

Towards a Smart Phone-Based Cardiac Monitoring Device using Electrical Impedance Tomography

Saaid H. Arshad, Jordan S. Kunzika, Ethan K. Murphy, Kofi Odame, Ryan J. Halter

Thayer School of Engineering, Dartmouth College

14 Engineering Drive, Hanover, NH 03755

Email: saaid.h.arshad.TH, jordan.s.kunzika.16, ethan.k.murphy, kofi.m.odame, ryan.j.halter@dartmouth.edu

Abstract—A novel framework for cardiac monitoring is presented that leverages a custom smart phone application and a wearable electrical impedance tomography (EIT) system. The smart phone application required to implement this framework has been designed and is described here. This technology could greatly improve telemonitoring of patients with chronic heart failure. A simulated portable EIT device was constructed that wirelessly sends data to the smart phone. The developed application receives wireless data transmission, performs EIT image reconstruction, and extracts a post-processing measure correlated to cardiac output. Data transmission rates of 4.2 MB/s were achieved between a simulated EIT system (server) and phone (client). Image reconstruction and post-processing implemented on a smart phone requires 332 ms, moderately longer than the 109 ms required on a modern laptop based implementation. The 332 ms is well within the approximately one second processing time needed to measure beat-to-beat cardiac output (i.e. assuming a 60 bpm heart rate).

Keywords—EIT, cardiac, cardiovascular, telemonitoring, smartphone, wearable technologies

I. INTRODUCTION

Telemonitoring is becoming increasingly important as the proportion of the population living with cardiovascular disease (CVD) is growing [1, 2, 3]. Currently used health parameters lack the sensitivity and specificity to accurately predict heart failure events, and thus force physicians to play a reactive versus proactive role in patient care [2].

This study presents initial development of a smart-phone application (app) that reads in data from a simulated electrical impedance tomography (EIT) system, performs EIT image reconstruction, and computes a post-processing measure related to cardiac output. This app is part of a proposed cardiac monitoring system that can potentially improve the technical capability of telemonitoring by adding cardiac hemodynamic measurements to the suite of current telemonitoring tools. This will improve clinical practice by providing timely information specific to heart function. The device will be easy-to-use for the patient and will be capable of performing quick and efficient EIT reconstructions on a smartphone so that cardiac output metrics can be tracked by the patient and clinician.

This paper begins by first describing EIT, the simulated EIT system and simulated scenario, and then details the smart-phone application. This smart phone app is then analyzed, performance results are provided and discussed, and future work is presented.

II. USING EIT FOR CARDIAC IMAGING

A. Electrical Impedance Tomography

Electrical impedance tomography (EIT) is a safe, inexpensive, and non-invasive imaging modality that can provide an image of a conductivity distribution within a domain of interest; this distribution is inferred from electrode-based current and voltage measurements recorded on the surface of the domain. Small alternating currents are injected through pairs of electrodes and the resulting voltages on the other electrodes are measured. From these currents and voltages, an estimate for the conductivity distribution of the object or changes in the conductivity within the object can be obtained [4]. Image reconstruction is formulated as a non-linear least-squares model-fitting problem, where a 2D or 3D Finite Element Method (FEM) approach is used to model the imaging region and to predict the measured potentials.

For this device, we are interested in evaluating changes in heart conductivity (associated with different ventricular blood volumes) at different phases of the cardiac cycle. This involves observing the *change* in impedance over time. In this case, a reconstruction algorithm solves the inverse problem using a reference data set (i.e. boundary voltages) acquired at some time and a subsequent data set recorded at some later time. The change in conductivity at each of the FEM mesh nodes, $\Delta\sigma$, is given as:

$$\Delta\sigma = (J^T J + \lambda L^T L)^{-1} J^T \{\Phi_{ref} - \Phi_{frame}\} \quad (1)$$

where J is the Jacobian, a sensitivity matrix dependent on the FEM mesh geometry, λ is the regularization parameter, L the regularization matrix, Φ_{ref} is a set of boundary voltages recorded from a reference conductivity distribution, and Φ_{frame} is a set of boundary voltages collected from a different conductivity distribution [5]. This equation can be simplified to the form:

$$\Delta\sigma = A * V \quad (2)$$

a simple matrix multiplication where $A = (J^T J + \lambda L^T L)^{-1} J^T$, V is the change in input boundary voltages ($\Phi_{ref} - \Phi_{frame}$), and $\Delta\sigma$ is the change in conductivity ($\sigma_{ref} - \sigma_{frame}$) within the domain that we want to reconstruct. Equation (2) is one of two computations required within in the smart-phone implementation.

The other computation required is an area calculation. An area of interest within the conductivity mapping can be specified to include the heart. Tracking the conductivity change within that area of interest during different phases of the cardiac cycle can yield a surrogate measure for cardiac output (i.e. the conductivity difference between end systole and end diastole). It has been demonstrated that the EIT-based conductivity changes associated with an expanding and contracting heart volume is directly correlated to cardiac output [6, 7]; this motivates our development of a wearable EIT-based cardiac output monitor.

Fig. 1 details the framework of the EIT system. A wearable EIT, electrocardiogram (ECG), and respiratory impedance plethysmography (RIP) system are proposed to record the appropriate physiological signals and wirelessly relay them to the patient's smart phone where multi-threading allows parallel handling of the following processes: Thread 1 manages the receiving and storage of data both before and after processing, thread 2 analyzes ECG data, thread 3 analyzes RIP data, thread 4 performs the EIT reconstruction and area calculation, thread 5 uses the processed data to gauge cardiac output, and thread 6 runs a graphic-user-interface (GUI) and provides a movie play-back option for the user. The following sections discuss the aspects of the framework that have been either simulated or implemented.

III. SIMULATED SCENARIO AND EIT SYSTEM

A. Simulated Scenario: 4D XCAT Model

The 4D XCAT model is an anatomically accurate model of the thorax with highly detailed static and dynamic descriptions of the intrathoracic anatomy [8,9]. Multiple 3D voxel-based images depicting organ and tissue movement during respiration and heart beating are extracted from the model. Dense 3D FEM meshes (267,502 nodes, 1.4 million tetrahedral elements) are constructed from these models so that simulated boundary voltages (Φ_{ref} & Φ_{frame}) can be computed and used to evaluate our smart phone implementation of EIT image reconstruction [9]. Conductivities assigned to the heart, blood, lungs, and boney structures of the model were 0.34 S/m, 0.66 S/m, 0.09 S/m, and 0.04 S/m, respectively, assuming a 20 kHz input current [10].

B. Simulated EIT System

The EIT system is assumed to consist of a belt of 16 electrodes equiangular spaced around the circumference of the thorax. Tetrapolar impedance measurements are used in which current is applied between two electrodes (II) and the induced voltage is measured across two other electrodes (VV); the set of four electrodes used for applying current and sensing voltage specifies an IIVV pattern. Electromagnetic simulations using custom software previously validated [11] were used to generate a set of simulated boundary voltages on the meshes derived from the 4D XCAT models assuming a complete set of IIVV patterns (all tetrapolar combinations of the 16 electrodes). Specifically, boundary voltages for 16 frames consisting of a single cardiac cycle were simulated assuming a breath hold.

IV. SMART-PHONE IMPLEMENTATION

The framework components that have been implemented are highlighted in red boxes in Fig. 1. These include Thread 1 and the two computational components of Thread 4. These have all been implemented on a Motorola Nexus 6 phone with 64 GB internal memory, 3 GB RAM, and 2.7 GHz CPU [12].

A. EIT Reconstruction Storage and Processing Specifications

EIT reconstruction requires a set of boundary voltages, a Jacobian, and a regularization matrix to solve (2). A 16-electrode configuration yields a total of 10,920 unique IIVV patterns each corresponding to a single boundary voltage measurement. A full vector of boundary voltages used for reconstruction consists of 10,920 double precision values (~87.4 kB) where each vector represents Φ_{ref} or Φ_{frame} .

Due to the ill-posedness of (1), the dense 3D mesh of 267,502 nodes is mapped to a coarse mesh of 302 nodes over which conductivity is estimated. The Jacobian, J in (1), is a 25.2 MB 10,920×302 matrix and the regularization term, $\lambda L^T L$ in (1), is a 58 kB 302×302 matrix. The augmented Jacobian that is calculated is also a 25.16 MB 302×10,920 matrix stored for use for every single matrix multiplication. The reconstructed conductivities, which are vectors of length 302, and the corresponding scalar area calculations require 2.36 kB and 8 bytes, respectively. In total, ~26 MB are required to store the augmented matrix, A , in (2). In addition, 2.36 KB and 8 bytes are needed for the reconstructed conductivities and area calculations, respectively, for every received vector.

For real-time EIT extraction of cardiac output, data needs to be acquired at least at an average patient's heart-rate, which at rest is one beat per second [8].

B. Libraries Used and Server-Client Setup

Libraries are needed to efficiently implement linear algebraic operations; the app uses the JMatIO library (<https://github.com/sinjax/JMatIO>) and the Java Matrix Package (<http://math.nist.gov/javanumerics/jama/>) for this purpose. JMatIO provides Java-based functionality to read, write, and manipulate data from .mat files created using Matlab. The Java Matrix Package was used for vector and matrix manipulation and computation..

A MATLAB-based server was used to simulate the portable wireless sensor system of Fig. 1, transmitting simulated boundary voltage data from the server to a smart phone. This was implemented on a Lenovo laptop with a 64-bit operating system running Windows 7 on an Intel Core i5-4300U processor running at 2.50 GHz with 12 GB RAM.

The simulated test is setup using a server-client relationship between the simulated EIT system (server) and the Android application (client). The server and client interact using a TCP socket connection created over WiFi. The server opens a TCP socket as a host using the computer's IP address and waits for a client to connect to that socket. The Android application performs preliminary steps and sets up a Java based TCP client socket protocol and connects to the server. Once the connection is created, the server simulates real-time data transfer by looping through the 16 vectors (one representing

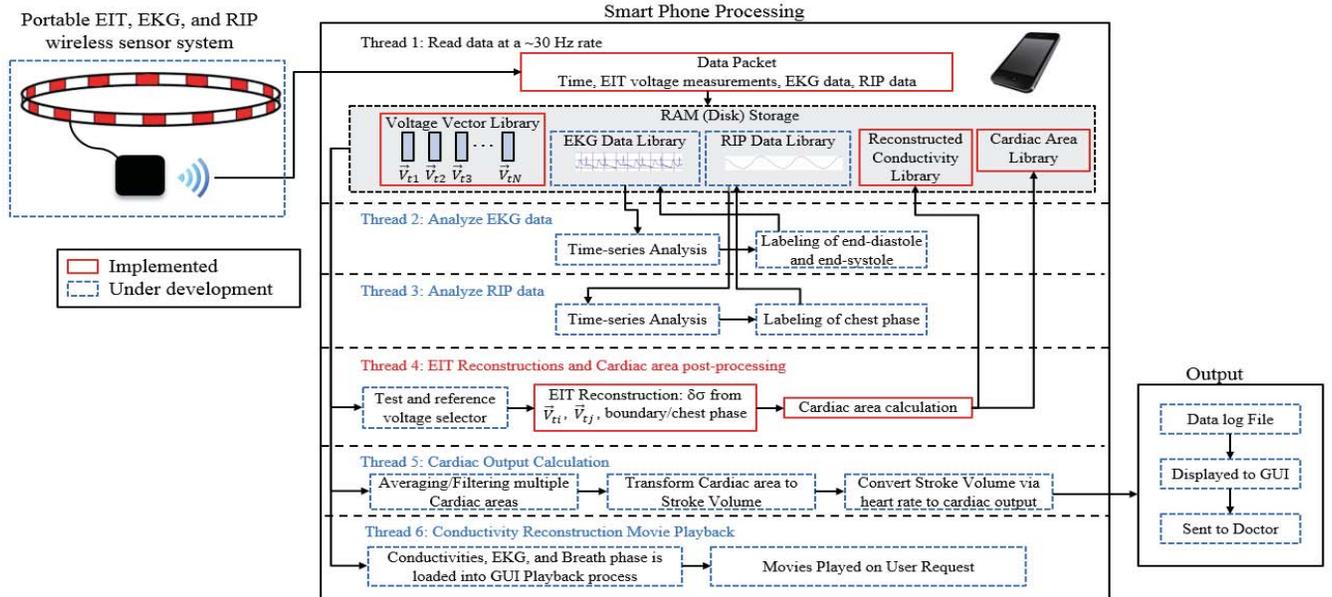


Fig. 1. This flow-chart details the expected configuration and structure of the cardiac EIT device and framework. The red boxes indicate components that have been implemented.

each changing heart frame), sending the data to the Android application via the socket, one vector at a time.

C. Android Application:

The Android application first reads in static matrices from .mat files, which are loaded in as application assets, and performs a series of matrix multiplications to compute $J^T J + \lambda L^T L$ in (1). This is then inverted and multiplied by J^T to obtain A in (2), and ultimately multiplied together with the incoming EIT data from the server. The Android client has a multithreading process that handles the incoming data (see Fig. 1). It uses a single asynchronous task as a background thread (Thread 1) for receiving data as byte arrays and converts those arrays into arrays of doubles. After performing this initial processing, for each vector that the server sends, a separate thread (Thread 4) is spawned that handles matrix multiplication and area calculations that need to be made for each vector, using thread locks to ensure that each vector is processed sequentially and separately. Thread locks are also used to ensure that no two vectors are processing simultaneously.

V. RESULTS

Performance data regarding timing and storage is presented in tables 1 and 2. Table 1 compares run-times for the inversion calculation and area calculation between the smart-phone implementation and the identical calculations being run on the same laptop running the server.

The initialization of the app includes the loading of the matrices and the calculation of A in (2) before inversion, which take on average 14.3 seconds and 11.92 seconds, respectively. The smartphone expectedly takes longer due to having less processing power. The inversion is the most computationally expensive step, taking 723.6 ms on the smartphone and 5.6 ms on the computer, however, this is only done once and can be included in the initialization time. The area calculation is the second most expensive step, taking 330 ms and 108.2 ms on the phone and computer, respectively.

Finally, the matrix multiplication is the fastest step, taking 1.6 ms and 0.6 ms on the phone and computer, respectively.

Table 2 shows the data reception and processing rates for the smart phone implementation. The significant differences between reception rate and processing rate in Table 2 will use up the available RAM and lead to unprocessed voltage vectors. The 3 GB RAM specified on this phone is limited to ~2.33 GB available RAM due to other passive processes running in the background.

Fig. 2 shows a sample reconstruction resulting from the inversion step performed on the phone. The image shown is a conductivity difference image computed with voltage data recorded from two different heart frames – one in which the heart is at its largest possible volume (at end-diastole), and a second frame in which the heart is in its smallest possible size (at end-systole). Heart-size is varied over the 16 frames with respect to an end-systole heart, so as the heart size changes from the small heart, the area reconstructed grows, and as the two hearts become closer in size, the area decreases. The area calculation step computes the area of the bright region (sum of all pixels with a conductivity difference exceeding 50% of the maximum conductivity change), representing the conductivity difference between the two different heart frames. Calculated areas are correlated to the “true” heart area extracted from the XCAT model with a coefficient of 0.887, as shown in Fig. 3, confirming the findings of [6] and [7].

Table 1: Run Times for Key Operations in EIT Reconstruction

Function	Android Time (ms)	Computer Time (ms)
Inversion to get A	723.6	5.6
Matrix Multiplication	1.6	0.6
Area Calculation	330	108.2

Table 2: Data Processing Rates

Android Implementation	Vectors/Second	Kilobytes/second
Data Reception Rate	53	4240
Total Data Processing Rate ^a	3	240

^aTotal Processing Rate consists of time for matrix multiplication and area calculation

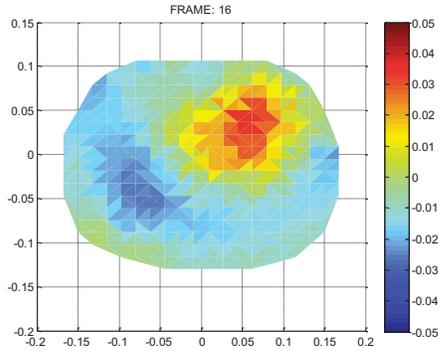


Fig. 2. An axial slice of a sample conductivity difference image with 302 nodes. This slice is at the center of the mesh, in the plane of the electrodes. The high change in conductivity region results from the change heart volume.

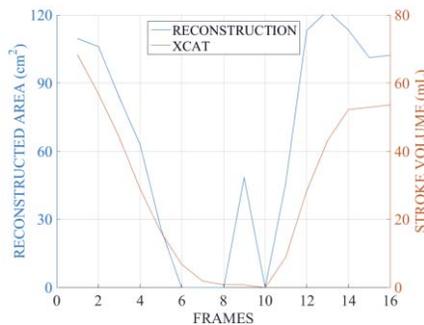


Figure 3: The red line shows the heart area difference at different frames. The blue line shows the reconstructed area. The correlation coefficient is 0.887.

VI. DISCUSSION

The smartphone implementation performs well within the required specifications for accurately measuring cardiac output and has processing speeds comparable but obviously slower than that of a desktop computer.

Each individual thread that is spawned for processing a particular vector takes on average 331.6 ms to successfully process the data. The asynchronous task that receives data is handled in a thread that is separate from the individual processing threads; the application can receive 16 vectors from the server in (on average) 300 milliseconds, which means that approximately 40-50 data vectors can be received per second, while the majority of the time would be spent waiting for each vector to be processed one by one afterwards at a speed of ~ 3 vector per second (see Table 2). This means that the system can receive and buffer the data in real-time and processing time would allow the system to update cardiac output to the user to at ~ 3 updates per second.

Since it is important to know how cardiac output evolves over minutes, hours, days, and weeks, it may be more important to provide a means to reconstruct images and compute cardiac output during times when data is not being recorded. This limited dataset (cardiac output as a single parameter at a given time as opposed to the vector at that time) would significantly limit the quantity of data that would need

to be transmitted via WiFi or other wireless communication protocol to the user or to a clinician reviewing the data, and therefore reduce power consumption associated with data transfer. While over 30 MB of data are used to compute cardiac output, cardiac output itself is an 8 byte scalar and transferring it would be trivial.

VII. CONCLUSIONS AND FUTURE WORK

The smart-phone implementation of an EIT reconstruction is a major-step towards completing this EIT-based cardiac imaging framework. Future steps regarding the smart-phone implementation in the short-term include 1) exploring the use of virtual memory to expand RAM capacity, and 2) implementing threads 2, 3, 5, 6, and the first block in thread 4 (Fig. 1). Current work apart from the smart-phone EIT reconstruction implementation includes determining how to account for inaccuracies in reconstruction introduced by breathing (both size change in the chest and lungs confound the desired reconstructed heart conductivities). Power consumption associated with data processing and data transmission are important design parameters we are considering as we develop the remaining elements of this wearable cardiac monitoring device.

ACKNOWLEDGMENTS

This project was made possible by Funding Source: NSF (SCH-INT) IIS-1418497.

REFERENCES

- [1] R. Purcell, S. McInnes, and E.J. Halcomb, "Telemonitoring can assist in managing cardiovascular disease in primary care: a systematic review of systematic reviews," *BMC Family Practice*, vol. 15, no. 1, pp. 43, 2014.
- [2] H. Ayesha and V. Paul, "Telemonitoring in chronic heart failure," *European heart journal*, vol. 32, no. 12, pp. 1457-1464, 2011.
- [3] A. Chaudhry, et al., "Telemonitoring in patients with heart failure," *The New England Journal of Medicine*, vol. 363, pp. 2301-2309, 2010.
- [4] M. Vauhkonen, "Electric resistance tomography and prior information [PhD thesis]." *Kuopio University, Finland*, 1997.
- [5] R. Halter, A. Hartov, and K. Paulsen, "Video rate electrical impedance tomography of vascular changes: preclinical development," *Physiological measurement*, vol. 29, no.3, pp. 349-364, 2008.
- [6] A. Vonk-Noordegraaf, et al., "Determination of stroke volume by means of electrical impedance tomography," *Physiological measurement*, vol. 21, no. 2, pp. 285-293, 2000.
- [7] R. Pikkemaat, et al., "Recent advances in and limitations of cardiac output monitoring by means of electrical impedance tomography," *Anesthesia & Analgesia*, vol. 119, no. 1, pp. 76-83, 2014.
- [8] W.P. Segars, G. Sturgeon, S. Mendonca, J. Grimes, and B.M.W. Tsui, "4D XCAT phantom for multimodality imaging research," *Medical Physics*, vol. 37, no. 9, pp. 4902-4915, 2010.
- [9] E.K. Murphy, R.J. Halter, and K. Odame, "Simulation study for the design of an EIT system for cardiac output monitoring," *41st Annual IEEE NEBC Conf. Proc.*, April, 2015.
- [10] S. Gabriel, R.W. Lau, and C. Gabriel, "The dielectric properties of biological tissues: II. measurements in the frequency range 10 Hz to 20 GHz," *Phys. Med. Biol.*, vol. 41, no. 11, pp. 2251-2269, 1996
- [11] A. Borsic, A. Hartov, K.D. Paulsen, P. Manwaring, "3D electric impedance tomography reconstruction on multi-core computing platforms," *Conference proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008.
- [12] http://www.gsmarena.com/motorola_nexus_6-6604.php