

Implementación Paralela de Secuencias de Resonancia Magnética mediante Programación en GPU

E. Moya-Sáez¹, D. Treceño-Fernández¹, R. de Luis-García¹, F. Simmross-Wattenberg¹, C. Alberola-López¹

¹ Laboratorio de Procesado de Imagen (LPI), ETSI de Telecomunicación, Universidad de Valladolid.

Resumen

La simulación de imágenes de resonancia magnética presenta gran utilidad en ámbitos como la investigación o la educación. No obstante, el principal problema de este tipo de simulaciones es el gran coste computacional que conllevan, lo cual se traduce en tiempos de ejecución muy elevados. Sin embargo, dado que el problema es intrínsecamente paralelo, en este trabajo se presenta una implementación de una secuencia de pulsos *spin echo* - *echo planar imaging* (SE-EPI) utilizando técnicas de computación paralela en GPU. Los resultados muestran que este tipo de técnicas permiten reducciones drásticas de los tiempos de ejecución del algoritmo, permitiendo así nuevas posibilidades a este tipo de simulaciones.

1. Introducción

El uso de la resonancia magnética para la obtención de imágenes médicas, es una potente técnica no invasiva en donde la variación de los distintos parámetros y secuencias toma un papel fundamental en cómo el contraste entre los diferentes tejidos queda plasmado en las imágenes. Una simulación detallada de las secuencias incluyendo un seguimiento de la magnetización a lo largo del tiempo es necesaria, entre otras cosas, para la simulación de artefactos como las inhomogeneidades de campo, desplazamiento químico, imperfección en los gradientes y otros [1,2]. Asimismo, resulta de gran utilidad con fines educativos para la formación de personal clínico [3]. No obstante, este tipo de simulaciones implican un elevado coste computacional, lo que se traduce en que, mediante una programación secuencial, los tiempos de ejecución sean muy altos. Sin embargo, debido a que las mismas operaciones deben realizarse para cada vóxel de la imagen, el problema es intrínsecamente paralelo. Por este motivo, la introducción de técnicas de computación paralela en unidades gráficas de procesamiento (GPUs) permite un ahorro relevante en los tiempos de cálculo.

Otras propuestas para la simulación de imágenes de resonancia magnética en GPU, concretamente mediante CUDA-C, han sido planteadas [4]. A pesar de ello, dado que el objetivo final del simulador aquí desarrollado es la ejecución en un entorno menos potente y multiplataforma se optó por una simplificación que supla mejor estas necesidades. Asimismo, se realiza un estudio sistemático de la influencia de las inhomogeneidades del campo.

En este artículo se plantea una implementación de la secuencia de pulsos *spin echo* - *echo planar imaging* (SE-

EPI) mediante programación en GPU. Para ello, se detallará el método utilizado y la implementación realizada. Adicionalmente, se analizarán las imágenes obtenidas y se realizará un análisis de las medidas de rendimiento del algoritmo implementado en términos de tiempos de ejecución, en comparación con una implementación secuencial y mediante matrices *sparse* [5] del mismo.

2. Método

En la Figura 1 aparece representado un diagrama de bloques del simulador implementado.

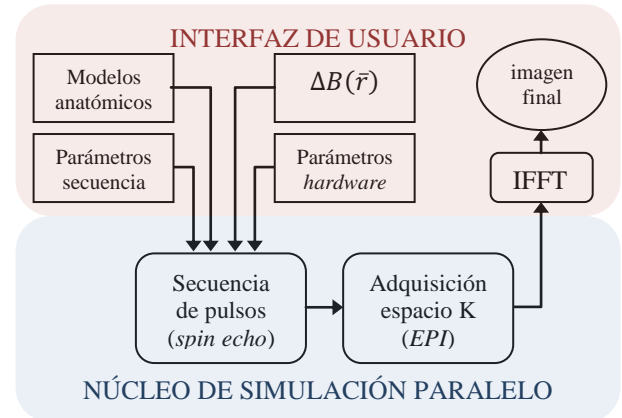


Figura 1. Diagrama de bloques del simulador.

2.1. Modelos anatómicos

La simulación usa como base un modelo anatómico formado por tres volúmenes contenedores de la información característica del tejido, es decir, el tiempo de relajación longitudinal (T1), el tiempo de relajación transversal (T2) y la densidad de protones (DP). El modelo utilizado a lo largo de este artículo es un *slice* de un corte axial del cerebro obtenido de una adquisición de resonancia magnética.

2.2. Secuencia de pulsos

En la Figura 2 está representada la secuencia de pulsos SE tratada en este artículo. No obstante, el método para la implementación de otra secuencia sería similar variando el orden, número y *flip angle* α de los pulsos.

El método que aplica la secuencia de pulsos sigue la expresión algebraica proveniente de la ecuación de Bloch, que describe la evolución del vector de magnetización neta \mathbf{M} en el tiempo [6].

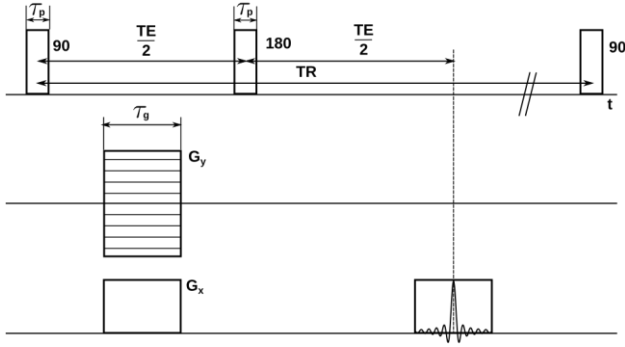


Figura 2. Diagrama de la secuencia spin echo.

$$\frac{d\mathbf{M}}{dt} = \gamma(\mathbf{M} \times \mathbf{B}_{eff}) - \frac{M_x \hat{i} + M_y \hat{j}}{T_2} - \frac{(M_z - M_z^0) \hat{k}}{T_1} \quad (1)$$

Donde γ es la constante giromagnética. M_x , M_y , y M_z , representan las componentes de \mathbf{M} en cada dirección, M_z^0 es el valor de equilibrio de \mathbf{M} bajo la presencia únicamente del campo magnético estático B_0 . Por otro lado, \mathbf{B}_{eff} representa el campo magnético total al que están sometidos los espines del tejido, y se puede expresar como:

$$\mathbf{B}_{eff}(\vec{r}, t) = (xG_x(t) + yG_y(t))\vec{z}' + \Delta B(\vec{r})\vec{z}' + B_1\vec{x}' \quad (2)$$

La resolución de la expresión vectorial de la ecuación (1), se realiza mediante operaciones matriciales sencillas de rotación y escalado [7]. Se pueden distinguir tres partes principales en la secuencia de pulsos:

- **Pulsos de RF:** en esta secuencia se aplican pulsos de 90° y 180° de duración τ_p segundos y amplitud constante B_1 que tendrán un valor dependiente del ángulo de giro. Dado que existe una inhomogeneidad de campo $\Delta B(\vec{r})$ función de cada punto del *slice* excitado, se producirá un efecto de *off-resonance* [6]. A efectos prácticos, la aplicación del pulso de RF es equivalente a aplicar una rotación sobre el vector de magnetización neta \mathbf{M} , por lo que será implementado mediante matrices de rotación [7].
- **Gradientes del campo magnético estático:** la aplicación de los gradientes en las direcciones de codificación de frecuencia y de fase (direcciones X e Y respectivamente), junto con la inhomogeneidad del campo, provocarán una rotación del vector de magnetización. Esto es simulado de igual modo mediante matrices de rotación [7].
- **Relajación de la señal:** tras la excitación de la señal, esta sufre un proceso de relajación, que es simulado mediante matrices de escalado [7].

Una explicación más detallada sobre el método para aplicar la secuencia de pulsos se puede encontrar en [5].

2.3. Adquisición del espacio K

Para la adquisición del espacio K se hace uso de la secuencia *EPI*. Esta secuencia se caracteriza por la obtención de un tren de ecos dirigidos por gradientes (gradientes de lectura en la dirección de codificación de frecuencia, junto con gradientes *blip* en la dirección de codificación de fase). Por este motivo se consiguen

imágenes en tiempos de captación muy breves (decenas de milisegundos) en comparación con otras secuencias [8]. En la secuencia *single-shot EPI* todo el espacio K se adquiere a partir de una única excitación de RF (*shot*). Esto trae consigo que se produzca una elevada atenuación de la señal. Una solución ampliamente utilizada para resolver este problema son las secuencias *multi-shot EPI*, en la que el espacio K se adquiere por bloques en función de un parámetro conocido como *Echo Train Length* (ETL) [8]. En una adquisición rectilínea del espacio K existen diferentes formas de ir adquiriendo los distintos bloques. Algunas de ellas se muestran en la Figura 3.

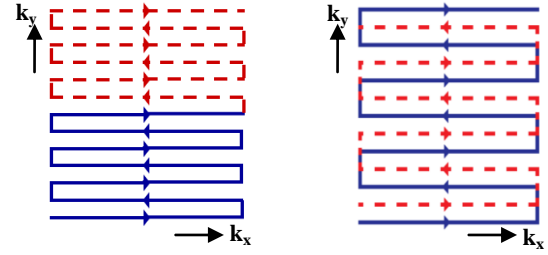


Figura 3. Recorrido del espacio K en secuencias multi-shot EPI secuencial (izquierda) y con interlineado (derecha).

3. Implementación

En la implementación realizada, como se puede ver en la Figura 1, se distinguen dos partes, la interfaz gráfica de usuario y el núcleo paralelo de simulación.

3.1. Interfaz gráfica de usuario

Se crea una interfaz gráfica de usuario en MATLAB con el objetivo de aumentar la usabilidad de la aplicación. En la interfaz se pueden seleccionar los modelos anatómicos de partida, así como modificar de forma dinámica algunos parámetros de la simulación tales como el mapa de inhomogeneidad del campo, el TE (tiempo de eco) y TR (tiempo de repetición), el ETL, el número de pulsos preparatorios y la forma de escritura del espacio K (secuencial o con interlineado). Una vez se han configurado todos los parámetros, desde la interfaz se ejecuta el núcleo de simulación paralelo, el cual retorna el espacio K correspondiente a la imagen simulada. Finalmente, en la interfaz se realiza la reconstrucción de la imagen y se muestra.

3.2. Núcleo de simulación paralelo

Se desarrolla un algoritmo para la simulación de síntesis de imágenes de resonancia magnética, utilizando la API y el lenguaje de programación OpenCL. Los *kernels* están diseñados buscando un paralelismo a nivel de datos. Este paralelismo explota el hecho de que cada vóxel del modelo anatómico puede ser tratado independientemente del resto. Siguiendo el método expuesto, se divide la secuencia de pulsos en cuatro fases separables, para las cuales se desarrollan *kernels* fácilmente reutilizables:

- **Pulsos de RF:** se crea un *kernel* que permita simular la aplicación de un pulso de RF de α grados.
- **Gradientes:** se crean varios *kernels* que permitan calcular los ángulos de rotación de la magnetización como consecuencia de los gradientes. Asimismo, se

desarrollan *kernels* para simular la aplicación de esos gradientes.

- **Relajación de la señal:** se crea un *kernel* para simular la relajación de la señal durante distintos intervalos de tiempo de acuerdo con la secuencia de pulsos implementada.
- **Espacio K:** se implementan varios *kernels* que permitan adquirir una línea del espacio K. Para ello es necesario sumar todos los elementos de la componente *X* (parte real del espacio K) y la componente *Y* (parte imaginaria del espacio K) de la magnetización para cada instante de tiempo.

El primer paso llevado a cabo en el algoritmo es la creación de objetos de memoria que permitan transferir datos desde la memoria de host (CPU) a la memoria global de la GPU y viceversa. Concretamente, se crean objetos de memoria contenedores del modelo anatómico y del mapa de inhomogeneidad del campo. Posteriormente, los *kernels* se instancian en un orden dependiente de la secuencia de pulsos a implementar. En cada *kernel* se computa en paralelo el nuevo vector de magnetización correspondiente a cada vóxel del modelo anatómico como consecuencia de los eventos producidos en la secuencia de pulsos. Para el caso concreto de la secuencia SE-EPI planteada en este artículo, se aplican pulsos de 90° y 180° , relajaciones de la señal un tiempo $TE/2$ y gradientes en las direcciones de fase y frecuencia para situarse en el punto deseado del espacio K. El orden seguido es el que muestra la Figura 2. Para la adquisición del espacio K, la cual se realiza de acuerdo a alguna de las dos trayectorias representadas en la Figura 3, se aplica por cada línea un gradiente de lectura en la dirección de codificación frecuencia y un gradiente *blip* en la dirección de codificación de fase. Una vez la secuencia de pulsos haya concluido se transfiere el objeto de memoria correspondiente al espacio K (ya adquirido en su totalidad) de la memoria global de la GPU a la memoria del *host*. El contenido de este objeto de memoria se corresponde con la salida retornada por el núcleo de simulación paralelo.

4. Resultados

4.1. Medidas de rendimiento

Para analizar las medidas de rendimiento del algoritmo, se realiza un experimento de reconstrucción de un corte axial de resonancia cerebral para distintos tamaños de imagen. El experimento se lleva a cabo utilizando una máquina con procesador *Intel Xeon* de 2.30 GHz con 27 núcleos e *hyperthreading* y con 64GB de memoria RAM. La GPU utilizada es una *AMD Radeon RX 480* con 8GB de memoria RAM y *device* con versión *OpenCL 1.2 AMD-APP*. Se estima el tiempo de ejecución del núcleo paralelo de simulación a partir de 1000 mediciones del mismo. Tras aplicar a los datos el test de Lilliefors y comprobar que no se cumplía la hipótesis de normalidad con un nivel de significación del 5%, se opta por presentar los resultados de forma no paramétrica. En la Tabla 1 se puede observar la mediana (Me) y el rango intercuartil (IQR) de los tiempo de ejecución empleados por cada una de las implementaciones [5].

Tamaño	64 x 64		128 x 128		256 x 256	
	Me	IQR	Me	IQR	Me	IQR
Secuencial	73.65	-	1040.30	-	16762.31	-
<i>Sparse</i>	23.80	-	413.76	-	7335.70	-
Paralelo	0.28	0.009	0.30	0.011	0.35	0.011

Tabla 1. Me e IQR del tiempo de ejecución (en segundos) para las distintas implementaciones y tamaños de imágenes.

4.2. Imágenes simuladas

Para comprobar el correcto funcionamiento del algoritmo, se llevan a cabo varios experimentos de reconstrucción de un corte axial de resonancia cerebral con tamaños de imagen de 256x256 píxeles. En un primer experimento, con el objetivo de examinar las diferencias entre *single-shot SE-EPI* y *multi-shot SE-EPI*, se varía el número de *shots* utilizados para adquirir el espacio K correspondiente a la imagen (uno y cuatro *shots*) sin la existencia de una inhomogeneidad del campo.

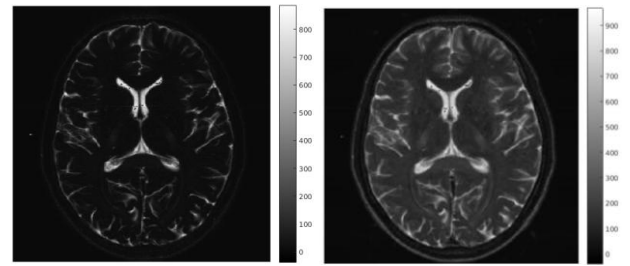


Figura 4. Reconstrucción simulada de la imagen con un *shot* (izquierda) y cuatro *shots* (derecha).

En la Figura 4 se puede observar que, como era de esperar, al aumentar el número de *shots* que se utilizan para adquirir el espacio K, el nivel de intensidad de la imagen y, por consiguiente el nivel de señal es mayor dado que la señal se atenúa menos.

En sucesivos experimentos se varían los mapas de inhomogeneidades del campo magnético en imágenes obtenidas mediante *single-shot SE-EPI*. Con el fin de facilitar la observación de los efectos de la inhomogeneidad, las imágenes han sido realizadas mediante una corrección gamma con factor 0.7.

Un mapa de inhomogeneidad de campo constante (Figura 5 a1) provoca un desplazamiento de la imagen en la dirección de codificación de fase *Y* (Figura 5 b1). Por otro lado, un mapa de inhomogeneidad del campo en forma de gradiente en la dirección de codificación de frecuencia *X* (Figura 5 a2), da lugar a un *shearing* en la imagen (Figura 5 b2), mientras que un gradiente en la dirección de codificación de fase *Y* (Figura 5 a3) da lugar a un escalado de la imagen (Figura 5 b3). Por último, en las Figuras 5 b4 y 5 b5 está representado el efecto sobre las imágenes de inhomogeneidades de campo en forma de funciones sinusoidales en las direcciones *X* e *Y* respectivamente (Figuras 5 a4 y 5 a5).

Se puede comprobar que el efecto de la inhomogeneidad de campo para el caso de una secuencia *single-shot SE-EPI*, es equivalente a aplicar una translación de los puntos del espacio imagen de acuerdo a la ecuación (3).

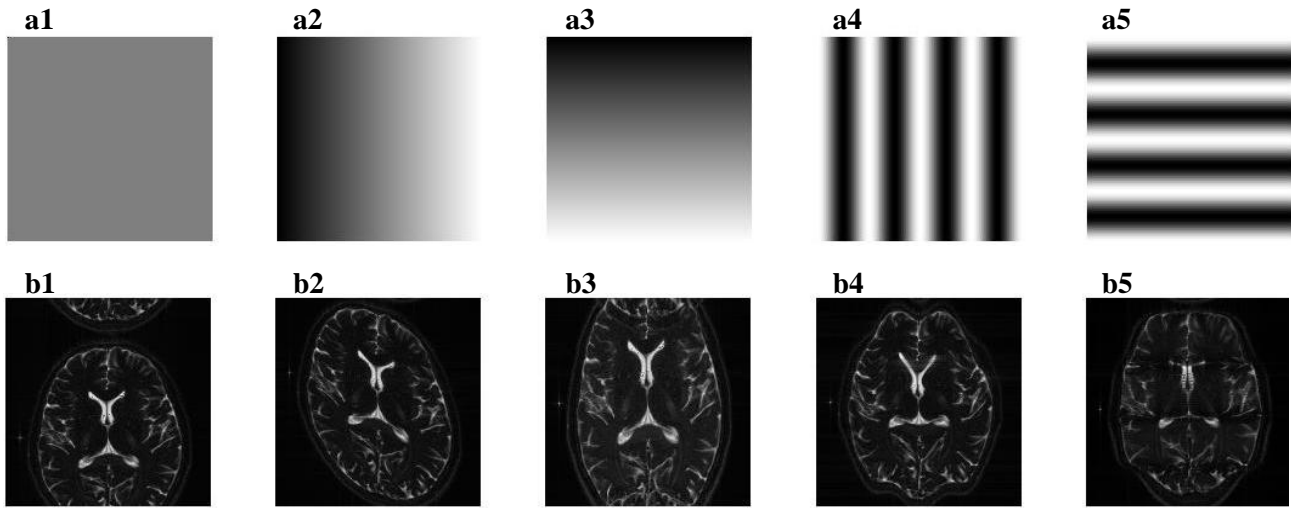


Figura 5. Parte superior, mapas de inhomogeneidad del campo simulados: constante de amplitud $5 * 10^{-6}$ T/m (a1), gradiente en X de amplitud entre $\pm 5 * 10^{-6}$ T/m (a2), gradiente en Y de amplitud entre $\pm 5 * 10^{-6}$ T/m (a3), sinusoidal en X de amplitud entre $\pm 9 * 10^{-7}$ T/m (a4), sinusoidal en Y de amplitud entre $\pm 9 * 10^{-7}$ T/m (a5). Parte inferior, reconstrucciones simuladas de las imágenes para los distintos mapas de inhomogeneidad del campo.

$$i'(x', y') = i\left(x - \frac{\Delta B(\vec{r})}{G_x}, y - \frac{\Delta B(\vec{r})(T_{acq} + \tau_g)}{G_y \tau_g}\right) \quad (3)$$

Donde $i'(x', y')$ representa la imagen considerando una inhomogeneidad del campo $\Delta B(\vec{r})$, siendo $i(x, y)$ la imagen original sin inhomogeneidad. G_x es el valor del gradiente en la dirección de codificación de frecuencia, G_y el del gradiente en la dirección de codificación de fase, T_{acq} el tiempo de adquisición de una línea del espacio K y τ_g el tiempo de aplicación del gradiente *blip*.

Nótese que todos los efectos sobre las imágenes se manifiestan, principalmente, en la dirección de codificación de fase Y, debido al menor ancho de banda en esta dirección.

5. Conclusiones y líneas futuras

La implementación aquí planteada adolece de ciertas limitaciones, tales como que no se han considerado los desfases que pudiera introducir la inhomogeneidad del campo magnético en algunos intervalos de tiempo (desde el final del pulso de 90° al inicio de los gradientes en las direcciones de codificación de frecuencia X y de fase Y, o desde el final de estos al comienzo del pulso de 180°). Asimismo, para poder obtener simulaciones más realistas sería interesante incluir fenómenos de *slew rate* en los gradientes, y considerar otros efectos que se producen en la práctica como el desplazamiento químico. Por otro lado, es importante señalar que la relajación de la señal entre las distintas líneas del espacio K que se adquieren a partir de una única excitación de RF se ha realizado a razón de T_2 en lugar de T_2^* . Efectuar una relajación a razón de T_2^* requeriría una simulación con múltiples *isochromats*.

No obstante, se ha demostrado que la implementación mediante programación en GPU con OpenCL de esta secuencia de pulsos para la simulación de imagen de resonancia magnética, permite una reducción drástica de

los tiempos de ejecución con respecto a una implementación secuencial de la misma.

Agradecimientos

Los autores agradecen al MINECO por el proyecto TEC2014-57428-R y a la Junta de Castilla y León por el proyecto VA069U16.

Referencias

- [1] Benoit-Cattin H et al., The SIMRI Project: a versatile and interactive MRI simulator. *Journal of Magnetic Resonance*, vol 173, no 1, 2005, pp 97-115.
- [2] Xanthis CG et al., Block-Based MRI System Simulator Considering Realistic Electromagnetic Fields for Calculation of Signal, Noise and Specific Absorption Rate. *Magnetic Resonance in Medicine*, vol 72, no 1, 2014, pp 237-247.
- [3] Rundle D et al., Magnetic resonance imaging simulator: A teaching tool for radiology. *Journal of digital imaging*, vol 3, no 4, 1990, pp 226-229.
- [4] Xanthis CG et al., MRISIMUL: A GPU-Based Parallel Approach to MRI Simulations. *IEEE Trans. on Medical Imaging*, vol 33, no 3, 2014, pp 607-617.
- [5] Treceño D, Calabia J, de Luis R, Alberola C. Una implementación Eficiente No Paralela de Secuencias de Resonancia Magnética mediante Matrices Sparse. Actas del XXXIV Congreso Anual de la Sociedad Española de Ingeniería Biomédica (CASEIB'16), Valencia, 2016, pp 177-180 (ISBN: 978-84-9048-531-6).
- [6] Liang ZP and Lauterbur PC. Principles of magnetic resonance imaging. A signal processing perspective. IEEE Press Series in Biomedical Engineering, 2000.
- [7] Bittoun J et al., A Computer Algorithm for the Simulation of Any Nuclear Magnetic Resonance (NMR) Imaging Method. *Magnetic Resonance Imaging*, vol. 2, no 2, 1984, pp 113-120.
- [8] Bernstein MA et al., Handbook of Pulse Sequences. Elsevier Academic Press, 2004.