

## Diseño e Implementación de una Aplicación Móvil para la Visualización de Imágenes Acústicas Pulmonares

C. G. Colin Tenorio<sup>1,\*</sup>, E. Montero Delgado<sup>1</sup>, S. Charlestón Villalobos<sup>2</sup>, T. Aljama Corrales<sup>2</sup>, F. Rodríguez Arellano<sup>3</sup>, B. A. Reyes<sup>1</sup>

<sup>1</sup> Facultad de Ciencias, Universidad Autónoma de San Luis Potosí, San Luis Potosí, S.L.P., México

<sup>2</sup> Departamento de Ingeniería Eléctrica, Universidad Autónoma Metropolitana-Iztapalapa, Ciudad de México, D.F., México

<sup>3</sup> Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación, Aguascalientes, Ags., México

\*carmen\_colin@alumnos.uaslp.edu.mx

**Resumen**—Las imágenes torácicas acústicas respiratorias (RATHI) permiten obtener información funcional de la ventilación pulmonar regional de un paciente de forma no invasiva vía el procesamiento multicanal de los sonidos respiratorios. Desafortunadamente, la disponibilidad de los sistemas para la obtención de estas imágenes es limitada en nuestro país. En este trabajo se presentan resultados preliminares del desarrollo de una aplicación móvil para el despliegue interactivo de imágenes RATHI directamente en un dispositivo inteligente bajo Android. Las imágenes RATHI son creadas mediante la transformada de Hilbert y el método de interpolación bidimensional Hermite.

**Palabras clave**—mHealth, ventilación pulmonar, imágenes acústicas.

### I. INTRODUCCIÓN

Los sonidos respiratorios (SR) son aquellos producidos por el flujo de aire a través de las vías aéreas durante la respiración, y se pueden detectar en toda la superficie torácica. Con la ayuda del estetoscopio, el médico puede conocer el estado de los pulmones a partir de los SR característicos y detectar un problema dada la alteración o ausencia de estos. Sin embargo, la auscultación pulmonar vía el estetoscopio posee varias limitaciones, e.g., no se realiza un análisis cuantitativo de la información y no se puede hacer la adquisición ni el análisis de más de un foco de auscultación simultáneamente.

Los sistemas multicanales para la adquisición y el análisis computarizado de los SR han hecho posible obtener imágenes torácicas acústicas respiratorias (RATHI, *Respiratory Acoustical THoracic Imaging*). Como alternativa a métodos más complejos tales como la centellografía, la RATHI permite estudiar de forma no invasiva la distribución de los SR y el grado de ventilación pulmonar en las diferentes regiones del tórax [1]. Lamentablemente, estos sistemas multicanales presentan limitaciones, e.g., presentan escasa movilidad, realizan el procesamiento fuera de línea y hasta el momento, su uso en nuestro país es exclusivo para la investigación. Por lo tanto, se requieren soluciones de bajo costo que permitan trasladar su aplicación a clínicas y laboratorios no especializados.

Actualmente, se cuenta con aplicaciones móviles (*apps*) para la salud (*mobileHealth, mHealth*) diseñadas para dispositivos inteligentes, e.g. *smartphones*, que son útiles para recopilar y entregar datos de salud clínica y comunitaria a profesionales de la salud, investigadores y pacientes, monitorear los signos vitales y brindar atención remota a los pacientes [2]. Cabe mencionar que la gran mayoría de *apps* comerciales para los SR son del tipo educativo, mientras que las orientadas a su análisis son desarrolladas para propósitos de investigación y se limitan al análisis monocanal [3,4]. Así, es prometedor el desarrollo de sistemas móviles para el análisis multicanal de SR.

El objetivo de este trabajo es presentar los resultados preliminares acerca del diseño e implementación de una *app* para dispositivos inteligentes, denominada mRATHI, capaz de desplegar imágenes torácicas acústicas respiratorias directamente en dispositivos Android. Para ello, se empleó el entorno de desarrollo integrado (IDE) oficial Android Studio (Google Inc., CA, USA). Cabe mencionar que, actualmente no se cuenta en el mercado ni en el ámbito de la investigación con una aplicación que permita visualizar directamente en el *smartphone* la información acerca de la ventilación regional de un sujeto obtenida de manera no invasiva mediante el procesamiento multicanal de los SR.

### II. METODOLOGÍA

#### A. Base de datos

Se utilizaron registros multicanales de SR adquiridos de pacientes con enfermedades intersticiales difusas por nuestro grupo de trabajo [1]. Cada voluntario proporcionó su consentimiento informado, de acuerdo con los protocolos de Helsinki. Cada registro consta de 25 canales de SR obtenidos de un arreglo de 5x5 micrófonos, separados 5 cm entre sí y ubicados sobre la superficie posterior del tórax. De esta forma se obtienen sonidos en ambos hemitórax desde la base hasta los ápices pulmonares, como se muestran en la Fig. 1. Se emplearon micrófonos electret subminiatura (Knowles Electronics, LLC., IL, USA) encapsulados en una campana plástica y la adquisición se realizó a una frecuencia de muestreo de 10 kHz y 12 bits-por-muestra. Se solicitó a los voluntarios respirar en un neumotacómetro Fleisch a un

flujo de aire de 1.5 l/s durante 10 s. Las señales fueron almacenadas para su posterior procesamiento en Matlab (The MathWorks, MA, USA).

### B. Mapas acústicos pulmonares

A partir de los SR adquiridos se generaron las imágenes RATHI como sigue. Primero, la ganancia de los canales fue uniformizada empleando un archivo de calibración con sonidos a campo abierto. Después, cada sonido fue filtrado vía un filtro pasabanda con respuesta al impulso finita (FIR) de orden 201 y ancho de banda de 75 a 1000 Hz para reducir interferencias del sonido cardiaco. Posteriormente, las fases inspiratorias y espiratorias fueron segmentadas automáticamente empleando la información del flujo respiratorio, para generar los mapas de inspiración y espiración. Para cada SR  $x(t)$  se calculó su envolvente  $a(t)$  mediante la magnitud al cuadrado de su señal compleja  $z(t) = x(t) + jy(t)$  obtenida mediante la transformada de Hilbert, i.e.,

$$a(t) = \sqrt{x^2(t) + y^2(t)} \quad (1)$$

donde  $y(t)$  es la transformada de Hilbert dada por

$$y(t) = \frac{1}{2\pi} PC \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau \quad (2)$$

y  $PC$  indica el valor principal de Cauchy.

El mapa RATHI se generó con una función de interpolación que mapea la envolvente de todos los canales de SR a una imagen (matriz) de acuerdo a su posición en el arreglo de micrófonos. El procedimiento de interpolación transforma la imagen original  $S(P, f)$  en otra denotada por  $SI(PI, fI)$  con el objetivo de incrementar el nivel de discretización original de  $S$ . De acuerdo con nuestro trabajo previo [1], se empleó la función de interpolación Hermite definida como:

$$\begin{aligned}
 & f^I(p_{k^I}, n) \\
 &= \frac{f(p_k, n)(p_{k^I} - p_{k+1})^2 (2(p_{k^I} - p_k) + (p_{k+1} - p_k))}{(p_{k+1} - p_k)^3} \\
 &+ \frac{f(p_{k+1}, n)(p_{k^I} - p_k)^2 (2(p_{k+1} - p_{k^I}) + (p_{k+1} - p_k))}{(p_{k+1} - p_k)^3} \quad (3) \\
 &+ \frac{f'(p_k, n)(p_{k^I} - p_k)(p_{k+1} - p_{k^I})^2}{(p_{k+1} - p_k)^2} \\
 &+ \frac{f'(p_{k+1}, n)(p_{k^I} - p_{k+1})(p_{k^I} - p_k)^2}{(p_{k+1} - p_k)^2}
 \end{aligned}$$

donde la variable  $P$  representa un conjunto finito de píxeles con coordenadas  $(p_k, p_l)$ ,  $f(P, n)$  es la función de intensidad que produce diferentes valores de paleta de colores,  $I$  representa la imagen interpolada, y  $n$  es el tiempo discreto en el que se interpolan los datos 2D [5]. La extensión a 2D se puede llevar a cabo mediante interpolación separable; i.e., la interpolación de imágenes

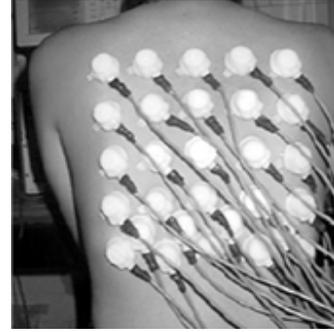


Fig. 1. Posición real del arreglo de micrófonos empleados para la adquisición multicanal de sonidos respiratorios [1].

2D se obtiene por interpolación consecutiva en 1D, primero por filas y luego por columnas. Para el despliegue de los mapas RATHI, se empleó un mapa de colores donde las áreas de mayor ventilación se representan en un color rojo intenso y las menos ventiladas con un color negro.

Cabe mencionar que, para el desarrollo y prueba de la aplicación mRATHI en el *smartphone* se seleccionaron los mapas acústicos de un sujeto voluntario generados a lo largo de una fase inspiratoria. La fase se dividió en 33 ventanas de tiempo, cada una de 50 ms de duración. Así, se generó un mapa acústico localizado al tiempo central de la ventana resultando en una secuencia de 33 imágenes RATHI interpoladas de 5x5 a 41x41 píxeles. La secuencia de imágenes fue almacenada en un archivo de texto separado por comas para su posterior apertura, procesamiento y despliegue en la *app*, la cual se detalla a continuación.

### C. Aplicación móvil

La *app* mRATHI se implementó para el sistema operativo Android vía Android Studio (versión 3.03) instalado en una computadora personal con sistema operativo Microsoft Windows 8.1 de 64 bits, procesador Intel® Core™ i5-4200U CPU @ 1.60GHz y 8GB de RAM. Aunque actualmente Kotlin (JetBrains, St.Petersburg, Rusia) es el lenguaje de programación recomendado, se decidió trabajar en Java (Sun Microsystems, Oracle Corporation, CA,USA) dada la mayor cantidad de documentación disponible actualmente y a su compatibilidad con Kotlin.

Actualmente, la *app* mRATHI desarrollada cuenta con tres actividades (*activities*): 1) *Welcome Activity*, 2) *Load Data Activity* y 3) *Main Activity*. Cada una de las actividades representa una pantalla de la interfaz gráfica de usuario (GUI). En *Welcome Activity* se despliega la pantalla de bienvenida de la *app*, en la *LoadData Activity* se lee un archivo de texto con los mapas acústicos a desplegar, mientras que la *Main Activity* permite visualizar los mapas RATHI y el flujo respiratorio asociado en forma de una animación, pudiendo iniciarse, pausarse y reiniciarse a petición del usuario.

### C.1 Apertura de datos en la app

En *LoadData Activity* se creó una nueva carpeta denominada *raw* dentro de la carpeta de recursos existentes (*res*), donde se colocaron los datos provenientes del archivo de datos con los mapas RATHI. Al leer el archivo de datos, se obtuvo una matriz de 1681 filas (41x41 pixeles) y 33 columnas (cantidad de mapas), de tal forma que cada vector columna representa una imagen RATHI, y a su vez, cada imagen es una matriz de 41x41 pixeles, como se muestra en la Fig. 2. Así, se implementó una función capaz de leer el archivo de texto con los datos de la secuencia de mapas, como de extraer y ajustar las dimensiones de estos para su posterior despliegue como una secuencia o animación en la *app*.

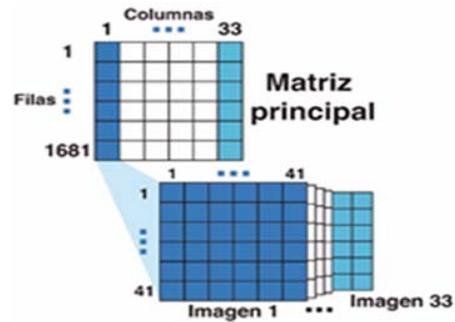


Fig. 2. Apertura de los datos contenidos en el archivo de texto. Cada columna representa un mapa de 41x41 pixeles. Para generar cada imagen RATHI en la *app*, cada columna se redimensionó a una matriz.

### C.2 Despliegue de los mapas RATHI en la app

Los mapas acústicos se desplegaron mediante el empleo de las librerías *Bitmap* y *AnimationDrawable*. Sin embargo, es importante mencionar que se debió crear una clase y métodos especiales, adecuados a la estructura de los datos, para desarrollar los gráficos. La clase y métodos desarrollados se describen a continuación:

- Clase *myAnimation*: contiene los métodos *Play*, *Pause* y *Reset*, para manipular el despliegue de la secuencia de mapas RATHI.
- Método *onCreate*: se inicializan variables (para la imagen, gráfica y botones) y se declaran las características *OnClickListener* (detección de clics) de los botones.
- Método *LoadData*: permite cargar los datos que se encuentran en el dispositivo móvil y genera las matrices RATHI.
- Método *CreateImage*.: genera un *Bitmap*, el cual representa la información de un solo mapa acústico.
- Método *CreateBitmap*: genera el arreglo tipo *Bitmap* que representa los mapas acústicos.
- Método *CreateDrawable*: convierte el arreglo tipo *Bitmap* en tipo *Drawable* necesario para la animación.

Para poder cargar datos, se llama al método *CreateBitmap* el cual a su vez manda llamar al método *CreateImage*. Este último método se encarga de llenar una matriz que proviene del método *LoadData*, transformando esa matriz en un arreglo tipo *color* que es convertida al formato tipo *Bitmap*. La clase *Bitmap* permite transformar datos numéricos en datos representados por un color. Una vez que se realiza la creación de un *Bitmap* (un mapa RATHI), se repite para todos los mapas disponibles, generando así un arreglo tipo *Bitmap*.

Para poder dar un efecto de animación se utilizó la función *AnimationDrawable*, para ello se transformó el arreglo tipo *Bitmap* en tipo *Drawable* puesto que es el tipo

requerido en el argumento de esa función. La clase convierte toda la secuencia de mapas RATHI en tipo *Drawable*, al llamar a los métodos *CreateBitmap* y *CreateDrawable*.

En los métodos de la clase se establecen las características para poder iniciar y finalizar la animación. En el método *onCreate* se crea un objeto de la clase y se crean dos botones (*play* y *reset*), el primero de ellos tipo *toggle*, los cuales al ser activados mandan llamar a los métodos de la clase correspondientes, permitiendo controlar interactivamente el despliegue de la secuencia de mapas RATHI, i.e., poder pausarlos y reiniciarlos.

### C.3. Despliegue de la señal de flujo respiratorio

La señal de flujo es importante como referencia temporal para los eventos de SR; en nuestro caso, para conocer la localización temporal de un mapa RATHI específico dentro de la fase respiratoria. Para poder visualizar la señal de flujo se empleó la librería *MP Android Chart* (PhilJay, Austria), y el tipo de gráfico *LineChart*, útil para desplegar señales como formas de onda.

## III. RESULTADOS Y DISCUSIÓN

En la Fig. 3 se muestra un ejemplo de los mapas RATHI obtenidas con el método de interpolación Hermite. La intensidad de cada imagen se normaliza con respecto al máximo de todas las imágenes generadas en la secuencia. En la Fig. 4 se muestran capturas de pantalla de la actividad de bienvenida y de la actividad principal de la *app* mRATHI desarrollada en Android Studio. En la Fig. 4.b. se detalla la actividad principal de la *app*, se observa el área de despliegue de los mapas acústicos pulmonares, el área de despliegue de la señal de flujo respiratorio asociada a la maniobra, y los botones que permiten manipular la animación, i.e., *play/pause (togglebutton)* y *reset*. Cabe mencionar que el despliegue de la imagen RATHI se encuentra sincronizado temporalmente con el despliegue de la señal de flujo, lo cual permite conocer la ubicación

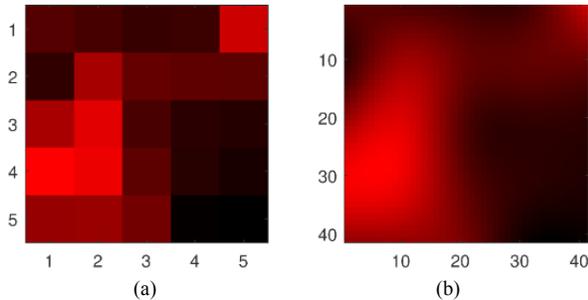


Fig. 3. Ejemplo de mapas respiratorios acústicos torácicos. a) Sin interpolar (5x5 píxeles). b) Interpolado vía Hermite (41x41 píxeles).

temporal del mapa de ventilación pulmonar con respecto a la maniobra respiratoria. Finalmente, en la Fig. 5 se muestra un ejemplo de las imágenes RATHI correspondientes a la porción central del ciclo inspiratorio desplegadas con Matlab y con la *app* mRATHI desarrollada. Observe el alto grado de similitud entre ambas imágenes.

#### IV. CONCLUSIONES

En este trabajo se presentan resultados preliminares del desarrollo de una aplicación móvil para dispositivos móviles inteligentes con sistema operativo Android que permita el procesamiento de registros multicanales de SR para generar y desplegar mapas torácicos acústicos respiratorios. Por el momento, la *app* carga una secuencia de mapas RATHI interpolados en Matlab, y los despliega adecuadamente en el dispositivo móvil de forma sincronizada con el flujo respiratorio asociado que resulta fácil de interpretar por los usuarios. El despliegue de los mapas y el flujo es interactivo, permitiendo pausar, continuar o reiniciar la secuencia de imágenes RATHI. Cabe mencionar que, aunque el despliegue de una secuencia de matrices, en este caso los mapas RATHI, pareciera una tarea trivial, existe escasa documentación al respecto para Android Studio o incluso los métodos requeridos no han sido implementados. Aunado a esto, la constante actualización del IDE y del lenguaje computacional dificulta la búsqueda de soluciones a problemas particulares asociados a la *app* para el procesamiento y despliegue computarizado de SR.

Actualmente, se está implementando la generación de los mapas RATHI y la interpolación Hermite en Android Studio para procesar los datos crudos de un registro multicanal de SR directamente en la *app*. A la par, se trabaja en el sistema de adquisición de los SR para contener con limitantes del sistema actual, e.g. la limitada movilidad del usuario, mediante el uso de una red inalámbrica de sensores corporales. Concluido el sistema, se procederá a su comparación con técnicas estándar como la centellografía.

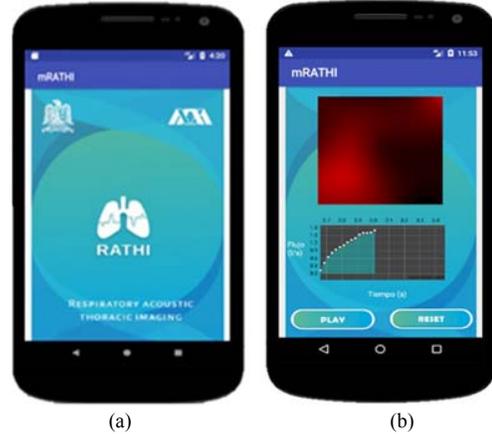


Fig. 4. Actividades de la *app* mRATHI desarrollada para Android. a) *Welcome Activity*. b) *Main Activity*. Esta última se compone por el área del mapa acústico (gráfico superior), el área de la señal de flujo (gráfico medio) y botones de manipulación de la animación (parte inferior).

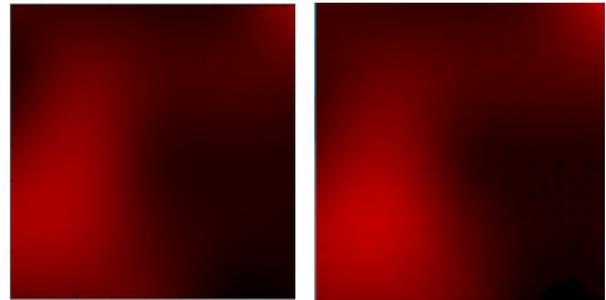


Fig. 5. Imágenes RATHI desplegadas en Matlab (izquierda) y su contraparte desplegada en la *app* desarrollada en Android (derecha).

#### RECONOCIMIENTOS

Este trabajo fue realizado parcialmente con el apoyo del Programa para el Desarrollo Profesional Docente (PRODEP) de la Secretaría de Educación Pública (SEP).

#### BIBLIOGRAFÍA

- [1] S. Charleston-Villalobos, S. Cortés-Rubiano, R. González-Camerena, G. Chi-Lem, y T. Aljama-Corralles, Respiratory acoustic thoracic imaging (RATHI): Assessing deterministic interpolation techniques, *Med. Biol. Eng. Comput.* 42(5): 618-626, 2004.
- [2] P. Germanakos, C. Mourlas, y G. Samaras, A Mobile Agent Approach for Ubiquitous and Personalized eHealth Information Systems, *Proceeding of the Workshop on Personalization for e-Health o the 10<sup>th</sup> International Conference on User Modeling (UM'05):* 67-70, 2005.
- [3] N. C. Olvera-Montes, B. A. Reyes, S. Charleston-Villalobos, R. González-Camerena, A. T. Aljama-Corralles, y M. E. Mejía-Ávila, Sistema para la Detección Automática de Estertores Crepitantes basado en Dispositivos Móviles Inteligentes, *Mem. Congr.Nac.Ing.Bioméd.(4)*: 320-323, 2017.
- [4] B. A. Reyes, N. Reljin, Y. Kong, Y. Nam, S. Ha, and K. H. Chon, Towards the Development of a Mobile Phonopneumogram: Automatic Breath-Phase Classification Using Smartphones, *Ann. Biomed. Eng.* (44)9: 2746-2759, 2016.
- [5] G. J. Grevera, J. K. Udupa, y Y. Miki, A task-specific evaluation of three-dimensional image interpolation techniques, *IEEE Trans. Med. Imaging*(18)2: 137-143, 1999.