:1	<b>4.93:1</b>	4.30:1
f114	42.93 %	3.01:1	3.46:1	<b>5.05:1</b>	3.38:1	5.05:1	4.19:1
sk8	42.57 %	4.51:1	5.32:1	<b>7.36:1</b>	6.89:1	7.32:1	6.20:1
sk6	41.84 %	3.45:1	3.83:1	5.25:1	4.39:1	<b>5.32:1</b>	4.66:1
f115	41.69 %	3.42:1	4.07:1	<b>6.10:1</b>	4.28:1	6.05:1	4.99:1
sk7	40.64 %	3.62:1	4.07:1	5.52:1	4.97:1	<b>5.63:1</b>	4.97:1
sk4	38.84 %	3.76:1	4.21:1	5.67:1	4.86:1	<b>5.79:1</b>	5.14:1
sk9	38.21 %	3.75:1	4.15:1	5.64:1	4.77:1	<b>5.76:1</b>	5.07:1
sk5	37.40 %	3.86:1	4.28:1	5.82:1	4.99:1	<b>5.95:1</b>	5.24:1
f1512	34.20 %	3.95:1	4.79:1	<b>7.31:1</b>	5.20:1	7.11:1	5.86:1
f1513	32.69 %	4.23:1	4.96:1	<b>7.34:1</b>	5.28:1	7.31:1	6.14:1
f414	30.93 %	4.65:1	5.55:1	<b>8.72:1</b>	6.20:1	8.48:1	6.98:1
f1514	12.62 %	9.84:1	11.29:1	15.63:1	11.48:1	<b>16.50:1</b>	14.23:1

Tabla 4.8: Comparación con otros métodos para estudios enmascarados (tasa de compresión calculada con  $tamaño_{16}$ )

## 4.7. Discusión

### 4.7.1. Alternativas consideradas

- El arreglo tridimensional  $ED$  (diferencias entre cortes cuantizados) puede considerarse una matriz semi-esparsa, por la cantidad de valores 0 que contiene. Una de las alternativas consideradas para codificarla, fue utilizar Run length encoding (RLE) o alguna variación, pero los resultados obtenidos no fueron comparables con los obtenidos con el codificador aritmético (incluso antes del reordenamiento)
- Se trabajó también en métodos alternativos de detección del fondo ruidoso. En la sección A.1 se detalla uno de ellos.
- Se intentó utilizar la transformada S+P en 2D para reducir la entropía de  $ER$ , aplicándola por cortes. El problema principal es que esta transformada incluye a los datos del fondo. Se desarrolló entonces una transformada S+P 2D enmascarada, pero los resultados no fueron mejores que la vectorización propuesta en la sección 3.6.1

### 4.7.2. Eje de las diferencias

El eje en el que se calculan las diferencias (sección 3.5.1) afecta a la compresión final. Se debe buscar el eje en el cual se detecte mayor correlación

entre cortes cuantizados. En el caso de estudios de cabezas, ese eje es el vertical.

Un procedimiento automático para determinar el eje podría ser considerar los 3 ejes y comparar la entropía del arreglo  $ED$  obtenida para cada uno. Esto tiene, por supuesto, un costo computacional.

### 4.7.3. Cantidad de clases utilizadas para segmentación

La cantidad de clases utilizadas para la segmentación y cuantización en los datos de pruebas fue obtenida experimentalmente. Se estimaron los bpv que resultarían de segmentar estudios con distintas cantidades de clases, considerando también los residuos obtenidos. La figura 4.3 muestra un gráfico comparativo para algunos estudios. Se utiliza un gráfico de líneas para poder identificar los mínimos.

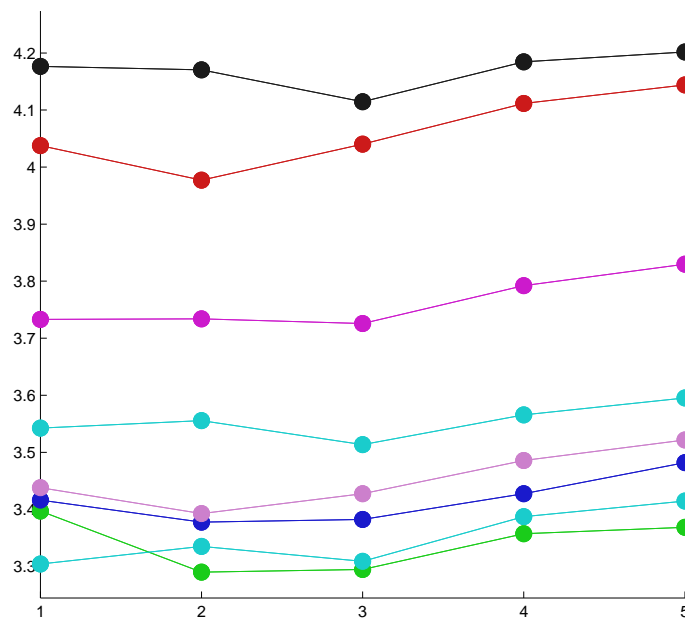


Figura 4.3: bpv estimados para distintas cantidades de clases.  
Cada color representa a un estudio.

Si bien las diferencias no son mayúsculas (notar que la escala de  $y$  no empieza en 0), se evidencia que el valor que minimiza los bpv está entre 2 y 3 clases.

### 4.7.4. Reordenamiento

El tamaño de los bloques para el reordenamiento (sección 3.5.2) también fue obtenido experimentalmente, balanceando el resultado de codificar

el arreglo tridimensional con el costo de guardar el orden de los bloques.

La tabla 4.9 muestra un cuadro comparativo para varios estudios. Se reordenó y codificó el arreglo  $ED$  utilizando distintos tamaños de bloque ( $C_X \times C_Y \times C_Z$ ). Los bpv calculados incluyen el costo de codificar el vector con el orden de los bloques. Se muestra también los bpv calculados para  $ED$  sin reordenamiento.

Estudio	s/reord	$16 \times 16 \times 10$	$8 \times 8 \times 5$	$16 \times 16 \times 5$	$32 \times 32 \times 10$
sk1	0.52996	<b>0.46836</b>	0.49093	0.47035	0.47796
sk2	0.56293	<b>0.49221</b>	0.50887	0.49376	0.51018
sk3	0.54057	<b>0.48381</b>	0.50370	0.48495	0.49312
sk4	0.44498	<b>0.38718</b>	0.40769	0.38813	0.40125
sk5	0.47623	<b>0.40993</b>	0.43226	0.41068	0.42139
sk6	0.51745	<b>0.45591</b>	0.47678	0.45651	0.46575
sk7	0.45475	<b>0.39614</b>	0.41783	0.39789	0.40794
sk8	0.50846	<b>0.43877</b>	0.45671	0.43953	0.45280
sk9	0.46780	<b>0.40116</b>	0.42230	0.40250	0.41300

Tabla 4.9: Resultados de codificar  $ED$  utilizando distintos tamaños de bloques para el reordenamiento. (Medidas en bpv)

#### 4.7.5. Codificación alternativa del arreglo $ESQ$

Se analizó la posibilidad de codificar el arreglo  $ESQ$  utilizando un método más directo. Se implementó una variación del algoritmo, donde el arreglo  $ESQ$  se codificó usando SPIHT por cortes, y el arreglo  $ER$  se codifica de la misma manera que la propuesta. La elección de SPIHT se basó en los resultados obtenidos para los estudios completos, donde tuvo la mejor eficiencia. La tabla 4.10 muestra los resultados obtenidos.

De estos resultados, se concluye que el método de compresión y codificación del arreglo  $ESQ$  presentado en este trabajo es más eficiente que la compresión por SPIHT por cortes. Esta ventaja está dada por las características particulares de  $ESQ$ , que permiten el reordenamiento propuesto para maximizar la eficiencia del codificador aritmético. La tabla 4.11 muestra los resultados de codificar sólo el arreglo  $ESQ$  para algunos estudios. La tercera columna muestra la diferencia porcentual entre los 2 valores calculados, obtenida con la siguiente fórmula:

$$diferenciaporcentual = \frac{(ESQ_{SPIHT} - ESQ_{PROP})}{ESQ_{SPIHT}} * 100$$

donde  $ESQ_{SPIHT}$  y  $ESQ_{PROP}$  son los bpv obtenidos codificando con SPIHT y con el método propuesto, respectivamente. Se puede apreciar que en algunos casos el método propuesto tiene una eficiencia hasta 14% mejor que SPIHT. En promedio, tiene una eficiencia 10,21% mejor.



Estudio	$E_{ESQ-SPIHT}$	$E_{ESQ-PROP}$
sk1	3.8580	<b>3.7912</b>
sk2	3.7493	<b>3.7193</b>
sk3	2.9965	<b>2.9409</b>
sk4	3.1651	<b>3.1115</b>
sk5	3.1044	<b>3.0506</b>
sk6	3.4701	<b>3.4324</b>
sk7	3.2847	<b>3.2167</b>
sk8	2.6289	<b>2.5791</b>
sk9	3.2109	<b>3.1539</b>
fl114	3.8602	<b>3.8140</b>
fl115	3.2483	<b>3.2034</b>
fl414	2.3246	<b>2.2911</b>
fl512	2.8167	<b>2.7293</b>
fl513	2.6463	<b>2.6044</b>
fl514	1.2034	<b>1.1244</b>

Tabla 4.10: Resultados para la compresión de estudios completos con la alternativa para la codificación de  $ESQ$ . (medidas en bpv)  
 $E_{ESQ-SPIHT}$ :  $ESQ$  codificado con SPIHT.  
 $E_{ESQ-PROP}$ :  $ESQ$  codificada con el método propuesto

Estudio	$ESQ_{SPIHT}$ (bpv)	$ESQ_{PROP}$ (bpv)	Diferencia porcentual
sk1	0.53210	<b>0.46836</b>	11.9790 %
sk2	0.51780	<b>0.49221</b>	4.9421 %
sk3	0.53591	<b>0.48381</b>	9.7218 %
sk4	0.43764	<b>0.38718</b>	11.5300 %
sk5	0.46116	<b>0.40993</b>	11.1089 %
sk6	0.48999	<b>0.45591</b>	6.9552 %
sk7	0.46107	<b>0.39614</b>	14.0825 %
sk8	0.48571	<b>0.43877</b>	9.6642 %
sk9	0.45540	<b>0.40116</b>	11.9104 %

Tabla 4.11: Resultados para la compresión de matrices  $ESQ$  con el método propuesto y con SPIHT

## 4.8. Conclusiones

Se ha presentado un método de compresión de datos volumétricos orientado a estudios de resonancias magnéticas. La naturaleza ruidosa y caótica del fondo en los estudios hace que tenga una alta entropía y sea, por ende, difícil de comprimir eficientemente. El método presentado separa a la figura relevante del fondo ruidoso, reemplazándolo por un valor nulo y comprime sin pérdida la información contenida en la figura.

Para la separación de fondo se ha utilizado un método ya propuesto en trabajos anteriores que consiste en calcular un umbral que separe los histogramas de ruido y figura. Luego de aplicar el umbral se aplican operaciones morfológicas y de rellenado sobre cada corte del estudio. Se ha extendido el método para mayor robustez, aplicándolo en las tres dimensiones del estudio, y luego considerando la cantidad de votos obtenidos para cada voxel.

Sobre el estudio enmascarado, se realiza una segmentación en 3D usando el algoritmo k-medias. Luego, se cuantiza el estudio utilizando las clases obtenidas. Para la compresión de la figura sobre el fondo nulo, el método propuesto divide la información en dos partes. Uno contiene al estudio cuantizado y el otro los residuos (la diferencia entre la cuantización y el estudio original). La codificación de los residuos se hace aplicando la transformada wavelet S+P y luego codificando con un codificador aritmético adaptativo.

En el procesamiento y codificación del estudio cuantizado es donde se encuentran los principales aportes de esta tesis. Se calculan primero las diferencias entre cortes sucesivos lo cual disminuye su entropía, y luego se reordenan en 3D para maximizar la eficiencia del codificador aritmético adaptativo usado finalmente para codificar. Con este procesamiento, que utiliza las particularidades de la información a comprimir, se obtienen resultados muy satisfactorios de compresión para el estudio cuantizado, superiores a la aplicación de compresores sin pérdida para imágenes, como SPIHT.

Se han mostrado y analizado alternativas al método propuesto y se ha evaluado su eficacia.

Se han presentado los resultados de comprimir estudios reales con el método propuesto. Los resultados son competitivos, obteniéndose tasas de compresión promedio de 5.56:1.

## 4.9. Posible trabajo futuro

Algunos posibles caminos para continuar este trabajo son:

- **Mejorar la codificación del arreglo  $ER$ .** Encontrar la forma de codificar el arreglo  $ER$  de una manera más eficiente, para mejorar la eficiencia total del compresor.
- **Agregar región de interés a la figura.** En la sección A.2 se describe una idea para comprimir la figura con pérdida, pero declarando una región de interés donde se conserve la información.
- **Interfaz gráfica.** Integrar el compresor a una interfaz gráfica, para validar la correcta separación de fondo y ajustar detalles.

# Apéndice A

## Anexos

### A.1. Método alternativo de detección de fondo

Durante la investigación, se desarrolló un método alternativo para detectar el fondo del estudio, aunque no fue usado en el método final. A continuación se presenta el método. Se redefinirán algunos conceptos para mayor claridad en la explicación.

#### A.1.1. Ventanas deslizantes

Inicialmente se promedian varios cortes consecutivos para tener promedios con menos ruido, en los cuales el procedimiento de reconocer el fondo es más sencillo. Se obtiene así una máscara de 2 dimensiones que identifica al fondo. Esta máscara se propaga luego en la tercera dimensión, promediando otros cortes, si se verifica que las estadísticas del nuevo promedio no se han alterado. Los promedios se realizarán sobre grupos de  $P$  cortes consecutivos, para identificar el fondo en los cortes centrales del grupo. Estos grupos se solapan, para poder continuar el proceso.

A continuación, se definen algunos elementos que luego se usarán en el algoritmo.

Cada estudio  $E$  se considera compuesto por  $N_z$  cortes  $E_{\bullet,\bullet,k}$  de dimensiones  $N_x \times N_y$ , como los definidos en la sección 3.2.

Se definen promedios  $\bar{E}^{(i)}$  de la siguiente forma:

$$\bar{E}^{(0)} = \text{promedio}(E_{\bullet,\bullet,1}, \dots, E_{\bullet,\bullet,P})$$

$$\bar{E}^{(i)} = \text{promedio}(E_{\bullet,\bullet,iJ+1}, \dots, E_{\bullet,\bullet,iJ+P})$$

donde  $J$  es el desplazamiento entre promedios sucesivos y  $P$  es la cantidad de cortes promediados. La Fig. A.1 esquematiza la selección de dos promedios consecutivos.

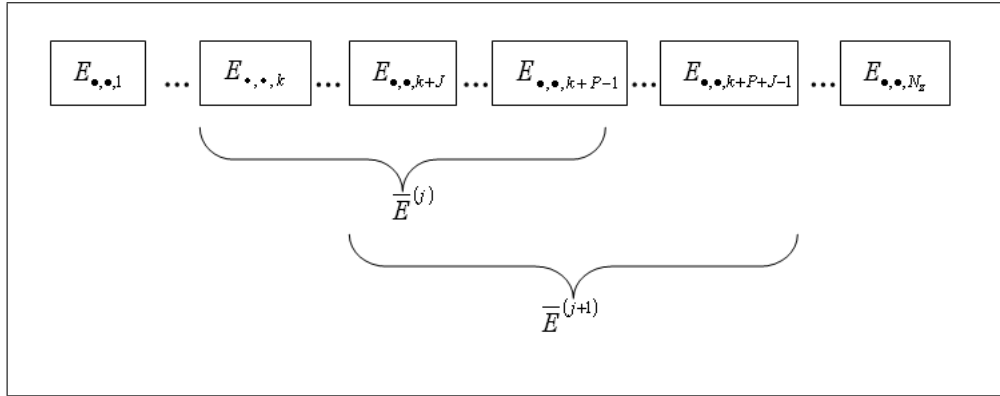


Figura A.1: Cortes contenidos en dos promedios consecutivos

Al identificar el fondo sobre el promedio  $\overline{E}^{(i)}$ , se construye una máscara para ese grupo. Esa máscara consta de  $N_x \times N_y$  valores, que tendrán un 1 si el valor corresponde al fondo, y un 0 si corresponde a la figura.

### A.1.2. Identificación del fondo

Para construir la máscara, se comienza desde los bordes superiores (del corte promedio) izquierdo y derecho, que pertenecen al fondo, y se van incrementando estas regiones. Para esto se aplica una ventana deslizante. La Fig. A.2 muestra dos posiciones en el deslizamiento.

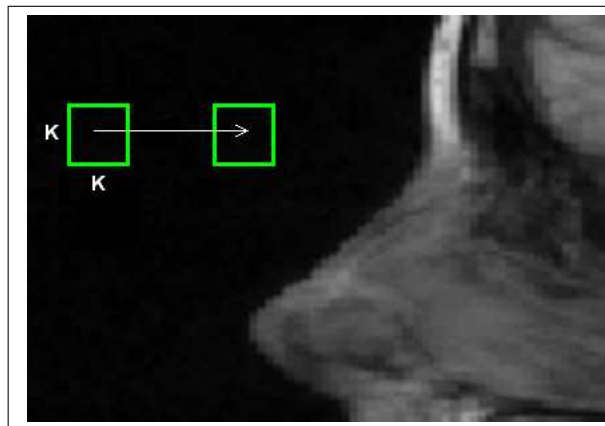


Figura A.2: Detalle de un promedio. Dos posiciones de la ventana deslizante. ( $K=15$  en este caso)

Se define entonces una ventana de  $K \times K$ , que se utilizará para recorrer

cada uno de los promedios  $\overline{E}^{(i)}$ :

$$V_{I,J}^{(s)} = \left\{ \overline{E}_{i,j}^{(s)} \text{ con } \begin{array}{l} i = I, \dots, I + K \\ j = J, \dots, J + K \end{array} \right\}$$

Se definen máscaras  $M^{(i)}$  de  $N_x \times N_y$  donde

$$M_{i,j}^{(s)} = \begin{cases} 0 \\ 1 \end{cases}$$

Existe una máscara  $M^{(i)}$  por cada promedio  $\overline{E}^{(i)}$ .

Finalmente, ventanas VM de  $K \times K$  sobre las máscaras:

$$VM_{I,J}^{(s)} = \left\{ M_{i,j}^{(s)} \text{ con } \begin{array}{l} i = I, \dots, I + K \\ j = J, \dots, J + K \end{array} \right\}$$

Se permite el abuso de notación:

$$VM_{i,j}^{(s)} \leftarrow x$$

para indicar que todos los elementos incluidos en  $VM_{i,j}^{(s)}$  deben tomar el valor  $x$ .

### A.1.3. Algoritmo

Se calculan primero los promedios  $\overline{E}^{(i)}$ , y por cada uno de ellos, se ejecuta el siguiente algoritmo:

Sean  $\epsilon_1, \epsilon_2$  valores de 2 umbrales. Notamos con  $\mu$  y  $\sigma$  respectivamente la media y la desviación estandar. Estos se calculan sobre la ventana aplicada al promedio del grupo actual.

$$i \leftarrow 1$$

$$j \leftarrow 1$$

$$M^{(s)} \leftarrow 0$$

$$\text{mientras } \left| \sigma \left( V_{i,j}^{(s)} - V_{i+1,j}^{(s)} \right) \right| < \epsilon_1 \left| \sigma \left( V_{i,j}^{(s)} \right) \right| \text{ y} \\ \left| \mu \left( V_{i,j}^{(s)} - V_{i+1,j}^{(s)} \right) \right| < \epsilon_2 \left| \mu \left( V_{i,j}^{(s)} \right) \right|$$

$$i \leftarrow i + 1$$

*mientras*  

$$\left| \sigma \left( V_{i,j}^{(s)} - V_{i,j+1}^{(s)} \right) \right| < \epsilon_1 \left| \sigma \left( V_{i,j}^{(s)} \right) \right| \text{ y}$$

$$\left| \mu \left( V_{i,j}^{(s)} - V_{i,j+1}^{(s)} \right) \right| < \epsilon_2 \left| \mu \left( V_{i,j}^{(s)} \right) \right|$$

$$j \leftarrow j + 1$$

$$VM_{i,j}^{(s)} \leftarrow 1$$
*fin mientras*  

$$j \leftarrow 1$$
*fin mientras*

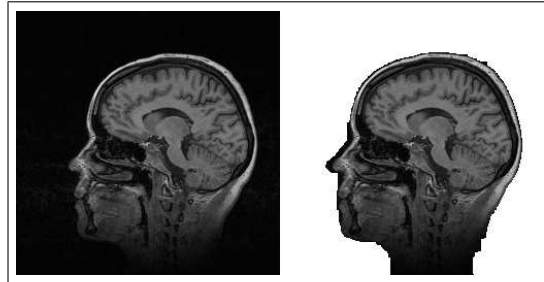
La idea es marcar en la máscara  $VM^{(s)}$  con 1's a todas las zonas que tengan una media y una desviación estándar similar a la encontrada en una ventana de  $K \times K$  en la parte superior izquierda del promedio. Se avanza desde el borde izquierdo hacia la derecha hasta encontrar una zona que no cumple con las condiciones, y se intenta nuevamente con la línea siguiente. Una vez ejecutado, debe ejecutarse un algoritmo similar, pero que avance desde la derecha hacia la izquierda.

Cada una de las máscaras  $VM^{(s)}$  es luego aplicada a los  $J$  cortes centrales del grupo que se utilizó para crear el promedio  $\bar{E}^{(s)}$ . De esta manera, en todos los  $i, j$  donde  $VM^{(s)} = 1$ , el corte toma valor 0, anulándose el ruido detectado.

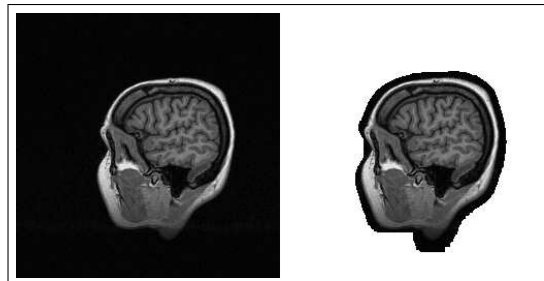
#### A.1.4. Resultados

La Fig. A.3 muestra algunos resultados obtenidos por el algoritmo. Lo que se ve en cada una de las figuras es el cuadro original a la izquierda y la máscara obtenida para dicho cuadro en el lado derecho (se superpone al cuadro original para mostrar el alcance no invasivo de la máscara).

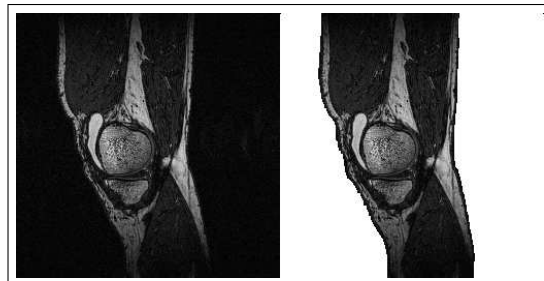
El problema que presenta este algoritmo es que, además de no ser performante, obtiene máscaras que son subóptimas. Esto se ve en los bordes de las máscaras, que no siempre coinciden con el borde de la figura. Esto se debe a que muchos cortes comparten la misma máscara.



(a) Corte de una cabeza



(b) Otro corte de una cabeza, menos cerca del centro



(c) Corte de una rodilla

Figura A.3: Resultados del algoritmo para algunos cortes. Izquierda: corte original, Derecha: corte con máscara superpuesta.



## A.2. Agregado de región de interés a la figura

Una posible mejora al método consistiría en comprimir también la figura con pérdida de información, pero definiendo una región de interés donde no ocurra pérdida.

Esta región de interés podría definirse en la etapa de segmentación y cuantización, haciendo una segmentación semántica, es decir que contemple la naturaleza de los objetos segmentados. Integrando el método con algún algoritmo de segmentación inteligente (que probablemente requiera algún tipo de intervención humana), se podría clasificar las clases de segmentación como relevantes/no relevantes. Aquellas clases relevantes, el método permanece igual, pero para las no relevantes, los datos correspondientes en la matriz  $ER$  podrían cuantizarse, bajando la entropía de los vectores correspondientes, y logrando un tamaño final aún menor.

Para un caso particular de estudio de resonancia magnética de cabeza, el objeto de estudio más común es el cerebro. Con un algoritmo o sistema que segmente el cerebro dentro de un estudio, las clases para cuantización se dividen en 2: las relevantes (que están dentro de la zona del cerebro) y el resto, no relevantes. La figura A.4 muestra una cuantización de un corte, previa segmentación del cerebro. Se pueden ver las clases marcadas como relevantes, con la información obtenida de la segmentación.

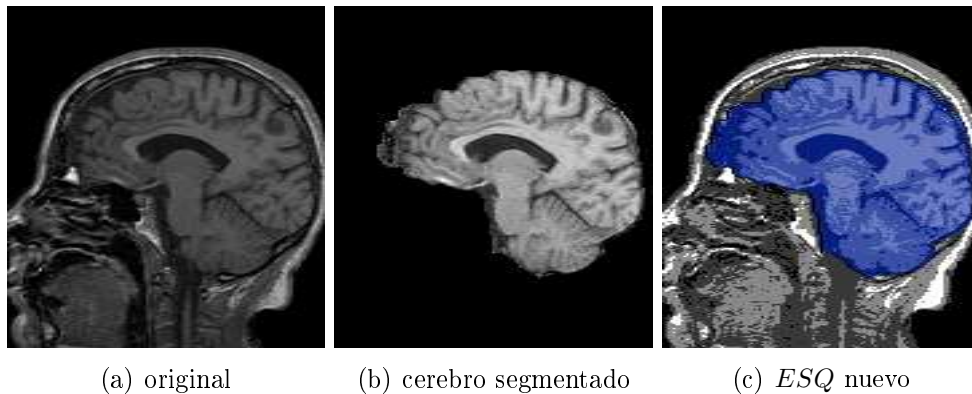
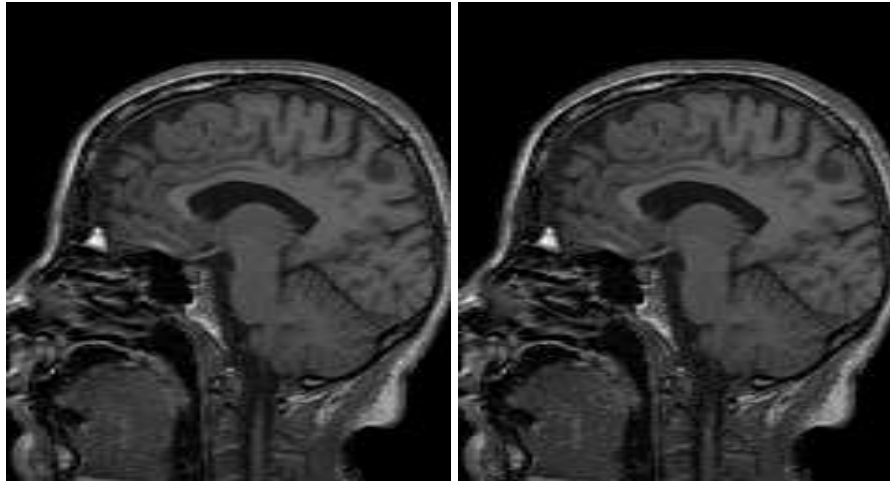


Figura A.4: **Segmentación semántica.** Las clases en azul(región de interés) son comprimidas con el método convencional. Las clases en gris son cuantizadas en  $ER$ .

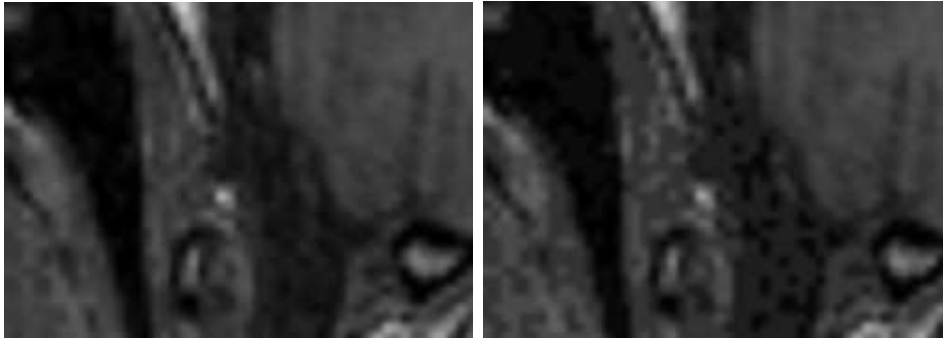
El arreglo  $ESQ$  se procesa de la misma manera que con el método convencional, pero los valores de  $ER$  son cuantizados solo para las clases no relevantes. De esta manera se disminuye la entropía de  $ER$  y los vectores  $VR_i$ . El proceso de descompresión es el mismo que el propuesto, por lo que este es un cambio menor al proceso en sí. La única dificultad radica en tener un buen algoritmo de segmentación semántica. La figura A.5 muestra la reconstrucción de un corte con y sin pérdida. Notar que la parte relevante (el

cerebro), en ambos casos es igual, pues no se cuantizó su respectivo vector  $VR_i$ .



(a) sin pérdida

(b) con pérdida



(c) sin pérdida(detalle)

(d) con pérdida(detalle)

Figura A.5: Resultados de compresión con pérdida en la parte no relevante. Las zonas que pertenecen al cerebro no pierden detalle.

Por último, es importante recalcar que la pérdida de detalle en zonas de importancia para diagnóstico no es aceptable, y debe ser evitada a toda costa. El algoritmo de segmentación debe ser, por lo tanto, muy confiable y robusto.

## REFERENCIAS

- Bai et al.(2004)** Xin Bai, Jesse S. Jin, Dagan Feng, “Segmentation-Based Multilayer Diagnosis Lossless Medical Image Compression,” *Pan-Sydney Area Workshop on Visual Information Processing (VIP2003), Sydney. Conferences in Research and Practice in Information Technology*, **Vol. 36** (2004).
- Brummer et al.(1993)** Marijn E. Brummer, Russell M. Mersereau, Robert L. Eisner, Richard R. J. Lewine, “Automatic detection of brain contours in MRI data sets.” *IEEE Transactions on Medical Imaging*, **12(2)**,153-166,(June 1993).
- Chan et al.(1987)** Chi-Shiang Chan, Chin-Chen Chang “A Lossless Medical Image Compression Scheme Using Modified S-tree Structure” *Proceedings of the 19th International Conference on Advanced Information Networking and Applications* 1550-445 (2005)
- Chiu et al.(2000)** E.Chiu, J. Vaisey, S. Atkins, “Joint Space-Frequency Segmentation, Entropy Coding and Compression of Ultrasound Images,” *Proceedings of the 2000 International Conference on Image Processing (ICIP 2000)*, IEEE Computer Society, Vancouver, Canada, (2000).
- Cover and Thomas(1991)** T. Cover, J. Thomas, “Elements of information theory,” Wiley-Interscience, (1991).
- Hartigan et al.(1979)** J. A. Hartigan, M. A. Wong, “A K-Means Clustering Algorithm”, *Applied Statistics*, **Vol. 28, No. 1**, 100-108 (1979)
- Lloyd(1982)** S. P. Lloyd “Least Squares Quantization in PCM”, *IEEE Transactions on Information Theory* **Vol. 28, no. 2** 129-137 (1982).
- Nowak(1997)** Robert D. Nowak, “Wavelet-Based Rician Noise Removal for Magnetic Resonance Imaging,” *IEEE Transactions on Image Processing*, (August 1997)
- Moffat et al.(1998)** A. Moffat, R.M. Neal, Ian H. Witten “Arithmetic Coding Revisited” *ACM Transactions on Information Systems* **Vol 16, Issue 3** 256-294 (July 1998)
- Ranganath et al.(1988)** S. Ranganath, H.Blume “Hierarchical image decomposition and filtering using the S-transform”, *Proc. SPIE* **vol 914** 799-814 (1988)
- Papoulis (1984)** Papoulis, A. “Probability, Random Variables, and Stochastic Processes, 2nd ed.” *McGraw-Hill*, 104 & 148 (1984)

- Roos et al.(1988)** P. Roos, M. Viergever, M. Van Dijke, J. Peters, “Reversible Intraframe Compression of Medical Images,” *IEEE Transactions on Medical Imaging, Pattern Recognition*, **Vol.7, No.4**, 328-336 (1988).
- Said et al.(1996)** A. Said W.A. Pearlman “An image multiresolution representation for lossless and lossy compression”, *IEEE transactions on image processing* **Vol 5** 1303-1310 (Sept 1996)
- Srikanth et al.(2005)** R. Srikanth and A. G. Ramakrishnan, “Contextual Encoding in Uniform and Adaptive Mesh-Based Lossless Compression of MR Images,” *IEEE Transactions on Medical Imaging*, **Vol. 24, NO. 9**, (September 2005).
- Schiavi et al.(2004)** E.Schiavi, C. Hernández, Juan A. Hernandez, “Fully 3D Wavelets MRI Compression,” J.M. Barreiro et al. (Eds.): ISBMDA 2004, LNCS 3337, 9-20, (2004).
- Witten et al.(1987)** I. H. Witten, Radford M. Neal , John G. “Cleary Arithmetic Coding for Data Compression”, *Communications of the ACM*, **Vol. 30, No. 6** (June 1987)
- Wu(2003)** Wu, Z.Q. Ware, J.A. Jiang, J, “Wavelet-based Rayleigh background removal in MRI”, *Electronics Letters* **Vol. 39, Issue: 7** 603-604 (April 2003)