DISSERTATION

RECOVERY OF ORGAN BOUNDARIES IN ELECTRICAL IMPEDANCE TOMOGRAPHY IMAGES USING A PRIORI DATA, OPTIMIZATION, AND DEEP LEARNING

Submitted by

Michael Capps

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2019

Doctoral Committee:

Advisor: Jennifer Mueller

Margaret Cheney
Olivier Pinaud
Randy Bartels

ABSTRACT

RECOVERY OF ORGAN BOUNDARIES IN ELECTRICAL IMPEDANCE TOMOGRAPHY IMAGES USING A PRIORI DATA, OPTIMIZATION, AND DEEP LEARNING

In this thesis we explore electrical impedance tomography (EIT) and new aspects of the solutions to the inverse conductivity problem. Specifically we will focus on new methods for obtaining additional information from direct reconstructions on 2D domains using the D-bar method based on work by Nachmann in 1996 and Mueller and Siltanen in 2000. We cover the history of EIT as well as performing a review of relevant literature. Original work presented covers (1) an application of signal separation of cardiac and ventilation signals to the recovery of pulmonary measures and detection of air trapping in children with cystic fibrosis, (2) recovery of the boundaries of internal structures in EIT data sets using optimization of a priori data in the D-bar method, (3) recovery of the boundaries of internal structures in EIT data sets using deep neural networks applied to the scattering transform in the D-bar method. Results using both numerically simulated data and data collected on a tank with simulated organs made of agar are presented.

ACKNOWLEDGEMENTS

I would like to thank my advisor Jennifer Mueller for her patience and guidance, and thanks to everyone that helps make the Electrical Impedance Tomography Laboratory at Colorado State University such a special place.

Special thanks to Dr Rusinko and Dr Fortner-Wood for helping me find my way to graduate school.

DEDICATION

For Dad.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 ELECTRICAL IMPEDANCE TOMOGRAPHY

Electrical impedance tomography (EIT) is an imaging technique used to reconstruct the internal electrical properties such as conductivity and permitivity of an object using current and voltage information from its boundary. Current is applied to the boundary via electrodes, and the induced voltages are measured and used as the data to solve the underlying inverse conductivity problem. Since different internal conductivities change the induced voltages, this data can be used to recover the conductivities. When used to image a human chest cavity, for example, we recover information about the the structure of organs with different conductivities. Additionally, if measurements are taken as a time series we can recover information about how conductivities change over time. This can allow the extraction of data relating to respiration and blood flow between the heart and lungs. In practice, currents and voltages are measured through electrodes on the boundary, and special considerations must be taken when engineering a system for taking EIT measurements due to the sensitivity of the inverse conductivity problem to measurement and modeling errors.

EIT has several advantages. First, it does not involve any doses of radiation (i.e. it is non-ionizing). So unlike x-ray based imaging methods, such as CT scans, there is no additional risk due to microdoses of radiation and there is no yearly limit to the number of times a person may be imaged using EIT. EIT is also relatively inexpensive. An MRI machine may cost over one million dollars, so hospitals may be financially limited in the number of machines they might buy, but an EIT machine could cost less than $\$50000$. Lastly, EIT machines can be made small enough to be portable. This can be very convenient for patients in an intensive care unit, since moving them to an MRI machine or CT scanner may pose deadly risks.

Although EIT has advantages over other imaging techniques, it is mathematically very difficult since the inverse conductivity problem is both nonlinear and ill-posed (the conductivities do not

depend continuously on the data; see Alessandrini's example in the Mathematical Formulation section).

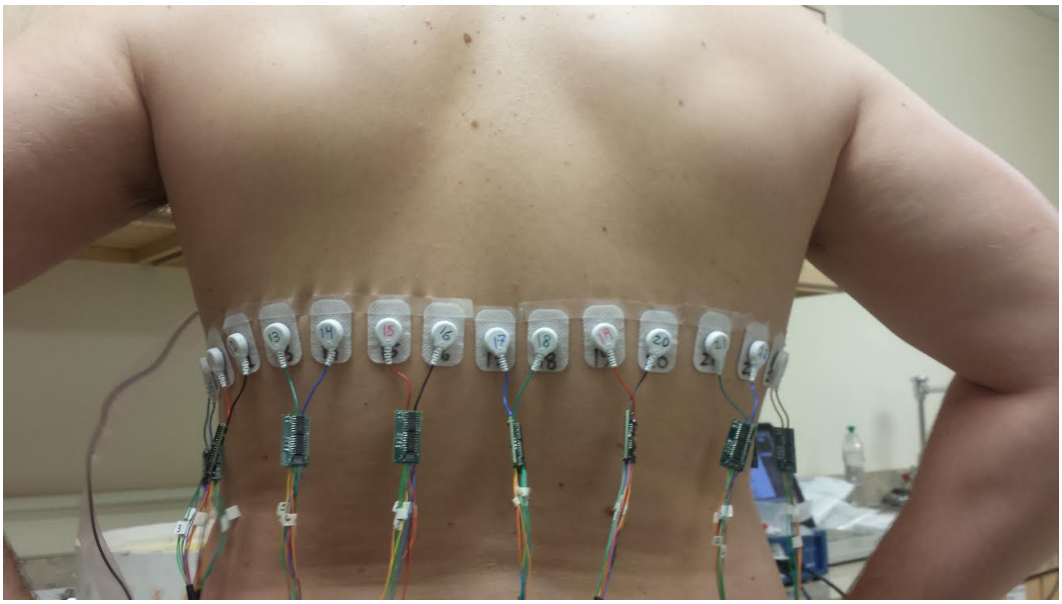


Figure 1.1: Data collection of 2D thoracic images

1.1.1 APPLICATIONS OF EIT

Since EIT is relatively inexpensive, non-ionizing, and nondestructive, it is an attractive option for many industrial and medical applications. For example, EIT can be used to monitor subsurface flows, detect subsurface objects whose conductivity differs from the background, and image pipes and concrete surfaces for defects [26] [29] [105] [101] [67] [44]. However, the focus of this paper will be on medical applications.

EIT is a good candidate for medical imaging due to the significant differences in electrical conductivity of different organs and tissues in the human body (see Table 1). In addition to being inexpensive when compared to CT and MRI, it does not use ionizing radiation so there is no increased risk of cancer to the subjects being imaged [82].

EIT is especially well suited for ventilation and perfusion imaging of human chest cavities: since air is non-conductive and blood is highly conductive, EIT can be used to image the flow

Table 1.1: Conductivity values of various human tissues and organs at 100kHz, based on values reported in [4].

Tissue or Organ	Approximate Conductivity (S/m)
Blood	0.7
Bone (Cancellous)	0.084
Bone (Cortical)	0.021
Bone Marrow	0.0033
Cardiac Muscle (transversal)	0.23
Cardiac Muscle (longitudinal)	0.63
Cartilage	0.18
Fat	0.024
Kidney	0.17
Liver	0.08
Muscle	0.36
Skin (Dry)	0.00045
Skin (Wet)	0.066
Lung (Deflated)	0.27161
Lung (Inflated)	0.10735

of blood between the heart and lungs. These images can in turn be used to assist a physician diagnosing of pulmonary problems such as pulmonary embolisms or build up of fluid in the lungs. Unlike CT scans, PET scans, and blood tests, EIT can be used for continuous monitoring and provide data over a long period of time instead of just a snapshot of the patient's condition at one time.

1.1.2 THE INVERSE CONDUCTIVITY PROBLEM

The inverse conductivity problem was first introduced by Calderón in 1980. The two-dimensional real case is as follows: given a bounded region $\Omega \subset \mathbb{R}^2$, the electric potential $u(x)$ and conductivities $\sigma(x)$ satisfy the generalized Laplace equation

$$\begin{aligned} \nabla \cdot (\sigma(x)\nabla u(x)) &= 0,\ x \in \Omega \\ u|_{\partial\Omega} &= f \end{aligned} \tag{1.1}$$

where $u|_{\partial\Omega} = f$ are the known or measured voltages on the boundary, known as the Dirichlet boundary condition.

The goal is to recover conductivities σ from knowledge of the *Dirichlet-to-Neumann map* that sends voltage distributions on the boundary to current-density distributions. This equation can be derived from Maxwell's equations by approximating the magnetic permeability of the human body $\mu = 0$ and linearizing equations for electric field E and magnetic flux B about $\mu \approx 0$.

1.1.3 THE DIRICHLET-TO-NEUMANN MAP

Our data for the inverse problem is the *Dirichlet-to-Neumann map (DN map)* and thus it is central to the methods presented here. We apply the voltage distribution f on $\partial\Omega$ corresponding to the Dirichlet boundary condition and measure the resulting current density distribution J on $\partial\Omega$:

$$J(z) = \sigma \frac{\partial u}{\partial \nu}(z), \quad z \in \partial\Omega,$$

where ν is the outward unit normal on $\partial\Omega$.

So the DN map

$$\Lambda_\sigma : u|_{\partial\Omega} \to \sigma \frac{\partial u}{\partial \nu}|_{\partial\Omega} \tag{1.2}$$

depends on σ and sends voltages to current densities on the boundary.

We assume that $\sigma \in L^\infty(\Omega)$ is strictly positive in Ω since conductivities in human tissues are positive and finite. Then from standard theory of elliptic PDEs we see that Λ_σ is a bounded linear operator between Sobolev spaces [42]:

$$\Lambda_\sigma : H^{\frac{1}{2}}(\partial\Omega) \to H^{-\frac{1}{2}}(\partial\Omega)$$

Knowledge of the DN map is equivalent to knowledge of the Neumann-to-Dirichlet (ND) map $\mathcal{R}_\sigma : \widetilde{H}^{-\frac{1}{2}}(\partial\Omega) \to \widetilde{H}^{\frac{1}{2}}(\partial\Omega)$ that sends current densities to voltages on the boundary (noting that $\widetilde{H}^s$ indicates H^s functions with mean value zero). In practice current densities are applied on the

boundary and the resulting voltages are measured because the DN map amplifies noise while the ND map dampens it.

We may also write the DN map in its weak form via integration by parts:

$$\langle \Lambda_\sigma f, g \rangle = \int_\Omega \sigma \nabla u \cdot \nabla v dx$$

where v is any $H^1(\Omega)$ function with trace g on the boundary and u is the unique $H^1(\Omega)$ solution of the Dirichlet problem above. Although the DN map itself is a linear operator, since u depends on σ we see that Λ_σ is a nonlinear function of σ due to the product of σ and ∇u in the weak formulation. This means the dependence of the data on the conductivity is nonlinear, and as a result the inverse problem itself is nonlinear.

We can compute the DN map from the Neumann-to-Dirichlet map (ND), so applying current densities instead of voltages doesn't cause any additional problems (see [92] for details on deriving the DN map from the ND map).

1.1.4 ILL-POSEDNESS AND ALESSANDRINI'S EXAMPLE

The EIT problem is a very difficult problem, mathematically and practically, for several reasons. In order for an inverse problem to be well-posed, it needs to meet Hadamard's three conditions:

1. Existence: there should be at least one solution
2. Uniqueness: There should be at most one solution
3. Stability: The solution must depend continuously on the data

Much effort has been put into proving (1) and (2) for various types of solutions, but because we can find conductivity distributions with large differences but arbitrarily close boundary measurement data, (3) is violated. This leads to the problem that even though unique solutions exist, small changes in the data due to noise, measurement error, or modeling errors can have large effects on the solution. This presents challenges for building practical systems as well since extra

precautions must be taken to reduce the impact of environmental noise and measurement errors. We use Alessandrini's example [2] to show that condition (3) is volated.

Consider two conductivity distributions σ_1 and σ_2 on the unit disc Ω,

$$\sigma_1(r,\theta) = 1, \qquad \sigma_2 = \begin{cases} 1+A, & 0 \le r \le R \\ 1, & R < r \le 1 \end{cases}$$

satisfying the generalized Laplace equation with the same Dirichlet boundary condition

$$\begin{aligned} \nabla \cdot (\sigma_1 \nabla u_1) &= 0 \quad \text{in } \Omega & \qquad \nabla \cdot (\sigma_2 \nabla u_2) &= 0 \quad \text{in } \Omega \\ u_1|_{\partial\Omega} &= \phi & \qquad u_2|_{\partial\Omega} &= \phi \end{aligned} \tag{1.3}$$

With some work we can show $\|\Lambda_{\sigma_1} - \Lambda_{\sigma_2}\|_{H^{1/2}(\partial\Omega)\to H^{-1/2}(\partial\Omega)} \le AR$, which can by made arbitrarily small (for any fixed A) by reducing R. But $\|\sigma_1 - \sigma_2\|_{L^\infty(\Omega)} = A$. So for any choice of ϵ and δ, we can find σ_1 and σ_2 such that

$$\|\Lambda_{\sigma_1} - \Lambda_{\sigma_2}\|_{H^{1/2}(\partial\Omega)\to H^{-1/2}(\partial\Omega)} < \delta$$

but

$$\|\sigma_1 - \sigma_2\|_{L^\infty(\Omega)} > \epsilon$$

by choosing A and R appropriately. This shows that the data can be very close together but come from very different conductivities. See [2, 92] for the full details of this example.

1.1.5 CURRENT PATTERNS

Since it is impossible to apply a continuous current pattern on the boundary we are forced to use a finite amount of electrodes, each passing their own current on the boundary. Since we cannot apply all possible current patterns we need to consider which ones to use. In general, given L electrodes on the boundary we apply a set of linearly independent current patterns (noting that any other current pattern will be a linear combination of those) and gather voltage data. Then the

voltage to current-density data is a finite-dimensional approximation to the DN-map and will be our data for the reconstruction. We cover some popular current patterns below.

An easy first choice to make would be to use the Fourier basis functions $\left\{\frac{1}{\sqrt{2\pi}}e^{in\theta}\right\}$ of $L^2(\partial\Omega)$, but since we wish to apply real-valued current patterns, we derive from this $L-1$ linearly independent vectors I^n of the form

$$I_\ell^n = \begin{cases} M\cos(n\theta_l), & n = 1, ..., \frac{L}{2} - 1 \\ M\cos(n\pi), & n = L/2 \\ M\sin((n - L/2)\theta_\ell), & n = \frac{L}{2} + 1, ..., L - 1 \end{cases}$$

where M is the maximum applied current, so our approximate DN map is an $L-1 \times L-1$ matrix. However, this requires L current sources and increases the cost and complexity of the system.

An alternative is to use a system with a bipolar current source requiring equal and opposite currents on a single pair of electrodes (to satisfy Kirchhoff's law). These are referred to as "skip n" patterns and are achieved by leaving n electrodes passive between the injecting pair. For example, applying skip 0 means we inject current on all pairs of adjacent electrodes, and skip 3 means there are 3 electrodes between the pair of active electrodes (i.e. we would inject on electrodes e_1 and e_4, e_2 and e_5, etc). This reduces the cost and complexity of the system, but since there are $L - \text{gcd}(L, n+1)$ linearly independent current patterns, care must the taken when choosing which skip pattern to use for a given number of electrodes [20]. The EIT lab in the Department of Mathematics at Colorado State University typically uses a bipolar current source to reduce the cost and technical overhead of multiple current sources.

1.2 LITERATURE REVIEW

In this section I will do a review of some papers relevant to the research in the rest of this paper. I start with a brief history of EIT, progress in the EIT problem and D-bar methods. Then I will do a review of literature concerning ventilation and cardiac signal separation in EIT images, internal boundary recovery, and applications of deep learning in EIT.

1.2.1 THE HISTORY OF EIT

CALDERÓN'S PROBLEM AND CGO SOLUTIONS

The study of the inverse conductivity problem got its start in 1980 due to Alberto Calderón, a civil engineer and mathematician. His work in geology prompted him to investigate the recovery of the internal conductivity of a body based on electrical measurements on the boundary, and his landmark paper [16] is credited with not only starting the study of EIT but as the beginning of the formal study of inverse problems. Here we note that even though Calderón's interest in the inverse conductivity problem arose from a practical interest in using it for geological surveys, this initial work assumed infinite precision and knowledge on the full boundary; work on computationally solving the problem with finite precision and noisy measurements wouldn't come for over 20 years.

Calderón posed the linearized version of the inverse conductivity problem: if the conductivity σ is a constant plus a perturbation (for ease let the constant be 1, or $\sigma(x) = 1 + \delta(x)$) then does the boundary data uniquely determine the perturbation $\delta(x)$? In his paper he shows that in this case the boundary data does uniquely determine the conductivities, and the study of EIT began.

He proves this using the quadratic form of the DN map

$$Q_\sigma(\phi) = \int_\Omega \sigma |\nabla w|^2 dx \tag{1.4}$$

where the voltages w are the solution to the generalized Laplace equation and ϕ are the induced voltages on the boundary. Letting $w = u + v$ where u solves Laplace's equation

$$\Delta u = 0 \in \Omega, \tag{1.5}$$

he then shows injectivity of the Fréchet differential about $\sigma = 1$

$$dQ_\sigma(\phi)|_{\sigma=1} = \int_\Omega \delta |\nabla u|^2 dx \tag{1.6}$$

which is sufficient to prove injectivity of the DN maps in the linearized case.

However, this was not enough to prove that DN maps are injective in the more general case. Despite that, Calderón's work opened the door to the EIT problem and paved the way for much of the work in the field. Specifically, his use of complex exponential solutions to Laplace's equation, given by

$$u_{\pm}(x) = e^{\pi i(z\cdot x)\pm\pi(a\cdot x)} \tag{1.7}$$

(where a and z are orthogonal vectors of equal magnitude in $\mathbb{R}^n$) led to more general proofs of injectivity of the DN map.

These types of complex solutions would turn out to be a key part of solving the EIT problem. Known as *complex geometric optics* (CGO) solutions due to their use in optics, these solutions grow exponentially in some directions but decay exponentially in others. CGO solutions were used by Faddeev in 1966 [38] as part of work on quantum scattering theory, and then later found use as part of many constructive proofs in inverse problems. CGO solutions allow us to use nonlinear Fourier analysis and construct custom nonlinear Fourier transforms to solve many problems (see [126] for a list of applications using CGO solutions of the Schrödinger equation).

GLOBAL UNIQUENESS

While Calderón's work focused on the linearized version of the problem we're interested in the full problem. The natural first step towards the more general case is proving that $\sigma \in \partial\Omega$ can be uniquely determined by boundary data. This was achieved by Kohn and Vogelius in 1984 with a proof of just that for the conditions dimension $n \geq 2$, $\Omega \in C^\infty$ bounded, $\sigma \in L^\infty(\overline{\Omega})$, and $\sigma \in C^\infty$ in some neighborhood of $\partial\Omega$ [72].

Global uniqueness theorems were not far behind: by 1987 Sylverster and Uhlmann, using CGO solutions, had proved global uniqueness for isotropic conductivities (i.e. the conductivity does not depend on the direction of the current) for $\sigma \in C^\infty$ in dimension $n \geq 3$ assuming $\partial\Omega \in C^\infty$ [122], and uniqueness in the dimension 2 case for near-constant isotropic conductivities [123]. In 1988 Nachman, Sylvester, and Uhlmann proved uniqueness in dimension 3 with the relaxed condition

of $\sigma \in C^{1,1}(\overline{\Omega})$ [95]. Nachman later that same year also relaxed the smoothness requirements of the boundary to $\partial\Omega \in C^{1,1}(\overline{\Omega})$ [96].

More general global uniqueness in the dimension 2 case turned out to be much more difficult: while the problem is overdetermined in dimensions $n \geq 3$ (see [124] for a detailed explanation), this is not the case for dimension $n = 2$. The first global uniqueness proof for $2 - D$ domains was not given until 1996 when Nachmann published a groundbreaking proof of injectivity of the DN map for dimension 2 isotropic $\sigma \in W^{2,p}, p > 1$ in Lipschitz domains . Not only was this an important theoretical breakthrough, but the proof was also constructive and outlined a metho [97]d for reconstructing the conductivities from the boundary data (in the infinite precision case). The method uses nonlinear Fourier analysis based on CGO solutions to the Schrödinger equation and the construction of a scattering transform. Since the technique relies on solving a PDE with a $\overline{\partial}$ operator it has become known as the D-bar method. This method will be the focus of the next section since much of the work in this dissertation heavily uses the scattering transform and related techniques.

The results have since been extended and generalized. For example, uniqueness for $\sigma \in W^{1,1}(\Omega)$ [15], non-smooth conductivities $\sigma \in L^{\infty}$ [5], and complex conductivities (called *admittivities*) $\gamma \in W^{2,\infty}$ (in bounded simply connected Lipschitz domains in $\mathbb{R}^2$) [47].

RECONSTRUCTION ALGORITHMS AND COMPUTATIONAL PROGRESS

An important part of EIT and the motivation that led to its invention is the ability to practically compute the internal conductivities of a body from electrical data on its boundary. While the proofs of uniqueness tell us that unique solutions exist, finding those solutions from real world data presents its own set of challenges. In particular we know that due to the ill-posedness of the problem, straight forward inverses are not possible. Some kind of regularization must be provided so that suitable solutions may be recovered despite the non-continous nature of the dependence of solutions on the data. Despite the challenges, great strides have been made in the field to recover useful solutions to the inverse conductivity problem in real world applications such as

medical imaging. Here we provide non-exhaustive overview of some of the most popular inversion approaches.

The first approach is linearization. Based on Calderón's work, it is assumed that the conductivities are a small perturbation from a constant or reference distribution and that linear inversion methods may be applied. Computation of Calderón's method was performed on simulated practical data by the early 1990s [19, 58–60] and experimental data in 2008 [8, 12]. Newton-like methods have been explored as well, including the successful Newton's One-Step Error Reconstructor (NOSER) algorithm [17], a least-squares method that utilizes one step of Newton-Raphson method starting at a constant conductivity guess.

Another popular and successful approach has been iterative methods. Like many other inverse problems, the inverse problem is written as a regularized least-squares problem that seeks a conductivity distribution σ that minimizes some functional (often using some type of quasi-Newton methods) [10, 11, 33, 34, 36]. However, like many least-squares problems this comes with its own challenges, including choosing a regularization term, determining appropriate hyper-parameters, and it is not guaranteed to converge quickly (and thus are unlikely to be useful for real-time imaging processes). Despite these challenges, when the solutions converge they are often highly accurate.

Also of note are the layer-stripping [114] (where layers are "stripped" off the domain by solving the problem near the boundary and taking that solution as the boundary information for a new problem) and statistical inversion methods from Somersalo [113]. While layer-stripping is unstable in the presence of noise, the Bayesian approach to solving inverse problems has gained some traction and have appeared in several applications (see [65, 74, 106, 116, 121]). Unfortunately, like many of the Newton-like iterative methods, the statistical inversion methods can be prohibitively slow and ill-suited for real time imaging applications.

Lastly we have a family of methods known as D-bar methods that rely on the D-bar operator $\overline{\partial}$, such as the method provided by Nachman's constructive proof in [97] that utilize CGO solutions. After the regularization proof in [71] these became the first direct methods with a true regularization strategey. Since the problem is ill-posed, a regularization strategy is required to recover

reasonable solutions in the presense of noise. The implementation was in 2000 by Siltanen and Mueller [112] for C^∞ radially symmetric conductivities and then applied them to simulated chest phantoms in 2002 [90]. Experimental tank data followed in 2004 [62], with application to cardiac imaging in human data not far behind in 2006 [61]. Clinical applications have included the recovery of pulmonary measures in children with cystic fibrosis [93] and the detection regions with air trapping in the lungs [89].

There have been many other advancements such as improved regularization techniques and methods [91], chest-shaped domains [49, 94], and numerical and computational improvements [30, 35]. This list is far from comprehensive: there are many people working to improve the efficacy of EIT through hardware, software, and signal and image processing.

Because D-bar methods are so central to this work we will return to D-bar methods and outline the core of Nachman's constructive proof using CGO solutions in the next section (see [92] for an in-depth look at EIT and D-bar methods).

1.2.2 SEPARATION OF CARDIAC AND VENTILATION SIGNALS IN EIT IMAGES

The current method for imaging lung perfusion (i.e. blood flow between the heart and lungs to oxygenate it before transport throughout the body) is a cardiac perfusion scan which involves injection of radioactive markers and the use of a gamma camera to see where the markers are absorbed. This is not ideal because it involves the injection of radioactive materials which may have their own side effects, such as risk of allergic reaction, exposure to ionizing radiation, and moving the patient which could involve removal of life support systems. Additionally, a gamma camera can cost from $400,000 to $2,000,000, and each use requires the purchase of radioactive dyes adding hundreds of dollars to the per-use cost.

This gives us clear motivation: separation of ventilation and cardiac signals could give a practical and safe method for monitoring pulmonary perfusion in patients that are at a risk of suffering conditions that could adversely affect blood flowing into the lungs (such as pulmonary embolism

or chest trauma) without expensive nuclear imaging machines or radioactive dyes. This would also allow continuous monitoring of patients and allow for data to be collected on unconscious patients or patients that may be risky to move. The ability to continuously monitor the V/Q ratio (the ratio of the amount of air reaching the tissue in the lung responsible for gas exchange to the amount of blood reaching the alveoli) of a patient with a portable device would be extremely useful for monitoring patients with critical ventilation or cardiac problems.

This is a very challenging problem due to the the nature of the signals involved; the source of the cardiac signal is blood flowing into and out of the heart and deformation of the heart itself. While blood more conductive than most internal tissues, heart tissue itself is conductive and thus the changes in electrical signals detectable at the skin due to blood flow are very small, especially when compared to the large signal changes due to non-conductive air flowing into and out of the lungs.

There are many techniques for separating mixed single (also called source separation) and these have been developed and studied independently of applications to EIT (see [102] for a more general review of blind source separation techniques).

One of the first attempts at separating these signals was electrocardiography (ECG/EKG) gating [37, 127]. This technique relies on a clean ECG measurment of several heart cycles taken synchronously with EIT data, and using the heart data from the ECG attempts to amplify the cardiac signal in the EIT data. However, this relies on long periods of stable cardiac cycles and does not perform well on patients with rapidly changing cardiac cycles or irregularities.

Fourier-analysis based techniques are another popular approach due to the well-developed tools in that field [68]. This leverages the fact that heart rates and respiratory rates are typically very different: resting heart rates are usually in the 50-100 bpm (0.83-1.67Hz) range, while resting respiratory rates are usually in the 10-40 breaths per minute (0.17-0.67Hz) range, depending on age and sex. Separation is then achieved by applying high/low/pass filters are appropriate to eliminate all but the desired frequencies. This is effective in some cases but is hindered by the large magnitude differences between the perfusion and ventilation signals, and overlap in harmonics between the

signals can cause loss of data during filtering. This is often combined with other approaches as a noise reduction step or to clean out remaining frequencies once the signals are primarily separated.

Principal component analysis (PCA) is another frequently used method due to its simplicity and speed [32]. PCA is a statistical method that takes a set of possibly correlated observations (in vector form) and returns a set of linearly independent vectors with maximum variance. That is, PCA finds an orthogonal basis for the data whose vectors that point in the directions of most variance. If perfusion and ventilation signals are linearly mixed, then the principal components will separate the signals into perfusion, ventilation, and noise components. In reality the mixing is not entirely linear, but the method is works well when the difference between noise and perfusion frequencies is large enough.

Other techniques include independent component analysys (ICA) [104], adaptive matched filtering [46, 129], and multi-dimensional ensemble empirical mode decomposition [117].

1.2.3 LOCATING ORGAN BOUNDARIES USING EIT

Another area of interest is finding the boundaries of internal structures of the domain. For example, the ability to locate the boundaries of organs when imaging a human thoracic cavity is another application of EIT that could be extremely useful for diagnosing and monitoring patients that are difficult or dangerous to move.

While approximate organ boundaries may be recovered from complete reconstructions, the reconstruction method itself may be smoothing, and the sharp discontinuities in conductivity (also called inclusions) where tissue transitions occur can be lost. The goal is to recover the shape and location of an unknown inclusion or inclusions embedded in a known background conductivity. Here we will do a review of some of the approaches that have been taken to recover the locations of such conductivities.

PROBING AND ENCLOSURE METHODS

Probing methods were first given by Ikehata in 1998 [54]. These methods rely on the behavior of the sequence of the energy gap

$$\int_{\partial\Omega} \{(\Lambda_0 - \Lambda_\Sigma)(v_n|_{\partial\Omega})\}\overline{v}_n|_{\partial\Omega} dS \tag{1.8}$$

for a chosen sequence $\{v_n\}$ of $H^1(\Omega)$ solutions to the Laplace equation, called a *needle sequence* that satisfy

$$\lim_{n\to\infty} (\|v_n(\cdot) - G(\cdot - x)\|_{L^2(K)} + \|\nabla\{v_n(\cdot) - G(\cdot - x)\}\|_{L^2(K)}) = 0 \tag{1.9}$$

for each compact set $K \subset \Omega\backslash\sigma([0,1])$, where Λ_0 is the DN map on the background and Λ_Σ is the DN map corresponding to an $(n-1)$ dimensional submanifold contained in Ω (see the definition of the inverse crack problem in [56] for details of Λ_Σ). The method receives its name from so-called *probes* or *needles*, piecewise linear curves contained in $\overline{\Omega}$ that connect the boundary $\partial\Omega$ to a point $x \in \Omega$ (called a needle with tip at x). The needle sequence blows up along the needle and is convergent locally outside the curve. Importantly, the existence of such a sequence is a consequence of the Runge approximation and relies on the denseness of the ranges of infinitely many linear integral operators so they can be very difficult to construct explicitly and use practically. Much of the work done since the introduction of these methods has been in the construction of such sequences. For example, Ikehata does this using a Carlemann function, a special fundamental solution for the Laplace operator [53].

Ikehata later introduced the *enclosure methods* to avoid using the Runge approximation property. These methods attempt to locate partial information about discontinuities in the conductivities by leveraging the discontinuities of coefficients of a partial differential equation. The original enclosure method uses exponential solutions with a large parameter and have been quite useful as an explicit replacement for the needle functions [55], but non-exponential solutions may also be used [57].

The enclosure methods allow the recovery of a single convex object embedded into a known background by use of the indicator function

$$I_\omega(\tau) = \int_{\partial\Omega} f_\omega(z;\tau)\left((\Lambda_\sigma - \Lambda_1)\overline{f_\omega(z;\tau)}\right) dS(z), \tag{1.10}$$

where f is typically an exponentially growing function in τ. For asymptotically large τ this allows the recovery of the convex hull of a single embedded discontinuity in Ω, but this poses challenges since f may blow up for large τ. This method works well in practice, but is unsuited for thoracic imaging due to the presense of multiple inclusions in human data.

1.2.4 DEEP LEARNING

Deep learning is a class of machine learning that utilizes layers of nonlinear processing units to perform tasks such as feature extraction, classification, and prediction. Since the boundaries of organs in 2D images are curves in the plane, deep learning can provide a wealth of tools for pulling these boundaries out of complicated nonlinear data such as scattering transforms. Here I review some previous applications of neural networks to EIT, noting that a technical introduction to deep learning will be provided in a later section.

Because deep learning is a relatively new technique that was long limited by available computation power its applications to EIT are very recent. In 2014 Michalikova et al. used a small network with radial basis activation functions to solve the EIT problem, attempting to map boundary voltage measurements to FEM model elements [88]. By 2017 Li et al. were performing a similar task, approximating conductivity distributions of simulated data using a 4-layer stacked autoencoder [81].

In 2017 Kosowski and Rymarczyk used an artificial neural network with EIT voltage data to detect a single round object against a constant background [75]. Their example application was the detection of fruit through an industrial process, and since forward feeding data through a neural network is extremely fast (on the scale of miliseconds) this could provide a method for real time imaging in an industrial setting where the goal is object detection and not necessarily accurate reconstructions of complicated conductivities. While their data set was small, only 320 data sets, it does serve as a proof of concept for a process that could be scaled up for industrial use. In 2018

they applied neural networks to test the spatial distribution of moisture inside of bricks and blocks of concrete [110].

In [48] Hamilton and Hauptmann use a convolutional neural network trained on simulated EIT data to sharpen reconstructions in post-processing with promising results. They achieve this by first performing a reconstruction with the D-bar method and then forward feeding the resulting image through a network trained to sharpen the images. The training data was created by simulating 4096 EIT data sets with the input to the network being the blurred reconstruction and the known (sharp) image as the output. In this way the network learns the function mapping blurred images to sharpened images. Also in 2017 Martin and Choi independently used artificial neural networks in as a post processing step in a linear solver to improve 2D and 3D reconstructions [83, 84]

Bayesian neural networks were applied to the problem of detecting gas bubbles in a circular water pipe by Lampinen, Vehtari, and Leinonen [78]. These networks use maximum likelihood estimation to find proper weights instead of standard training via optimization. They used these networks to approximate solutions to the EIT problem for detecting simulated gas bubbles in a circular water pipe (in particular they were concerned with finding the void fraction, or (area of the bubbles)/(area of the pipe)).

1.3 THE D-BAR METHOD OF RECONSTRUCTION

In this thesis we will focus on the D-bar method based on [97] as part of a constructive proof of the injectivity of the DN map in two dimensions. This method leverages exponentially growing complex geometric optics (CGO) solutions and the D-bar operator $\overline{\partial}$. It was first realized as a numerical reconstruction method in [112], and it directly solves the nonlinear inverse conductivity problem.

1.3.1 THE BOUNDARY INTEGRAL EQUATION AND CGO SOLUTIONS

First, recall that our goal is to recover information about σ in (1.1) given knowledge of the DN map Λ_σ (1.2). The method begins with a change of variables and the introduction of a complex

parameter k. We also must make a smoothness assumption on σ due to the 2nd order derivatives in (1.1), as well as an assumption that σ is constant near $\partial\Omega$.

Associate a point $(x, y) \in \mathbb{R}^2$ with $z = x + iy \in \mathbb{C}$ and assume $\sigma(z) \in C^2$ is bounded away from 0 for all $z \in \Omega$ and that $\sigma(z) \equiv 1$ in a neighborhood of $\partial\Omega$. Now define

$$q(z) = \sigma^{-1/2}\Delta\sigma^{1/2} \tag{1.11}$$

noting that σ constant near $\partial\Omega$ implies $q = 0$ near $\partial\Omega$, so we can smoothly extend $\sigma \equiv 1$ and $q \equiv 0$ out of Ω. This allows the change of variables $\widetilde{u} = \sigma^{1/2}u$, which transforms the conductivity equation into

$$(-\Delta + q(z))\widetilde{u}(z) = 0,\ z \in \mathbb{R}^2, \tag{1.12}$$

a well studied time-independent version of the Schrödinger equation.

A complex parameter is now introduced so that the CGO solutions can be leveraged. Let k be a complex parameter. Then the CGO solutions are solutions $\psi(z, k)$ to the Schrödinger equation

$$(-\Delta + q)\psi(\cdot, k) = 0,\ k \in \mathbb{C}/\{0\} \tag{1.13}$$

satisfying the asymptotic condition

$$e^{-ikz}\psi(z, k) - 1 \in W^{1,\tilde{p}}(\mathbb{R}^2) \tag{1.14}$$

for any $2 < \tilde{p} < \infty$. Note that the space $W^{1,\tilde{p}}(\mathbb{R}^2)$ is the Sobolev space of $L^{\tilde{p}}(\mathbb{R}^2)$ functions whose weak derivatives also belong to $L^{\tilde{p}}(\mathbb{R}^2)$, and that by the Sobolev embedding theorem $W^{1,\tilde{p}}(\mathbb{R}^2)$ functions are bounded and continuous [42]. Thus, the CGO solutions $\psi(z, k)$ are asymptotic to e^{ikz}, growing in some directions and decaying in others.

In order to define a PDE involving the CGO solutions, the D-bar operator $\overline{\partial}$ is needed. For a complex variable $z = x + iy$, define

$$\overline{\partial}_z = \frac{1}{2}\left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right), \quad \partial_z = \frac{1}{2}\left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right). \tag{1.15}$$

By the Cauchy-Riemann equations a function $f(z, \overline{z})$ is analytic if and only if $\overline{\partial}_z f = 0$ [115].

Now let $\mu(z, k) = e^{-ikz}\psi(z, k)$, noting that $-ikz$ is complex multiplication and μ is bounded, continuous, and asymptotic to 1 in the sense of (1.14). Substitute μ into the Schrödinger equation given above to get a D-bar equation for μ in the spatial variable z,

$$(-\Delta - 4ik\overline{\partial} + q)\mu(\cdot, k) = 0, \tag{1.16}$$

with the asymptotic condition $\mu(z, k) - 1 \in W^{1,\tilde{p}}(\mathbb{R}^2)$. The solution can be expressed as a Lippmann-Schwinger-type equation

$$\mu = 1 - g_k * (q\mu) \tag{1.17}$$

where $*$ is convolution and g_k is a fundamental solution to $(-\Delta - 4ik\overline{\partial})g_k(z) = \delta(z)$. Multiplying by e^{ikz} gives

$$\psi(z, k) = e^{ikz} - G_k * (q\psi) \tag{1.18}$$

where $G_k(z) = e^{ikz}g_k(z)$ is the Faddeev Green's function for the Laplacian [97]. We now need Alessandrini's identity [2, 92]:

ALESSANDRINI'S IDENTITY: For any two solutions $v_1, v_2 \in H^1(\Omega)$ to $(-\Delta + q_k)v_k = 0$ in Ω, the following identity holds:

$$\int_\Omega (q_1 - q_2)v_1 v_2 dx dy = \int_{\partial\Omega} v_1(\Lambda_{q_1} - \Lambda_{q_2})v_2 ds \tag{1.19}$$

where $\Lambda_q f := \frac{\partial f}{\partial v}|_{\partial\Omega}$ for functions $f \in H^{1/2}(\partial\Omega)$. See [92] or [2] for a proof and more details.

Now choose $q_2 = q = \sigma^{-1/2}\Delta\sigma^{1/2}, v_2 = \psi(\cdot, k), q_1 = 0$, and $v_1 = G_k(z - \zeta)$ with $\zeta \in \mathbb{C}\backslash\overline{\Omega}$. This results in

$$\int_{\partial\Omega} G_k(z - \zeta)(\Lambda_0 - \Lambda_q)\psi(\cdot, k) ds(\zeta) = -e^{ikz} + \psi(z, k),$$

and restricting z to $\partial\Omega$ gives the boundary integral equation

$$\psi(z,k)|_{\partial\Omega} = e^{ikz}|_{\partial\Omega} - \mathcal{S}_k(\Lambda_q - \Lambda_{q=0})\psi(\cdot,k) \tag{1.20}$$

where the Faddeev single-layer operator $\mathcal{S}_k$ is defined by

$$\mathcal{S}_k\phi(z) = \int_{\partial\Omega} G_k(z-\zeta)\phi(\zeta)ds(\zeta). \tag{1.21}$$

Since $\sigma = 1$ in a neighborhood of $\partial\Omega$, $\Lambda_q = \Lambda_\sigma$ and $\Lambda_{q=0} = \Lambda_1$ (the DN map for the homogeneous conductivity $\sigma = 1$), (1.20) provides a way of computing $\psi(z,k)$ on $\partial\Omega$.

1.3.2 THE SCATTERING TRANSFORM $\mathbf{t}(k)$

The scattering transform is defined as

$$\mathbf{t}(k) := \int_\Omega q(z)e^{i\bar{k}\bar{z}}\psi(z,k)dz, \tag{1.22}$$

which is approximately the Fourier transform of q evaluated at $(-2k_1, 2k_2)$ [92]. While $\mathbf{t}(k)$ has no physical meaning and cannot be measured directly, it is central to the D-bar method.

Again apply Alessandrini's identity, but now let $q_1 = q, v_1 = \psi(\cdot,k), q_2 = 0$, and $v_2 = e^{i\bar{k}\bar{z}}$. Then we get the scattering transform in terms of our data:

$$\mathbf{t}(k) = \int_{\partial\Omega} e^{i\bar{k}\bar{z}}(\Lambda_\sigma - \Lambda_1)\psi(\cdot,k)ds \tag{1.23}$$

Since $\psi(z,k)$ can be computed on $\partial\Omega$ via (1.20), we can compute $\mathbf{t}(k)$ for any $k \in \mathbb{C}$. Note that this is the second time that the data Λ_σ is used: once in (1.20), and again in (1.23).

An approximation to $\mathbf{t}(k)$ can be made using the asymptotic behavior of $\psi(z,k)$ by substituting e^{ikz} in for $\psi(z,k)$ into 1.23. This avoids the computation of $\psi(z,k)|_{\partial\Omega}$ and results in

$$\mathbf{t}^{\exp}(k) = \int_{\partial\Omega} e^{i\overline{k}\overline{z}}(\Lambda_\sigma - \Lambda_1)e^{ikz}ds. \tag{1.24}$$

In practice the scattering transform cannot be evaluated for all $k \in \mathbb{C}$ and must be truncated to avoid the amplification of noise caused by the exponentially growing $\psi(z,k)$. Truncating $\mathbf{t}(k)$ acts as a nonlinear low pass filter and provides regularization [71, 92]. The truncated scattering transform is written as

$$\mathbf{t}_R(k) = \begin{cases} \mathbf{t}(k) & \text{for } |k| \leq R, \\ 0 & \text{for } |k| > R. \end{cases} \tag{1.25}$$

for truncation radius R.

For the practical computation of $\mathbf{t}(k)$ we can expand (1.24) and (1.25) in terms of Fourier coefficients. Let $a_n(k)$ be the Fourier coefficients of e^{ikz} and $\langle\cdot,\cdot\rangle$ be the continuous inner product. Then

$$\mathbf{t}_R^{\exp}(k) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} a_m(\overline{k})a_n(k)\langle e^{im\theta}, (\Lambda_\sigma - \Lambda_1)e^{in\theta}\rangle, \;\; |k| \leq R, \tag{1.26}$$

and we can use $z = e^{in\theta}$ and Taylor expansion to find Fourier coefficients of e^{ikz} [70, 92].

1.3.3 THE D-BAR EQUATION

Now that we have the scattering transform $\mathbf{t}(k)$, we can solve the inverse scattering problem by solving $\overline{\partial}$ equations. Differentiating $\mu = 1 - g_k * (q\mu)$ with respect to $\overline{k}$ gives us the D-bar equation [97]

$$\frac{\partial}{\partial\overline{k}}\mu(z,k) = \frac{1}{4\pi\overline{k}}\mathbf{t}(k)e^{-i(kz+\overline{k}\overline{z})}\overline{\mu(z,k)}, \tag{1.27}$$

noting that $z \in \mathbb{R}^2$ is fixed and the variable is $k \in \mathbb{C}$.

The integral form is given by

$$\mu(z,k) = \lim_{R\to\infty}\left\{\frac{1}{\pi R^2}\int_{|k|\leq R}\mu(z,k)dk\right\} + \frac{1}{(2\pi)^2}\int_{\mathbb{R}^2}\frac{\mathbf{t}(k')}{(k-k')\overline{k}'}e^{-i(k'z+\overline{k'}\overline{z})}\overline{\mu(z,k')}dk_1'dk_2'. \tag{1.28}$$

Using the fact that the Green's function for the D-bar operator $\overline{\partial}_k$ is $\frac{1}{\pi k}$, (1.28) can be written as the following Fredholm equation of the second kind:

$$\mu(z,k) = 1 + \frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} \frac{\mathbf{t}(k')}{(k-k')\overline{k}'} e^{-i(k'z+\overline{k'}\overline{z})}\overline{\mu(z,k')}dk_1'dk_2'. \tag{1.29}$$

The solutions to these are well-studied and won't be covered here. EIT solutions are a special case of Jost solutions to $\overline{\partial}_k$ equations.

We conclude the section with an overview of the D-bar algorithm. Let $\Lambda_\sigma : H^{1/2}(\partial\Omega) \to H^{-1/2}(\partial\Omega)$ be the infinite precision DN map. Now let $q(x) = \sigma^{-1/2}\Delta\sigma^{1/2}$ (the Schrödinger potential) and $\psi(z,k) = e^{ikz}\mu(z,k)$ (the CGO solutions). Then the D-bar algorithm is as follows:

1. For all $k \in \mathbb{C}\backslash 0$, solve the boundary integral equation
   $\psi(\cdot,k)|_{\partial\Omega} = e^{ikz}|_{\partial\Omega} - \mathcal{S}_k(\Lambda_\sigma - \Lambda_1)\psi(\cdot,,k)$, where $\mathcal{S}_k(\phi)(z) = \int_{\partial\Omega} G_k(z-\zeta)\phi(\zeta)ds(\zeta)$ is the Faddeev single-layer operator with $G_k(z) = e^{ikz}g_k(z)$ is the Faddeev Green's function for the Laplacian.

2. Choose a truncation radius R and evaluate the scattering transform $\mathbf{t}$ for $|k| \leq R$
   $\mathbf{t}(k) = \int_{\partial\Omega} e^{i\overline{k}\overline{z}}(\Lambda_\sigma - \Lambda_1)\psi(\cdot,k)ds$

3. For each $z \in \Omega$ solve the D-bar equation
   $\overline{\partial}_k\mu(z,k) = \frac{1}{4k\pi}\mathbf{t}(k)e_{-k}(z)\overline{\mu(z,k)}$

4. Solve for conductivity:
   $\sigma(z) = (\mu(z,0))^2$

1.4 INTRODUCTION TO DEEP LEARNING

Deep learning is a class of machine learning that utilizes layers of nonlinear processing units to perform tasks such as feature extraction, classification, and prediction. Deep learning specifically makes use of artificial neural networks composed of multiple layers of nonlinear processing units. These networks are incredibly powerful tools that are being applied to a huge variety of tasks,

including image classification, image segmentation, natural language processing, autonomous vehicles, and medical diagnosis assistance.

Since the boundaries of organs in 2D images are curves in the plane, deep learning can provide a wealth of tools for pulling these boundaries out of complicated nonlinear data such as scattering transforms. Our goal is to leverage deep learning to reconstruct the boundaries of organs by training an artificial neural network on scattering transform/boundary pairs and then forward feeding a scattering transform through the network to get an approximation of the organ boundaries. Here we provide a brief introduction to artificial neural networks and deep learning, for an in depth history of neural networks see [128].

1.4.1 ARTIFICIAL NEURAL NETWORKS

An artificial neural network (ANN) is a collection of nodes, also referred to as neurons or perceptrons, that are organized into layers that are then connected to successive layers. As the name implies, these neural networks are modeled after the biological networks of neurons that can be found in brains. Data is passed into the *input layer*, and the result of the computation is held in the *output* layer. Any layer between the input and output is referred to as a *hidden layer* and is typically not interacted with by the user during computation.

Connections between nodes are represented by *weights*, a numerical representation of the strength of the connection between those two nodes. Typically nodes in a layer are only connected to nodes in the layers before and after that layer with no intra-layer connections. These connections are linear, where the output of a node is its value times the weight of the connection leading to the next node, and the input to a (non input layer) node is simply the sum of all the outputs of the nodes in the previous layer along the connections that lead to it.

Formally, let o_j be the outputs of a layer, let x_i be the nodes in the next layer, and $w_{i,j}$ the weight connecting o_j to x_i. Then we can compute the values of the nodes in the next layer as

$$x_i = \Sigma_j o_j * w_{i,j} + w_{0,i}, \tag{1.30}$$

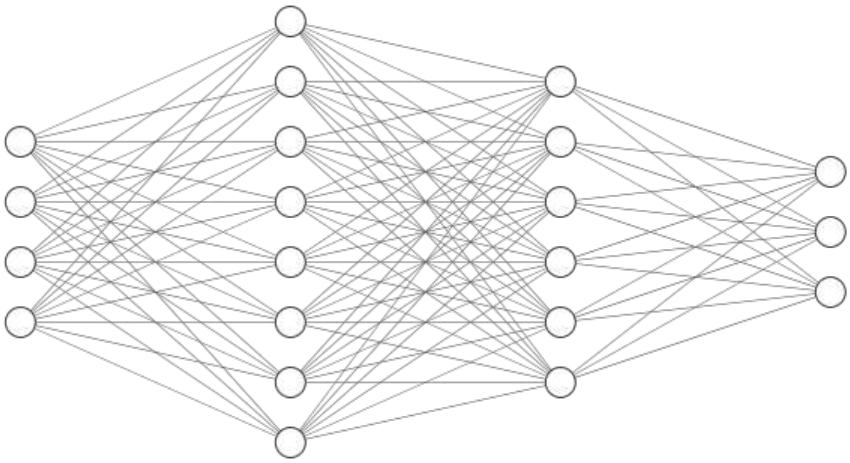


Figure 1.2: A fully connected artificial neural network with inputs in $\mathbb{R}^4$ and outputs in $\mathbb{R}^3$.

where $w_{0,i}$ is a bias term added at each node. Note that this is a linear operation, and so the output of a basic neural network is a linear function of its inputs. In fact, since each layer is linear it can be represented (and computed) by matrix operations. Let $\vec{x}^k$ be the vector of nodes of the k-th layer with nodes x_i^k, $W^k = \{w^k\}_{i,j}$ be the matrix of weights connecting nodes in layers $k, and k+1$ (where $w_{i,j}^k$ connects x_i^k to x_j^{k+1}), and $\vec{w}_0^k$ be the bias vector for layer k. Then layer $k+1$ can be computed as

$$\vec{x}^{k+1} = W^k \vec{x}^k + \vec{w}_0^k. \tag{1.31}$$

Thus each layer can be computed from the previous layer by a single matrix multiplication and a vector addition.

However if neural networks could only learn linear functions, they wouldn't be very useful! The way to get around this is to add an *activation function* $f(x)$ to the output of each layer. Some common examples include a rectified linear unit (ReLU) $f(x) = \max(0, x)$, $\tanh$, sigmoid $f(x) = (1 + e^{-x})^{-1}$, and softplus $f(x) = \log(1 + e^x)$ (also called SmoothReLU since it is a smooth approximation to ReLU). The nonlinearity of these functions are a powerful feature that allows

neural networks to learn nonlinearities in data. Note that the bias vector $\vec{w}_0^k$ simply shifts the output of the activation functions. A network like this with at least 3 layers (input, at least one hidden, and output) is called a *multilayer perceptron* (MLP).

Here we note that neural networks are really just approximations of functions. In fact, the *universal approximation theorem* states that an MLP with only one hidden layer using sigmoid activations can approximate any continuous function on compacts subsets of $\mathbb{R}^n$ arbitrarily closely [25] (though this does not guarantee the ability of a neural network to actually learn these functions, only that they may represent them).

We can now think of neural networks as approximations of real-valued functions $F(\vec{x}) : \mathbb{R}^n \to \mathbb{R}^m$ (for example, mapping vectors of data to classifications).

1.4.2 TRAINING

The goal of training a neural network is as follows: we want to programmatically find weights for the network $f(\vec{x})$ that cause it to approximate some target function $F(\vec{x})$ of the input data $\vec{x}$. If we have a set of known input/output pairs $\vec{x}, \vec{y} = F(\vec{x})$, then we can *train* the network by minimizing some error function $e(\vec{x})$ on the outputs of the network by updating the weights. Common error functions include least-squares $e(\vec{x}) = \frac{1}{2}(\vec{y} - f(\vec{x}))^2$ and cross-entropy (for classification).

The standard method for training the weights in a network is *backpropagation*. Since feeding data forward through a neural network is a composition of functions at each layer, when we take the gradient of the error function $e(\vec{x})$ with respect to the weights we can use basic calculus and the chain rule to compute it. For example, the weight updates for gradient descent at step t can be computed by

$$w_{i,j}(t+1) = w_{i,j}(t) + \gamma \frac{\partial e(\vec{x})}{\partial w_{i,j}}, \tag{1.32}$$

where γ is the learning rate. Gradient descent algorithms are by far the most popular algorithms for supervised training of neural networks and there are many variations including steepest de-

scent, conjugate gradient, quasi-Newton methods, and stochastic gradient descent methods. For an overview and practical explanation of the different types of algorithms see [109].

It is also common to train the network in *batches* where instead of computing the error for individual input/output pairs the error is minimized over larger batches of data. This is primarily to prevent *overfitting*, which is when the network fits well to the training data but performs very poorly when presented with new data (similar to fitting n data points to an n-th degree polynomial). Batching ensures that the network never sees individual data points during training and helps it generalize to new data. Additionally a stochasic term is often added to the gradient to further prevent overfitting.

The process, then, is to forward feed a batch of data through the network, and then adjust the weights of the network to minimize an error function that compares the actual outputs corresponding to the batch with the outputs of the neural network for the batch. Doing this one time for all the data in the training set is called an iteration or epoch. Since we want the network to perform well on data that it wasn't trained on it is typical to use a random set of data that isn't part of the training set to use as a validation set; iterations are then performed until the error is as small as possible on the validation set.

1.4.3 CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks (CNN) are a special class of ANNs that have proved to be very powerful tools for tasks such as image pre/post processing, image segmenting, and image classification. As the name suggests, they rely on the convolution kernels (also known as filters) which are used heavily in image processing. Instead of layers of nodes where every node is connected to each node in the next layer, CNNs have convolutional layers where the operator between layers is a convolution.

In a 2D discrete convolution the filter acts as a window sliding across the matrix being convolved, and a new matrix is output whose entries are the sum of the values in the filter multiplied by the values that they overlay in the matrix. In image processing these kernels can perform many

functions such as blurring, sharpening, and edge detection. For example, a 5×5 approximation to a Guassian blur can be achieved by convolving

$$K = \frac{1}{256} \cdot \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \tag{1.33}$$

with an image. When this matrix is overlayed on an image, the pixel at the center of the filter is weighted most heavily, but pixels surrounding that pixel play a role in the output pixel's intensity as well (with pixels further away having less effect). Similarly, 3x3 Sobel edge operators (horizontal and vertical edge detection kernels) are defined by

$$K_h = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, K_v = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \tag{1.34}$$

and performing a convolution with one of these matrices and an image results in an image mainly composed of the edges from the original. We can see in example 1.3 that the horizontal edge detection kernel generally produces horizontal edges while supressing vertical edges, and vice versa for the vertical edge detection kernel.

Note that matrix convolutions can be formulated as matrix multiplications so that efficient matrix multiplications on modern hardware can be leveraged when forward feeding through the CNN.

1.4.4 CONVOLUTIONAL LAYERS

What the network learns during training are the weights in convolutional filters. The hope, then, is that the kernels learned will represent different features of the training data, such as collections of edges and various shapes (for example, a filter or combination of filters could pick out different

Figure 1.3: Left: Original image. Center: The original image convolved with the horizontal edge detection filter. Right: The original image convolved with the vertical edge detection filter.

types of ears in pictures of animals). We have to be careful when assuming what kinds of features will be learned since the optimized weights will generally not be interpretable, especially in layers beyond the first convolutional layer.

In each convolutional layer we choose how many filters will be learned, say n, and the output of that layer is $n * m$ images where m is the number of images in the previous layer. For example, we could have an input image convolved with 64 filters, so the second layer would contain 64 copies of the original image that has been convolved with each of the 64 filters. If the next convolutional layer also had 64 filters, then the third layer would contain $64 * 64 = 4096$ copies of the original image of which each copy has been convolved twice with different filters.

Note that if the images are multi-channel, such as RGB, then the filters in the first layer will also have depth and usually output single channel images after the first convolutions.

1.4.5 POOLING AND DROPOUT

Since the number of images at each layer grows exponentially there are several steps that are taken to reduce the amount of data that must be held in memory. The input images are typically scaled to be as small as possible while still retaining the relevant information. Another option is to not pad the input image on the boundary with zeros so that each convolution reduces the size of

the image; this can be achieved by adjusting the *stride* as well, by "sliding" the filter several steps at a time instead of one step at a time.

The most common approach to reducing the size of the data in the network is to introduce *pooling layers*. A pooling layer takes sections of the images and performs an operation to reduce the size of those sections by replacing the entire section with a single piece of information about the section. For example, a *max-pool layer* will separate each image into $p \times p$ blocks and output only the maximum value from each block.

$$\left[\begin{array}{cc|cc} 1 & 3 & 2 & 1 \\ 0 & 1 & 1 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{array}\right] \xrightarrow{max-pool} \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix} \tag{1.35}$$

Other pool functions include averaging pooling and $L2$-norm pooling, although max-pooling is much more common and tends to work better in practice. This strategy greatly reduces the data in a layer each time it is performed and speeds up training since weights associated with any pixel that was discarded in a max-pool layer do not need to change during that particular optimization step.

One last common operation that we find in the training of neural networks is known as *dropout*. Dropout is the practice of randomly setting a percentage of weights between layers to 0. This is thought to reduce overfitting since the network must learn features in multiple ways: there is no single "path" for information about a specific feature to flow through the network. Typical percentages range from 20% to 50%. Dropout, batching, and stochastic gradients have proven to be powerful ways of preventing network overfitting.

Lastly the network ends with dense layers. The specifics of these dense layers are typically ad hoc, but these behave as in a normal multilayer perceptron. These layers are densely connected and have more weights to learn than the relatively sparse convolutional layers. The output of the last dense layer is the output of the network and represents the output of the function that the network

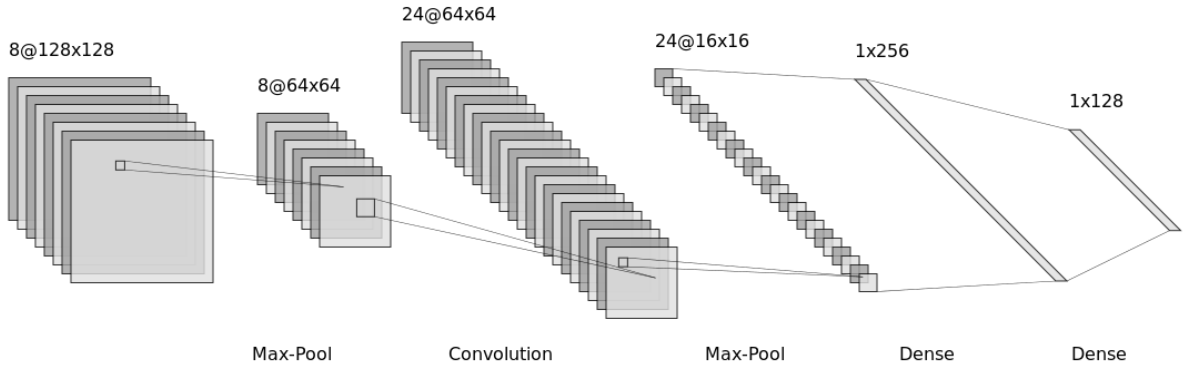


Figure 1.4: A convolutional neural network.

has learned. This output is often a probability vector where each bin represents a classification label (and the outputs are scaled with a function such as softmax), but could represent any type of data and even images.

There are many more details and subtleties to artificial networks that are beyond the scope of this dissertation, but I encourage the reader to venture into this new world and discover some of the really fascinating applications where neural networks are being used in.

Chapter 2

SEPARATION OF CARDIAC AND VENTILATION SIGNALS

2.1 INTRODUCTION

I have investigated several methods for separating signals, considering methods used for extraction of cardiac signals in echocardiograms (ECG) as well as more general methods, including wavelet transforms, independent component analysis [85], and ECG gated separation methods [103]. While the separation of signals in EIT images is still an open problem, I will focus one particular method in depth from the paper ‘‘Dynamic separation of pulmonary and cardiac changes in electrical impedance tomography’’ [32] and discuss the implementation, modifications, and results in a later chapter.

We first build some notation. Let X be an $N \times M$ matrix of vectorized EIT images with M pixels taken at equally spaced times $t_1, t_2, ..., t_N$, where row k corresponds to the image at time t_k.

We assume that each set of EIT images X is an additive combination of three signals: unknown ventilation, unknown cardiac, and noise (or remainder). Thus we write $X = X_V + X_C + X_R$ and our goal is to recover X_V and X_C, matrices that contain EIT images with changes only due to ventilation and cardiac signals respectively. The method in [32] is based on time domain filtering using templates.

2.1.1 TIME DOMAIN FILTERING TEMPLATES

Assume that a discrete time signal $\mathbf{x}(t)$ can be sufficiently approximated by a linear combination of some set of K *template signals* $\mathbf{b}_k(t), k = 1, ..., K$, with $t = t_1, ..., t_N$. Then

$$\begin{pmatrix} x(t_1) \\ \vdots \\ x(t_N) \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} b_{11} & \dots & b_{1K} \\ \vdots & \dots & \vdots \\ b_{N1} & \dots & b_{NK} \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_K \end{pmatrix}$$

for some weights a_i or we let $B = [\mathbf{b}_1 \dots \mathbf{b}_K]$, $\mathbf{a} = (a_1 \dots a_K)^T$ and $\mathbf{x} = B\mathbf{a}$. Since this doesn't always have a solution, we need to find the weights $\mathbf{a}$ such that

$$\mathbf{a} = \arg\min_{\mathbf{y} \in \mathbb{R}^K} (||B\mathbf{y} - \mathbf{x}||_2).$$

This has least-squares solution $\mathbf{a} = B^+\mathbf{x}$ where B^+ is the Moore-Penrose pseudoinverse of B. So the approximate signal $\overline{x}$ can be computed as

$$\overline{\mathbf{x}} = BB^+\mathbf{x}.$$

If we let X be an $N \times M$ matrix we can extend the above with the system $X = BA$ where the columns of X are the signals and A is now a weight matrix, noting that now $\mathbf{x}_i = B\mathbf{a}_i$ and we solve

$$A = \arg\min_{A \in \mathbb{R}^{K \times N}} (||BA - X||_2). \tag{2.1}$$

As above, this has the least-squares solution $\overline{X} = BB^+X = \text{LMS}(X, B)$.

If we let X be our matrix of EIT images, where each column corresponds to a single pixel changing over time, then if we can find template matrices B_V and B_C corresponding to ventilation and cardiac signals respectively, we can compute approximate image sets $\overline{X}_V$ and $\overline{X}_C$ containing separated images corresponding to changes due to ventilation and cardiac signals.

2.1.2 PRINCIPAL COMPONENT ANALYSIS

The approach to finding template matrices involves a well-studied statistical technique called *principal component analysis* (PCA) that takes in a set of observations and returns an ordered set of orthogonal basis vectors for the data called principal components ordered by decreasing variance

(i.e. the first principal component accounts for the highest amount of variability in the data). For a matrix of observations X this corresponds to finding the eigenvectors and eigenvalues of the covariance matrix $X^T X$.

Mathematically this is the same as finding the singular value decomposition $X = U\Sigma V^T$, and the columns of $T = U\Sigma$ are the principal components. Geometrically, the principal components are the principal semiaxes of the smallest ellipsoid that fits around the data. We note here that [32] does not use the SVD language, but that it provides a simple interpretation of principal components.

We can choose to ignore principal components that correspond to low variance in the data (i.e. the directions in which the data do not change very much) by using only the first k columns of $T = U\Sigma$. This is equivalent to ignoring the eigenvectors (of $X^T X$) with small corresponding eigenvalues. We give this the notation PCA(X, k).

2.1.3 THE SEPARATION ALGORITHM

Since the primary source of change in EIT images is ventilation (the cardiac and noise signals are very small in comparison), we choose template signal $B_V = \text{PCA}(X, 1)$. Then a first approximation of the ventilation matrix is computed as $X_V = \text{LMS}(X, B_V)$. This approximation is comparable to functional EIT images [32].

Next we find the first approximation of the cardiac matrix by $X_C = X - X_V$. This matrix is typically very noisy (due to X_R) so we pass X_C through a finite impulse response (FIR) bandpass filter between 0.92 and 4.6 Hz (55 and 280 bpm). In [32] filter order 30 is empirically chosen, and the filter is applied forwards and backwards to cancel out filter delays. See [100] for details on FIR filtering. So $X_C \xrightarrow{\text{FIR filter}} X_{C_{BP}}$.

Now we build a template matrix using the PCA of the filtered matrix: $B_C = \text{PCA}(X_{C_{BP}}, 2)$. Additionally, to account for blood flow through the electrode plane, we use phase-shifted versions of the columns of B_C (denoted B_{C1} and B_{C2}), shifting each by $\pm\frac{1}{3}$ of the average heart rate of the subject. Then

$$\widetilde{B_C} = [B_{C1}, {}^{+\frac{1}{3}}B_{C1}, {}^{-\frac{1}{3}}B_{C1}, B_{C1}, {}^{+\frac{1}{3}}B_{C2}, {}^{-\frac{1}{3}}B_{C2}]$$

is the cardiac template matrix.

The second cardiac approximation is then $X_{C1} = \text{LMS}(X_C, \widetilde{B_C})$ We also fit the cardiac template matrix to the ventilation approximation to extract any remaining cardiac signal from it: $X_{C2} = \text{LMS}(X_V, B_C)$. We then compute our final approximations:

$$\overline{X}_C = X_{C1} + X_{C2}, \ \overline{X}_V = X_V - X_{C2}$$

2.2 RESULTS

2.2.1 SIGNAL SEPARATION

Here we present the results of a modified implementation of the separation algorithm described in [32]. The principal components are computed via singular value decomposition, and all filtering performed using the MATLAB signal processing toolbox. All datasets were taken *in vivo* with various patients of Children's Hospital Colorado under COIRB approval 14-0652 whose data has been anonymized. We present representative results from a single dataset collected using the ACE1 system here at CSU with skip 0 patterns (see [87] for technical specifications of the ACE1 system).

First, we will look at the first 4 principal components of a set of EIT images gathered from a patient during tidal breathing. It's clear that the first principal component is primarily comprised of ventilation signal (see Figure 2.1). Note that the pixel data is is normalized (each pixel $X_i(t)$ is a unit vector) and centered at zero (mean shifted to 0), so the y-axis now represents relative conductivity and not true conductivity. The periodic behavior arises from low conductivity air filling and leaving the lungs.

Note the relative size of the amplitudes of each component: the first component is four times as large as the second component and ten times as large as the third and fourth components, meaning that the first component contains most of the information about how the pixels vary over time.

We also provide a spectral analysis of the set to visualize the real frequency composition of the pixel space (see Figure 2.2). Normal breathing rate is approximately 12-20 breaths per minute, or $0.2 - 0.33$Hz. The least-squares spectral analysis was performed using a Lomb-Scargle peri-

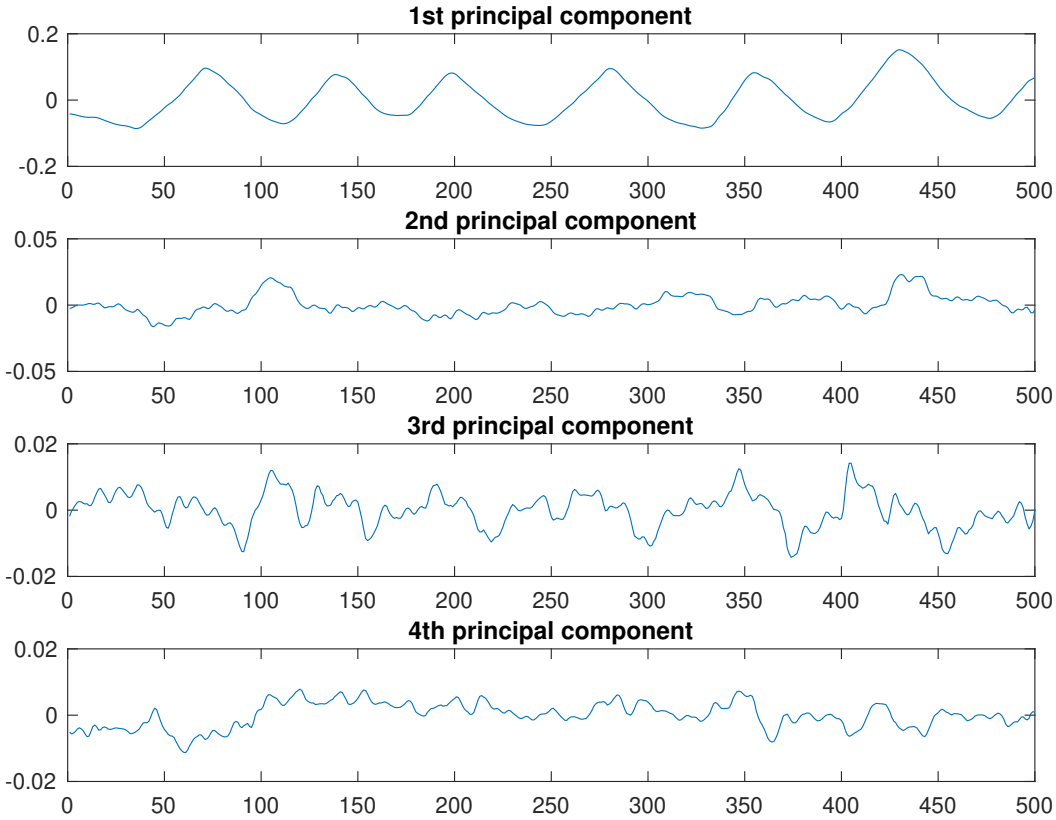


Figure 2.1: The first four principal components of a tidal breathing set. The vertical axis is relative conductivity and the horizontal axis is time in frames (20 frames/s).

odogram. Natural variations in breathing rate cause the power to be spread out over the lower frequencies, but it's clear that most of the power corresponds to the ventilation signal.

We should also note the lack of higher frequencies present in the signal. This could be due to the regularization caused by truncating the scattering transform, which naturally acts as a low-pass filter. A heart rate of 60bpm corresponds to 1Hz, so we should expect any cardiac signal to be in the 0.8Hz to 2Hz range (roughly 50bpm to 120bpm, varying by age, sex, and other factors such as stress at time of data collection). Although the regularization provided by the scattering transform is necessary, this may result in a loss of cardiac signal components.

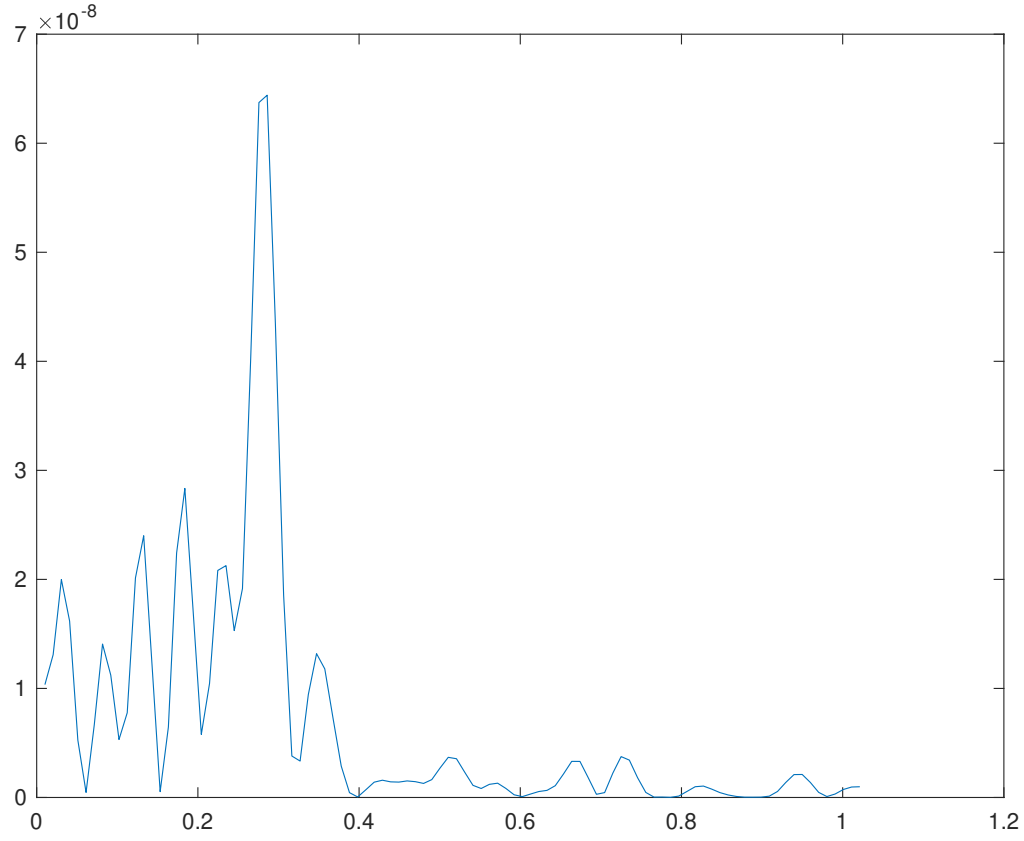


Figure 2.2: A plot of power against frequency (Hz) in the pixels of a tidal breathing data set.

Next we use least-mean squares to fit a number of principal components to our image matrix, $X_V = \text{LMS}(X, B_V)$ where B_v is the matrix whose columns are the principal components of X that we choose to fit with. At this step we must decide how many of the principal components that we'd like to fit to the data; in this case we will choose to fit only the first component since it clearly contains a strong ventilation signal, while the others appear to be noise. A possible approach to automate this is to use a cutoff criterion involving the singular values, but we did not thoroughly investigate this. Fitting too many principal components to the image matrix may amplify noise and result in a loss of ventilation signal. Once the number of components is chosen, we perform a fit and subtract that matrix from the original image matrix, resulting in a matrix that primarily contains cardiac signal and noise.

We can look at the principal components of X_V (Figure 2.3) to see that is comprised of only ventilation signal since the second principal component is noise on the order of 10^{-16}.

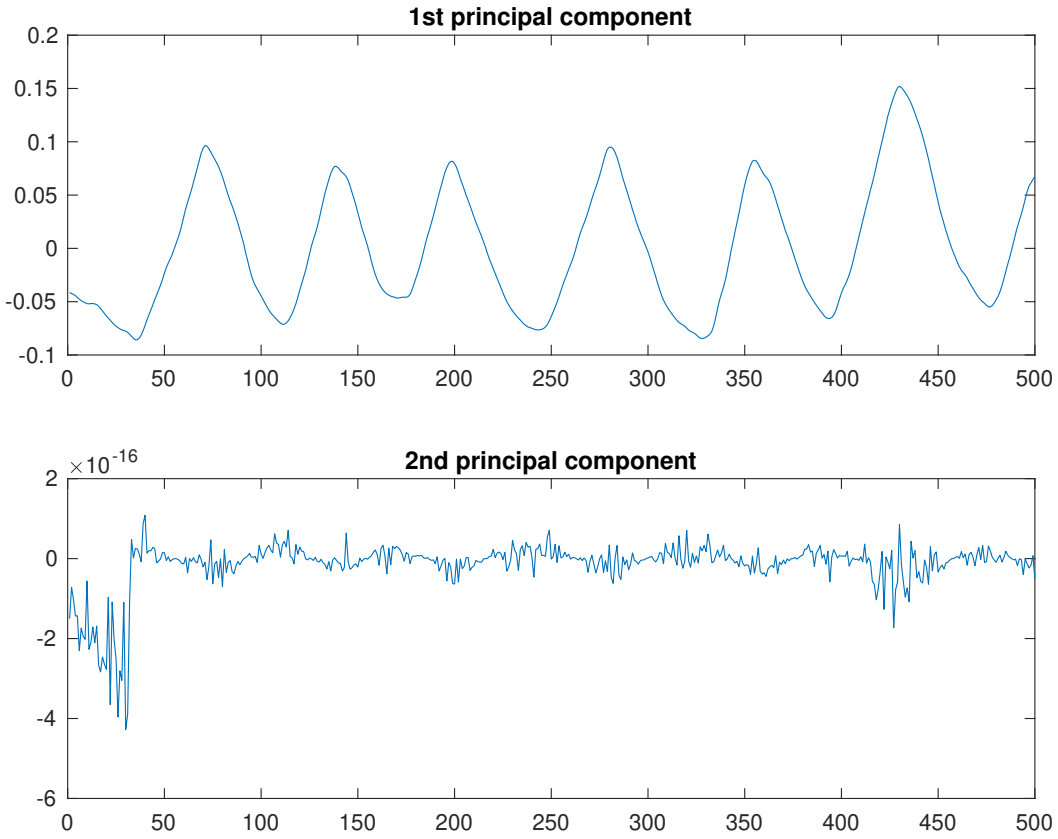


Figure 2.3: The first two principal components of X_V, the first ventilation matrix approximation.

We then subtract our ventilation approximation from our image matrix to obtain the first cardiac approximation, $X_C = X - X_V$. Since we expect this signal to be noisy, we pass it through a bandpass filter before performing the LMS fit. Our results confirm those of [32] that a filter order of 30 with a 0.92Hz to 4.6Hz bandpass works well, but we found that a filter range of 0.75Hz to 4.6Hz performs better.

Once X_C is bandpass filtered, we fit it to the first two principal components as well as each of the components phase shifted by $\pm 1/3$ the mean heart rate for the patient (which we put as columns

of the matrix B_C), which we approximate at 60bpm from ECG measurements taken at the time of data collection. The shifted components are to account for blood flow between the heart and lungs in the plane of measurement. So $X_{C2} = \text{LMS}(X_C, B_C)$ is our second cardiac approximation.

At this point we modify the algorithm provided in [32]. We found that fitting the small cardiac signals to the ventilation approximation X_V and subtracting them out often amplifies noise instead of reducing it. This is because during the least-squares fitting we're finding the linear combination of the columns of B_C that solves (2.1), since since the vectors in X_V are much larger than those of B_C, the matrix $\text{LMS}(X_V, B_C)$ has large noisy vectors. Subsequently, subtracting that matrix from X_V may result in a matrix that more closely matches noise than the ventilation signal.

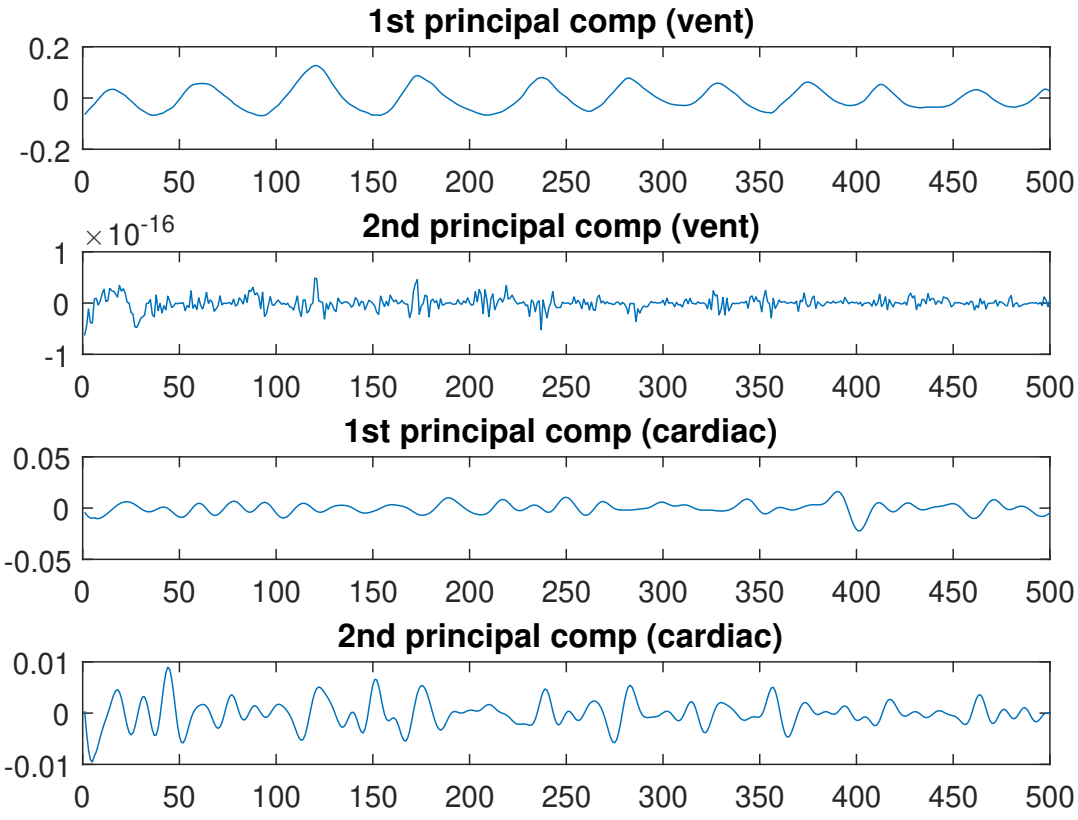


Figure 2.4: The first two principal components of the final ventilation and cardiac approximation matrices X_V and X_{C2}.

We can see in Figure 2.4 the principal components of the final two approximation matrices X_V and X_{C2} that the ventilation signal is well separated, but it is not clear if we have found a cardiac signal or if we have found noise. We perform a spectral analysis on X_{C2} and see that the most dominant frequencies are just above 1Hz, which does not correlate with the approximate 110-120bpm heart rate of the subject, so the cardiac signal, if present, is lost in the noise. Also note that the power is very low, on the order of 10^{-10}, so if the cardiac signal is present it is just too small to detect.

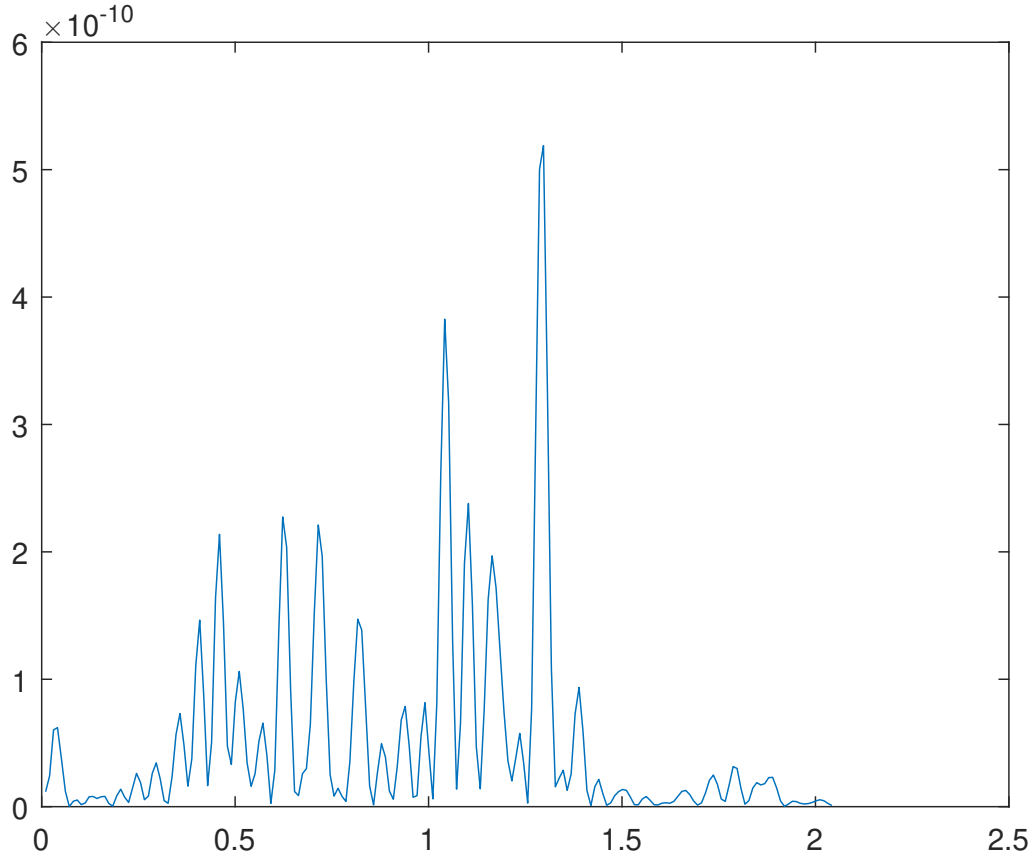


Figure 2.5: A plot of power against frequency (Hz) of the final cardiac approximation matrix X_{C2}.

For comparison we can perform the same type of analysis on a data set where the subject was holding their breath. Since there is no ventilation occurring there should only be two signals

present: changes due to cardiac activity and noise. However, when we inspect the first 4 principal components of a breath-holding set we immediately see that this is not the case as there is a larger low frequency component to the set. This may be due to involuntary contractions of lung tissue, but it's an issue that is actively being investigated. Despite that, it appears that the third principal component is a cardiac signal and can easily be separated from the low-frequency drift.

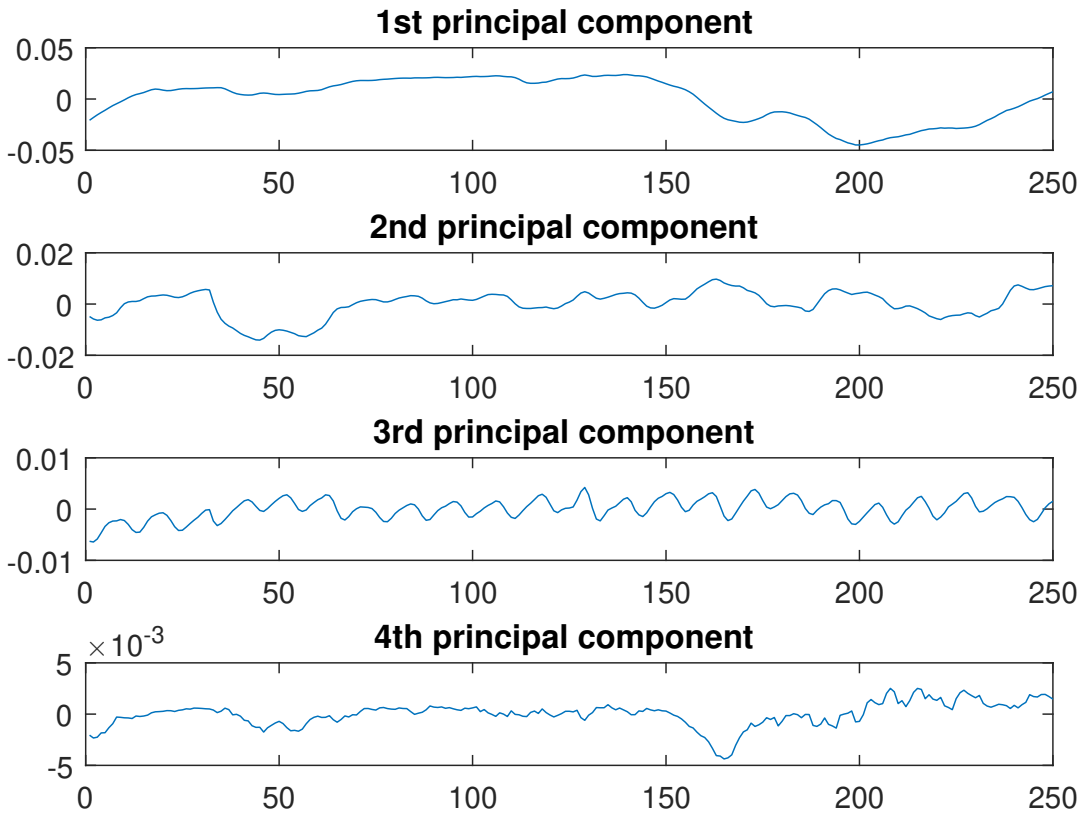


Figure 2.6: The first four principal components of a breath holding set.

We can verify that we have found the cardiac signal by comparing the third principal component in Figure 2.6 to the ECG collected at the same time (noting that synchronization is not perfect since the ACE1 system and the ECG system are started independently by separate people, see Figure 2.7. The patient had roughly 21 heartbeats during the collection, at a rate of about 120bpm, and we

count 21 peaks in both the ECG and the third component of Figure 2.6. The ECG provides strong heuristic evidence that the third component corresponds to the cardiac signal.

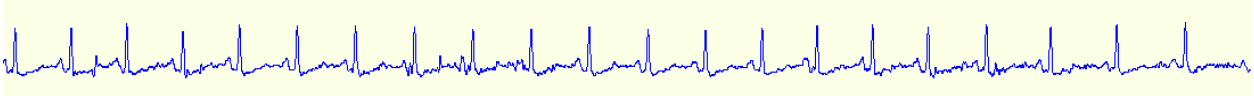


Figure 2.7: ECG collected during a breath-holding set. Compare to the third component in Figure 2.6

RECOVERY OF SPIROMETRY MEASURES FROM EIT IMAGES USING SIGNAL SEPARATION

Spirometry measures patient lung function by measuring the volume of air that a patient inhales and exhales, and is important for clinicians attempting to diagnose lung obstruction diseases and conditions such as pulmonary fibrosis. Since most variation in EIT thoracic images arise from changes in lung volume, we separate the ventilation signal from images taken during a pulmonary function test (PFT), also known as spirometry. During a PFT the patient inhales as much air as possible, and then exhales as quickly as possible. Several measures are tracked, centered around the volume of air and the rate of exhalation. Measures include the total amount of air exhaled, also called forced vital capacity (FVC), the amount of air exhaled over some period of time, called forced expiratory volume (over one second this is referred to as FEV1), as well as flow rates, and tidal volume, the amount of air inhaled and exhaled at rest.

The measure we focus on here is the ratio $\frac{\text{FEV1}}{\text{FVC}}$, also referred to as FEV1%FVC. This is the ratio of the amount of air that the patient can forcefully exhale in one second, divided by the total amount of air exhaled. A normal ratio for healthy adults is roughly 80% [39], but patients with pulmonary obstructions caused by disease or injury may cause the FEV1/FVC to be lower due to a lower outflow of air during the maneuver. While we cannot directly measure lung volume with EIT images, we can use the changes in conductivities in the lung region as a stand in for changes in lung volume; this allows us to approximate measures such as FEV1/FVC even though we cannot directly compute FEV1 or FVC. The algorithm is as follows.

First, separate the ventilation matrix X_V out of X using the methods in the previous section, and compute the first principal component $\vec{v}$ of X_V. Then, using $\vec{v}$ find the frame number corresponding to maximum inhalation before the exhalation maneuver (total lung capacity (TLC)), the frame number corresponding to one second after TLC (FEV1), and the frame corresponding to maximum expiration (end of forced expiration (EFE)). The conductivity in pixel p at these times is $\sigma_{\text{TLC}}(p), \sigma_{1\text{s}}(p)$, and $\sigma_{\text{EFE}}(p)$ respectively. Letting σ_M and σ_m be the maximum and minimum, respectively, over all pixels and all frames in the data set, the local volume fractions of air are computed as

$$f_{\text{TLC}}(p) = \frac{\sigma_{\text{TLC}}(p) - \sigma_M}{\sigma_m - \sigma_M},$$

$$f_{1\text{s}}(p) = \frac{\sigma_{1\text{s}}(p) - \sigma_M}{\sigma_m - \sigma_M},$$

$$f_{\text{EFE}}(p) = \frac{\sigma_{\text{EFE}}(p) - \sigma_M}{\sigma_m - \sigma_M}.$$

The volume corresponding to a pixel V_p is estimated by 1000 times the area of the pixel times the electrode height in meters. Then

$$\text{FEV1}(p) = (f_{\text{TLC}}(p) - f_{1\text{s}}(p)) * V_p$$

$$\text{FVC}(p) = (f_{\text{TLC}}(p) - f_{\text{EFE}}(p)) * V_p$$

and we can then approximate global FEV1%FVC for the patient by summing over all pixels in the lung region

$$\text{FEV1/FVC} = \frac{\Sigma_{i=1}^{N}\text{FEV1}(p)}{\Sigma_{i=1}^{N}\text{FVC}(p)}$$

where N is the number of pixels in the lung region.

This can be visualized by plotting the first component $\vec{v}$ and labeling each of the frames required. Note that we flip $\vec{v}$ vertically, since conductivity is inversely proportional to lung volume; as the air fills with lungs the overall conductivity in the thoracic cavity decreases. In this par-

ticular data set the approximate FEV1%FVC calculated from EIT data is 82.4% and the actual FEV1%FVC measured during spirometry was 85.3% (see Figure 2.8). We have a paper with further details and a statistical analysis of the method applied to 28 data sets, see [93].

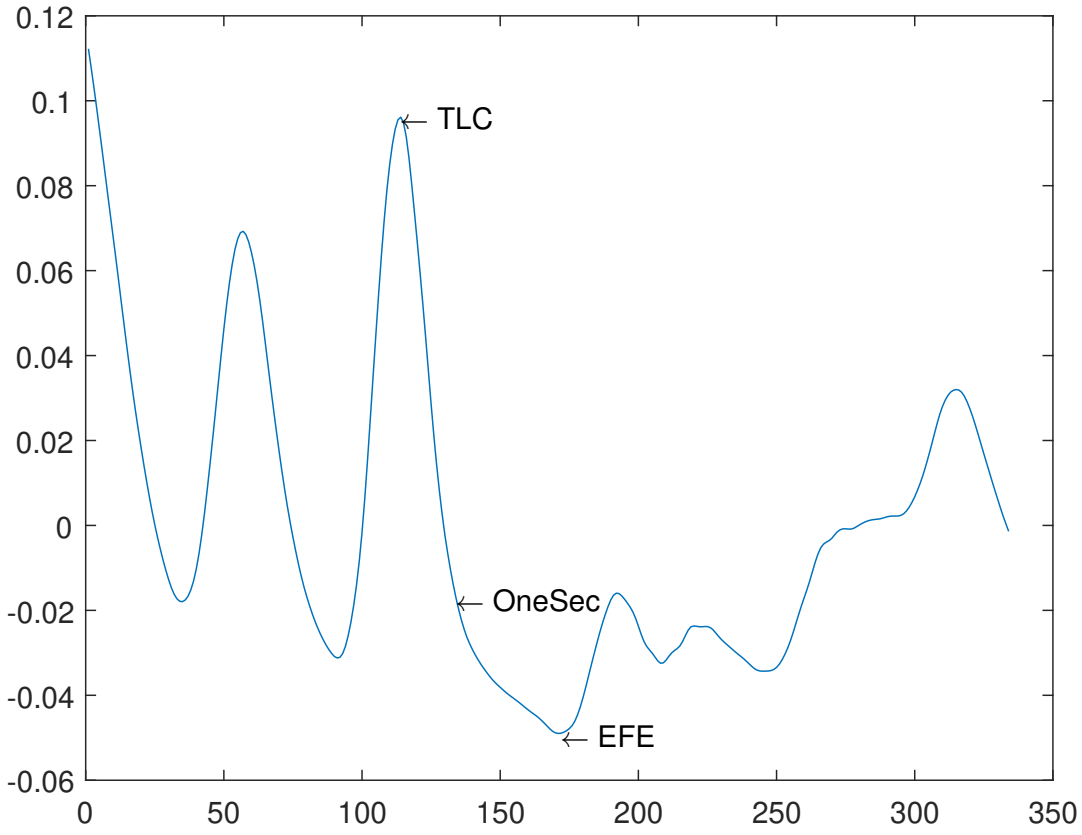


Figure 2.8: The first principal component of a spirometry data set with TLC, FEV1, and EFE labeled. The vertical axis is inverse relative conductivity and the horizontal axis is frames.

Chapter 3

INTERIOR BOUNDARY RECOVERY USING APRIORI INFORMATION

3.1 LOCATING ORGAN BOUNDARIES USING EIT DATA

While the D-bar method is a direct and powerful method for solving the inverse conductivity problem, some finer details on internal structures is lost due to the regularization arising from truncating the scattering transform acting as a low pass filter (necessary due to the presence of noise). This is a problem with any regularized reconstruction method. Previous approaches to identifying inhomogeneities without performing a full reconstruction include enclosure methods [55] and so-called "probing methods" [56], but none of these account for apriori knowledge of the internal structures of the domain. In this section we discuss results of [3] and the use of prior information in EIT reconstructions.

3.1.1 USING A PRIORI DATA IN RECONSTRUCTIONS

The goal here is to leverage prior knowledge about the conductivity distribution σ in our reconstructions. This knowledge can come from either assumptions about the conductivities or knowledge of the internal structure of the region being imaged (such as a CT scan showing the boundaries of organs). In [3] this is achieved by including a priori knowledge in the computation of the scattering transform, which we describe here to motivate our work in the next section.

First, let σ_{pr} be the *a priori* conductivity distribution and solve the forward problem, computing q_{pr} and $\psi_{pr}(z,k)$ by directly solving the Schrödinger equation (1.13) using the Lippmann-Schwinger equation (1.17). Then compute the scattering transform of the prior $\mathbf{t}_{pr}$ by

$$\mathbf{t}_{pr}(k) = \int_{\mathbb{R}^2} e^{i\bar{k}\bar{z}} q_{pr}(z)\psi_{pr}(z,k)dz, \tag{3.1}$$

where $q_{pr} = \frac{\Delta\sqrt{\sigma_{pr}}}{\sqrt{\sigma_{pr}}}$ (the Schrödinger potential) and $\psi_{pr}(z,k)$ the CGO solution corresponding to the prior. Then for noisy data Λ_σ^ϵ with noise level ϵ, an approximation to the scattering transform $\mathbf{t}^\epsilon_{R_1,R_2}$ that incorporates the prior can be given by [3]

$$\mathbf{t}^\epsilon_{R_1,R_2}(k) = \begin{cases} \int_{\partial\Omega} e^{i\bar{k}\bar{z}} \left(\Lambda_\sigma^\epsilon - \Lambda_1\right) \psi^\epsilon(z,k) ds(z), & |k| \le R_1(\epsilon) \\ \mathbf{t}_{pr}(k), & R_1(\epsilon) < |k| \le R_2 \\ 0, & |k| > R_2, \end{cases} \tag{3.2}$$

where $\psi^\epsilon(z,k)$ is the CGO solution corresponding to the data and is found via the boundary integral equation $\psi^\epsilon(\cdot,k)|_{\partial\Omega} = e^{ikz}|_{\partial\Omega} - \int_{\partial\Omega} G_k(z-\eta)(\Lambda_\sigma^\epsilon - \Lambda_1)\psi(\cdot,k)$.

This is a natural approximation to use since the scattering transform with noise needs to be truncated, and the noise-free $\mathbf{t}_{pr}$ can be truncated with a larger radius. This allows control over the influence of the prior via the secondary truncation radius R_2, with larger values allowing stronger influence from the prior.

Once $\mathbf{t}^\epsilon_{R_1,R_2}(k)$ has been computed, we can solve the integral form of the D-Bar equation while replacing the first term with [3]

$$\mu_{\text{int}}(z) := \frac{1}{\pi R_2^2} \int_{|k|\le R_2} \mu_{pr}(z,k) dk. \tag{3.3}$$

Then we can introduce a parameter α that gives further control over the influence of the prior, resulting in the equation

$$\mu^\epsilon_{R_2,\alpha}(z,k) = \alpha + (1-\alpha)\mu_{\text{int}}(z) + \frac{1}{(2\pi)^2} \int_{|k|\le R_2} \frac{\mathbf{t}^\epsilon_{R_1,R_2}(k')}{(k-k')\overline{k}'} e^{-i(k'z+\overline{k'}\overline{z})} \overline{\mu^\epsilon_{R_2,\alpha}(z,k')} dk' \tag{3.4}$$

Solving this allows the recovery of a conductivity distribution with influence from the prior conductivity distribution:

$$\sigma(z) = (\mu^\epsilon_{R_2,\alpha}(z,0))^2. \tag{3.5}$$

See [3] for examples of reconstructions performed using a prior conductivity distribution.

3.2 LOCATING INTERNAL BOUNDARIES VIA OPTIMIZATION

The goal of this chapter is to answer two questions: given some data Λ_σ, can we find an optimal prior distribution σ_{pr} to use in reconstructions without prior knowledge of organ boundary locations, can we perform a ‘‘sparse" approximation of a reconstruction using the prior, and can that prior distribution give us improved information about the internal structures of Ω?

We propose the following: given some data Λ_σ, and some initial prior conductivity distribution $\sigma_0(z)$ solve the minimization problem

$$\min_{\sigma_{pr}} \|\mu_{meas}(z,k)|_{\partial\Omega} - \mu_{pr}(z,k)|_{\partial\Omega}\|_2^2 + \alpha L(\sigma_{pr}) \tag{3.6}$$

where σ_{pr} is a piecewise constand conductivity distribution whose conductivity values and/or parameterized boundary is unknown, $\mu_{meas}(z,k)|_{\partial\Omega}$ is the CGO solution arising from the data (1.18), $\mu_{pr}(z,k)$ is the CGO solution arising from solving the Schrödinger equation (1.13) using the Lippman-Schwinger equation (1.17) with the current iterate of the conductivity distribution σ_{pr}, and $\alpha L(\sigma_{pr})$ is a penalty term with operator L with regularization parameter α acting on the prior distribution. Generally, σ_0 and σ_{pr} can be any conductivity distributions, but since we are attempting to extract the boundaries of organs, σ_0 and σ_{pr} will have parameterized organ boundaries and piecewise constant conductivities that get passed to the optimization routine.

The L_2 norm in Equation 3.6 is the squared sum of the difference of $\mu(z,k)|_{\partial\Omega}|$ and $\mu_{pr}(z,k)|_{\partial\Omega}$ for each $z \in \partial\Omega$ and $|k| < R$ where R is the truncation radius used to computed the scattering transform $t(k)$. In practice the boundary $\partial\Omega$ is discretized into 32 points since the ACE1 system [87] typically uses 32 electrodes. The implementation is straightforward:

$$\|\mu - \mu_{pr}\|_2^2 = \sum_{z\in\partial\Omega}\sum_{|k|<R}(\mu(z,k) - \mu_{pr}(z,k))^2. \tag{3.7}$$

Alternatively we can replace μ_{meas} and μ_{pr} in Equation 3.6 with the scattering transform t and t_{pr} or ψ and ψ_{pr} since the cost to compute them once μ_{meas} and μ_{pr} is computed is low.

Note that if we have good a priori conductivities for the organs but the organ boundaries are uncertain that we may fix the conductivities of the organs while allowing the organ boundaries to change during optimization. Thus the optimization is only over conductivity distributions with those fixed conductivities in the organ regions but the shapes of the organs are free. This approach is useful for testing against known distributions, but allowing the conductivities to also chage should result in a most robust approach.

We could consider σ_{pr} itself as an approximation to the full reconstruction. This has the advantage of being simple: interpretation is easy and no further processing is required. If we enforce sharp boundaries in the prior (if our prior is piecewise constant, for example) then this will also have sharp boundaries and may provide an approximate representation of organ boundaries. However, this method uses the data only indirectly (via $\mu(z,k)$ in (3.6)) and lacks the regularization provided by the truncated scattering transform.

Another option is computing μ_{int} (3.3). This is equivalent to setting $\alpha = 0$ in (3.4), and like σ_{pr} it will have sharp boundaries since it relies weakly upon the data.

Lastly, we can use the prior conductivities found in (3.6) and perform a full reconstruction with this prior via (3.4), choosing an $\alpha > 0$. This explicitly uses the data and takes advantage of regularization, although the boundaries found in the prior will be smoothed out in the reconstruction.

3.2.1 IMPLEMENTATION DETAILS

The first step of implementation is to partition the domain Ω into disjoint subsets $\Omega_1, \Omega_2, ..., \Omega_N$ and set constant conductivities $\sigma_1, \sigma_2, ..., \sigma_N$ where the conductivity on Ω_j is σ_j. In these results the conductivities are fixed, though future work will include optimizing the conductivities in each region concurrently with the boundaries.

Next is a choice of penalty term $L(\sigma_{pr})$. Several different terms were utilized during testing:

1. $L = 0$, no penalty term

2. $L = \dfrac{1}{\|v\|^2}$, where v is the vector of boundary points being optimized (encouraging the boundary to grow larger relative to the initial guess)

3. $L = \|v - v_0\|^2$ where v_0 is some default boundary locations (such as boundaries that represent typical heart and lung locations).

Many different algorithms were testing as well: interior-point, trust-region-reflective, sequential quadratic programming, quasi-Newton and trust-region, and simmulated annealing; these cover a large range of both constrained and unconstrained nonlinear optimization techniques. While some algorithms produced better results than others, computational performance with such a large number of variables was a problem for all of them, with simmulated annealing being the worst offender taking approximately 24 hours per run even in parallel with 32 cores.

I used an idealized forward solver; since the values of σ are known $q = \sigma^{-1/2}\nabla\sigma^{1/2}$ is directly computed after mollifying σ so that derivatives may be numerically computed. From q the Lippmann-Schwinger equation 1.17 $\mu = 1 - g_k * (q\mu)$ is solved directly for μ. From μ both ψ and $t(k)$ may also be computed directly. This solver is used to compute μ_{pr} from σ_{pr} during optimization.

μ_{meas} was computed using the D-bar method to solve the inverse problem. See [92] for an in-depth look at practical computation of solutions using the D-bar method.

All optimizations were performed using MATLAB and the MATLAB Optimization Toolbox [86].

3.3 RESULTS

This method has been tested on several data sets collected with a tank filled with saline, with a simulated heart and lungs made of agar (see Figure 3.1). Data was collected on Februrary 2nd 2016 using the ACE1 system [87] utilizing 32 electrodes and a "skip 0" current pattern.

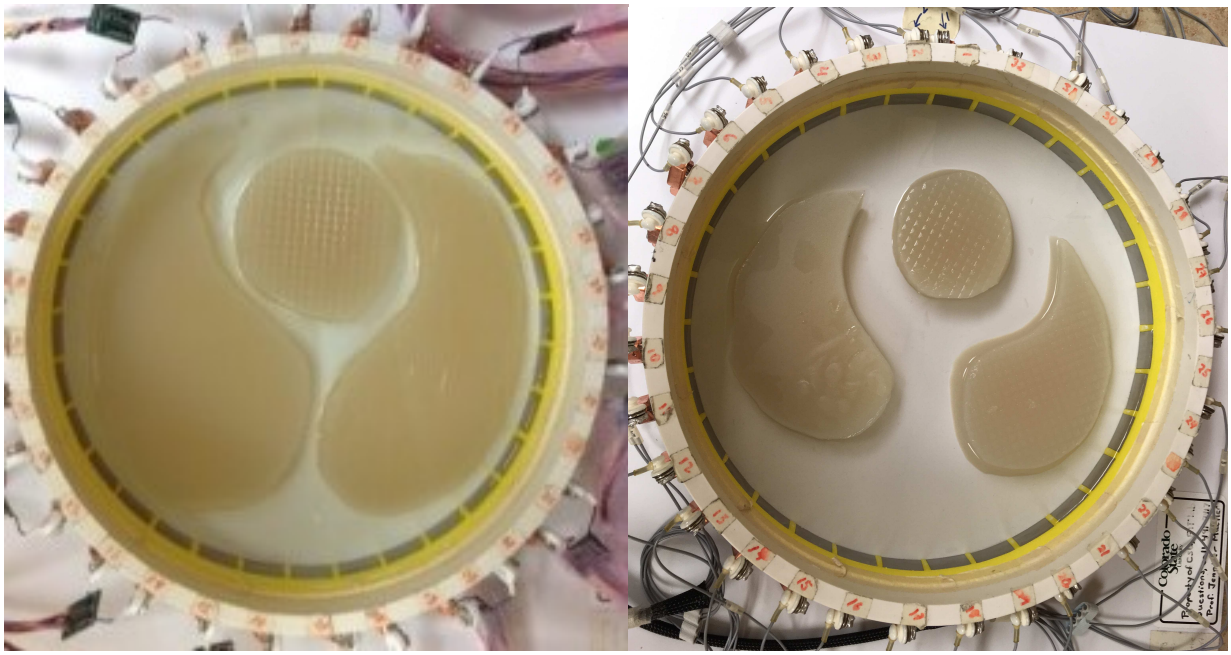


Figure 3.1: Tanks with simulated heart and lungs made of agar.

Using the agar data set, we split Ω into four regions: background, heart, right lung, and left lung. The boundaries of the heart and lung regions are discretized, and all points not contained in those regions are considered to be in the background (see Figure 3.2).

SIMULATED DATA

The boundaries from the tank data were used to create organ boundaries for simulated data. The conductivities of each organ were measured (at 190mS/m, 380mS/m, and 136mS/m for the background, heart, and lungs respectively) and the forward problem was solved for μ using the actual boundaries and conductivities. Since the forward solver (described in Section 3.2.1) was used to generate ground truth and the optimization depends on the forward solver, the error will be zero when σ_{pr} has the same boundaries as the true distribution. Indeed the optimization performed well, relatively speaking, when simulated data was used. While the boundaries are not perfect it clearly indicates that the method has merit.

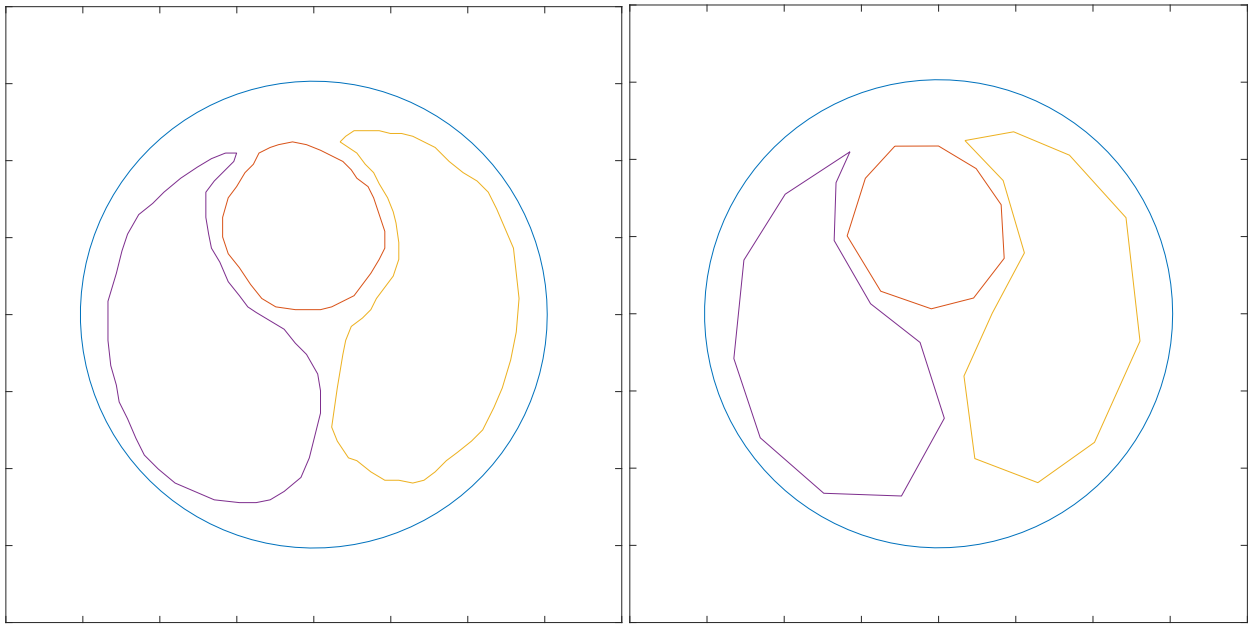


Figure 3.2: Full boundaries (left) with 35 points on the boundary of the heart, 58 and 51 points on the left and right lungs (resp.) and downsampled boundaries (right) with 10 points on the boundary of the heart and 12 points on the boundary of each lung.

3.3.1 IDENTICALLY SCALED BOUNDARIES

The first attempt at implementation on tank data was very simple: instead of letting each point on the boundary move independent of the other points we optimize over a single parameter that represents a scaling of the initial guess. That is, all points on the boundary are scaled by an identical factor, either shrinking or growing the organs uniformly. In this case the initial boundaries are a scaled versions of the actual, though one could use idealized lung and heart boundaries instead of the actual. The error function being minimized is Equation 3.6 with $L = 0$ and μ_{pr} is the result of solving the forward problem with the actual boundaries scaled by the scaling factor (and μ is the result of solving the inverse problem from the tank data).. Since the scaling factor is the only parameter being passed to the optimizer the number of points sampled on the boundary does not affect the speed of the optimization and the full boundaries (144 total points on the boundaries) may be used instead of downsampled boundaries.

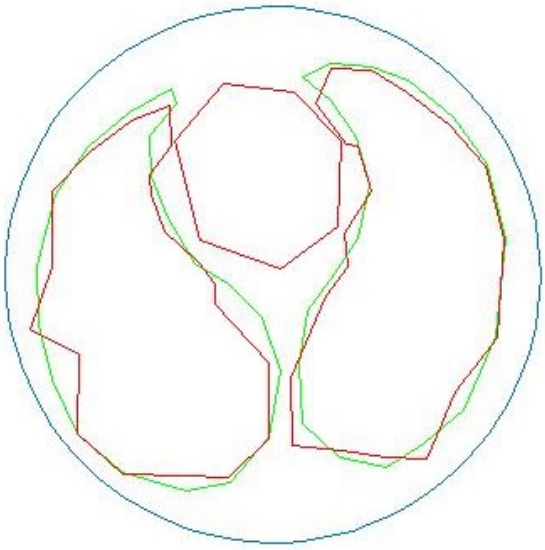


Figure 3.3: Results of optimizing boundaries using simulated data. Green is the actual, red is the result of the optimization.

This simplification works provided the initial boundaries are not scaled too far from the actual. Here we used $L = 0$ and the optimizer found an optimal scaling factor of 0.9946 (where a scaling factor of 1 is the actual). Plot of the result is omitted since it so closely matched the actual. Since the optimization routines are very fast for single variable error functions, I plotted the error vs scaling factor in Figure 3.4. The x-axis is the scaling factor (all boundaries scaled by that factor) and the y-axis is the value error function Equation 3.6 for the solution to the forward problem with the organs scaled by that factor. It's clear that if the initial scaling factor is below about 0.62 that a non-global optimizer is likely to converge to an even smaller scaling factor. There are also many local minimums that optimizers may get stuck in even with only one parameter, illustrating the nature of these types of optimizations.

Note that despite converging to the actual boundaries that the error function is still quite large with a value of 42 at the minimum. The likely cause for this is a mismatch between the results of solving the inverse problem from the data and the solution of the forward problem using the idealized forward solver described in Section 3.2.1. I also used the constrained optimization routine *fmincon* in MATLAB for the optimization with a maximum scaling factor of 1; without this

optimization it is possible that the optimization could have converged to boundaries larger than the actuals.

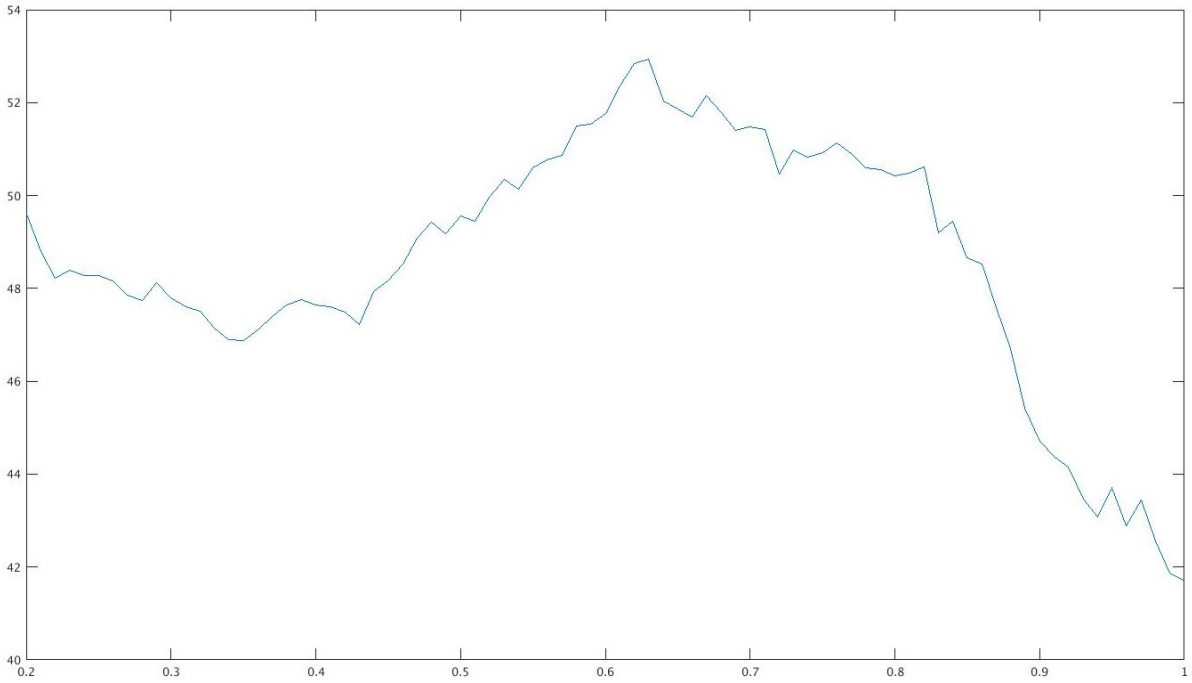


Figure 3.4: Scaling factor plotted against error in an identically scaled boundary optimization. The x-axis is scaling factor while the y-axis is the error function Equation 3.6 evaluated for the prior constructed using that scaling factor. Note that the error is still quite large even when the boundary scaling is 1 (i.e. scaled to the actual boundaries).

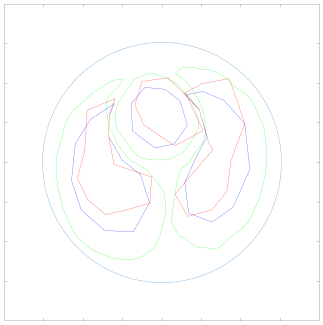
While this approach depends on an oversimplification of the problem it does provide a proof of concept for the method.

3.3.2 FREE BOUNDARIES

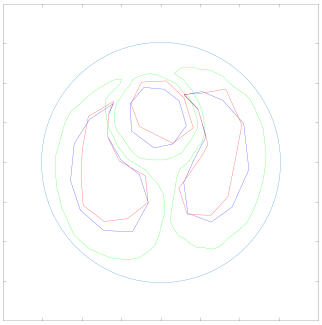
Since the identically scaled boundaries is a very unrealistic simplification, I needed to unlink points of the boundaries from each other. Instead of a single parameter representing the scale of every point on the boundaries, I allowed each point on the discretized and downsampled boundaries to be represented by its own parameter.

We can pass the full list of boundary points to an optimization routine in an attempt to find an optimal boundary, but even with parallelization a single minimization attempt using the highly nonlinear (3.6) with over 300 variables is prohibitively expensive to compute. We can improve this by greatly downsampling the boundary, trading spatial resolution between boundary points for speed in optimization (Figure 3.2). We can further reduce the number of variables by half by converting from Cartesian coordinates to polar coordinates (relative to the center of mass of each region), and passing only the radial coordinate, leaving the angle from the center fixed. Thus, each point on an internal boundary is represented by a single parameter.

Some results with the respective penalty terms and algorithms are presented in Figure 3.5:



Quasi-Newton with no penalty term.



Quasi-Newon with penalty term $200\frac{1}{\|v\|}$

Figure 3.5: Results using unconstrained optimization for various algorithms. The green boundaries are the actuals, the blues the initial boundary passed to the optimizer, and red the output of the optimization.

As we can see in Figure 3.5 the results are quite bad. There were many difficulties in optimizing free boundaries, not the least of which is the amount of time required to perform the optimization for each set of parameters (from 20 minutes to 24 hours depending on the algorithm and parameters). Additionally $\mu(z,k)$ and $\mu_{pr}(z,k)$ do not closely match even when ground truth boundary locations are used in the forward solver.

The results in Figure 3.5 used the following optimization parameters. The boundaries were discretized into 32 points total, 8 for the heart and 12 for each lung. Reducing the number of points on the boundaries of the organs reduces the time required for the optimization routine but

still maintains the general shape of the organs (especially the heart which is convex and quite regular). The organs were shrunk to 70% the original size and used as the start position.

With no penalty term the error started at 112.65 and converged to a minimum of 68.34 after 6 iterations using the Quasi-Newton algorithm in MATLAB, requiring 330 error function evaluations (i.e. the forward problem was solved 330 times) and taking 28 minutes. With a penalty term that encourages outward growth of the organs the error term started at 130.63 and converged to a minimum of 98.92 after 5 iterations using the Quasi-Newon algorithm in MATLAB, requiring 264 error function evaluations and taking approximately 25 minutes.

The upper limit before computing was set at 200 iterations but computations for most sets of hyperparameters converged long before getting close to that limit. Step and optimization tolerance were each set to 10^{-6}. Function evaluations for gradient computations were performed in parallel using 6 cores.

3.4 DISCUSSION

There are several likely reasons that this method has failed thus far. First is the parameterization of the boundaries: allowing each point on the boundary to move individually gives too many degrees of freedom for possible boundaries (with 150 points sampled on the boundaries gives 300 parameters to optimize). Realistic boundaries are quite regular and the optimization in its current state returns very irregular boundaries. The optimization landscape is too nonlinear for standard techniques to converge with this many parameters.

The second possible reason for failure to converge to the actual is the forward solver. The forward solver I used is very idealized: it assumes infinite precision on the boundary (so no electrode model is used) and idealized data. It also does not simulated voltages on electrodes or implement an electrode model. This speeds up the computations, but the output of the solver will not match data collected on an actual tank or human. In fact, when this forward solver is used with the actual conductivities and boundaries the error function is *larger* than the error function at the very incorrect boundaries output by the optimization (see 3.4. Ideally the forward solver would accurately

model real world data collection and the error function would be close to 0 when the actual conductivities and boundaries are used. Using a more realistic forward solver may increase accuracy and improve the results, but the already hefty computations will take longer. The positive results using simulated data strongly imply that the method will work with a more realistic forward solver.

Despite these methods not working so far I believe that with modifications that it is still a viable approach. A method that further reduces the number of variables and maintains reasonable boundary shapes at every step is much more likely to succeed, though I have no proposals at the moment for how this might be achieved.

Chapter 4

INTERIOR BOUNDARY RECOVERY USING DEEP LEARNING

In this chapter I will explore the application of deep learning to the problem of finding internal boundary structures from EIT data. Deep learning provides tools for approximating nonlinear functions by training an artificial neural network on known input/output pairs with the goal of having the network generate correct output values for input data that it was not trained on. I will present results from simulated data and data collected on a tank that contains a simulated heart and lungs made of agar. Computational details are provided, and we conclude with some ideas for future work.

4.1 INTRODUCTION

The idea here is straightforward: we want to recover the map that sends a scattering transform computed from the measured DN map to the 2D curves that define the boundaries of internal structures. We do this by training a deep neural network on labeled pairs of scattering transforms and the boundaries of the organs in the data those transforms were computed from. This allows the network to "learn" the nonlinear mapping between them by minimizing the error between the output of the network and known actual boundaries.

4.2 NETWORK INPUTS AND OUTPUTS

We use the scattering transform so that we can leverage the power of convolutional neural networks. These networks are ideal for processing images because the convolution operates on spatially local information, i.e. the value of a node is determined by the values of a specific pixel in the previous layer and the pixels surrounding it, with the weights in the network corresponding to the weights in a discrete convolution or filter. Because the scattering transform is complex-valued

and, in practice, evaluated on a grid in $\mathbb{C}$, we treat the real and imaginary parts as two separate channels in an image. That is, since we evaluate the truncated scattering transform on a (typically square) two dimensional grid $k_{p,q} = x_p + iy_q \in \mathbb{C}_{|z|<R}$, we let the grid points correspond to pixels and form two images by computing $t(k_{p,q})$ and separating the real and imaginary parts, $\mathrm{Re}(t(k_{p,q}))$ and $\mathrm{Im}(t(k_{p,q}))$, letting their values on the grid represent intensities in grayscale images.

The corresponding outputs are curves in $\mathbb{R}^2$ defined by a sequence of points (x_i, y_i). These are vectorized as $(x_1, y_1, x_2, y_2, ..., x_n, y_n)$ since the output of the network is a vector (noting that the length of the vector is $2n$ for the n points sampled on the internal boundary). The network is then trained by reducing the mean-squared error between the output of the network for a given input and the truth. I used a sampling of 128 points on the internal boundaries so the output is a vector in $\mathbb{R}^{256}$. Note here that the output does *not* contain information about the values of the conductivities in each region, only the locations of the boundaries themselves (though the network is trained on scattering transforms corresponding to distributions that are constant in each region).

So the input to the network is a 2-channel 16×16 grayscale image (the real and imaginary parts of the truncated scattering transform) and the output is a real-valued vector in $\mathbb{R}^{256}$; thus the network will learn functions $F : \mathbb{R}^{512} \to \mathbb{R}^{256}$.

4.3 NETWORK ARCHITECTURE

For this application I used a fairly simple convolutional neural network. In the training data $t(k)$ was evaluated on a 16×16 grid, so the input layer consisted of two channels (one channel for the real part, one channel for the imaginary part) 16×16 images padded with zeros on the outside so that the output of the first convolutional layer was still 16×16 images. The first two convolutional layers contained 32 5×5 filters with a relu activation function followed by a 2×2 max-pool layer (outputs are 8×8 images) and 50% dropout. Similarly, the third and fourth layers are convolutional layers with 64 3×3 filters with a relu activation function followed by another 2×2 max-pool layer and 50% dropout resulting in 64 4×4 blocks.

After the convolutional and max-pool layers is a densely connected 1024 node layer, followed by another 512 densely connected layer. The output layer is also densely connected and has 104 nodes and is the output vector $(x_1, y_1, x_2, y_2, ..., x_n, y_n)$ where (x_i, y_i) represents an approximate point on an internal boundary. See figure 4.1 for a graphical view of the architecture.

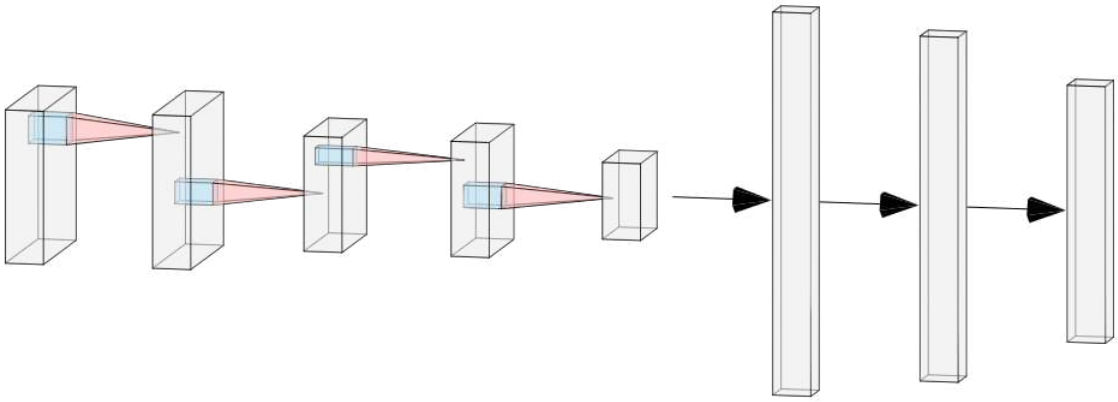


Figure 4.1: The network architecture. Blue boxes represent either a multiply/sum operation with a filter or a max pool on that region, with the output going to the next layer indicated by the red cone. Black arrows represent a densely connected layer. The layers are: convolution, max-pool, convolution, max-pool, and dense layers (noting that in the actual network I used the number of convolutional layers is doubled).

4.4 THE TRAINING DATA

The inputs to the network are 2-channel grayscale images consisting of the real and imaginary parts of the truncated scattering transform $t(k)$ while the outputs are a vector of the form $(x_1, y_1, x_2, y_2, ..., x_n, y_n)$ where (x_i, y_j) are the i-th internal boundary coordinate (assuming the internal boundaries have been discretized into n points). The inputs and outputs have all been scaled to be between -1 and 1 (the inputs being divided by the absolute value of the max value in the training set, and the size of the circular boundary reduced to radius 1) since it has been shown that this increases speed of convergence [79].

I simulated 100,000 thoracic cavity data sets by solving the forward problem for simulated heart and lungs in a circular domain with known conductivity values. Two sets of organ boundaries (see Figure 4.2) were used to simulate the data with each serving as a base 50% of the time, chosen at random. That is, each simulated distribution varies randomly from one of these two sets of boundaries. A numerically simulated conductivity distribution was randomized for each simulation by independently scaling each organ in the (randomly chosen) base by 55% to 105%. Additionally, each organ had a 25% chance of having a simulated injury where the bottom 20% to 75% of the organ was removed.

The background conductivity was fixed at $1\,S/m$, the heart was $2\,S/m$, and the lungs $0.5\,S/m$. These conductivities were each randomly scaled by adding between plus/minus 25% of the value of each conductivity. So the heart ranged from 1.5 to $2.5\,S/m$ and each lung was 0.375 to $0.625\,S/m$ (with each lung randomized independently). Since we assume twice differentiability on σ when computing the forward problem I also randomized the amount of mollification applied to the conductivities before solving the forward problem.

The forward solver used computes the Schrödinger potential $q(x) = \sigma^{-1/2}\Delta\sigma^{1/2}$, computes Faddeev's fundamental solution g_k, solves the Lippmann-Schwinger equation for μ (see Equation 1.17), and computes the scattering transform $t(k)$ directly (see Equation 1.22).

A k-grid size of 16×16 centered at the origin with a random truncation radius between four and eight (inclusive) was used. The forward problem was then solved for the scattering transforms corresponding to these randomized conductivity distributions.

4.5 RESULTS

The network was trained on $100,000$ randomly generated samples described above using Keras [24] with the TensorFlow backend [118] on an Nvidia RTX 2080 GPU. I trained the network for 2000 iterations (also known as epochs), taking approximately 2.5 hours. Optimzation was performed using AdaDelta [109], a stochastic gradient descent algorithm featuring a per-parameter learning rate (that is, each weight in the network has its own associated learning rate). Loss (in this

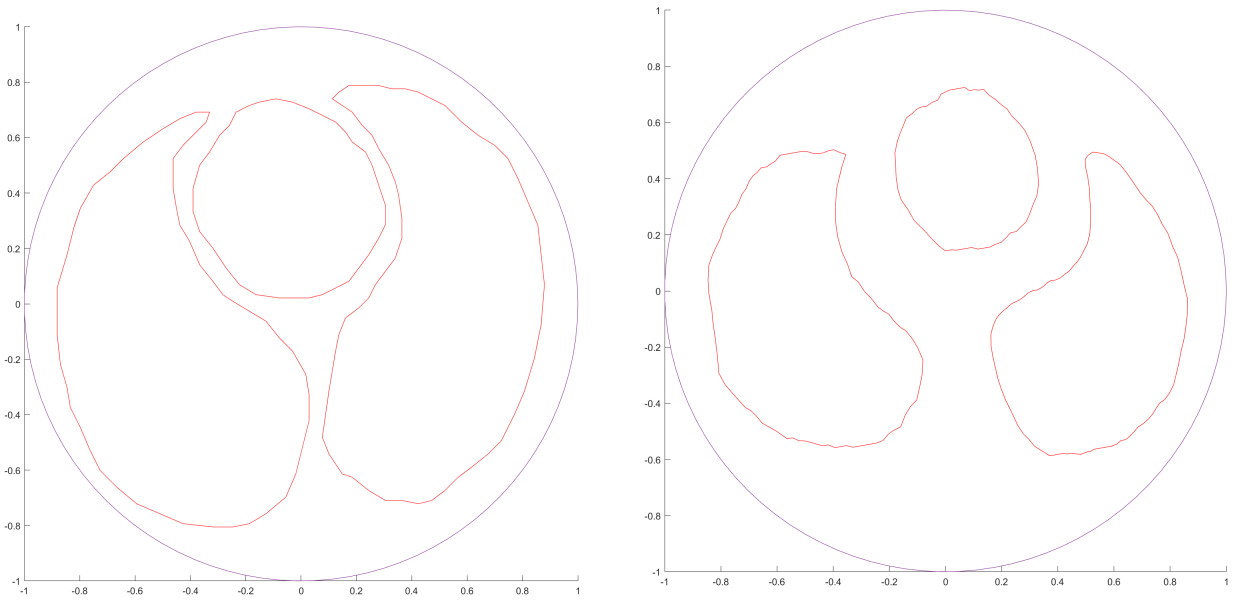


Figure 4.2: The internal boundaries of the two base conductivity distributions used to simulated the training data.

case the mean squared error between the output of the network and the actual) was tracked during training reaching a low of $9.7 * 10^{-4}$ on the test/validation set (see Figure 4.3). Note that loss is lower on the test set since the training set is subjected to dropout layers while the test set is not. Loss rapidly drops from 0.02 to 0.0025, but takes quite a few iterations to reach the order of 10^{-4}.

4.5.1 SIMULATED DATA

I applied the network to 50 scattering transforms that were not used in training and the results are quite clear: the network learned the map between scattering transforms and the internal boundaries that produce those transforms. The network performs a near perfect reconstruction of the boundaries on sets without injuries (see Figures 4.4a and 4.5d for example) and the reconstructions of simulated injuries described below are quite good.

This indicates that the complicated relationship between internal organ boundaries and the corresponding scattering transforms may be learned by neural networks. This is very promising because D-bar reconstructions are smoothed out by the regularization that truncating the scattering

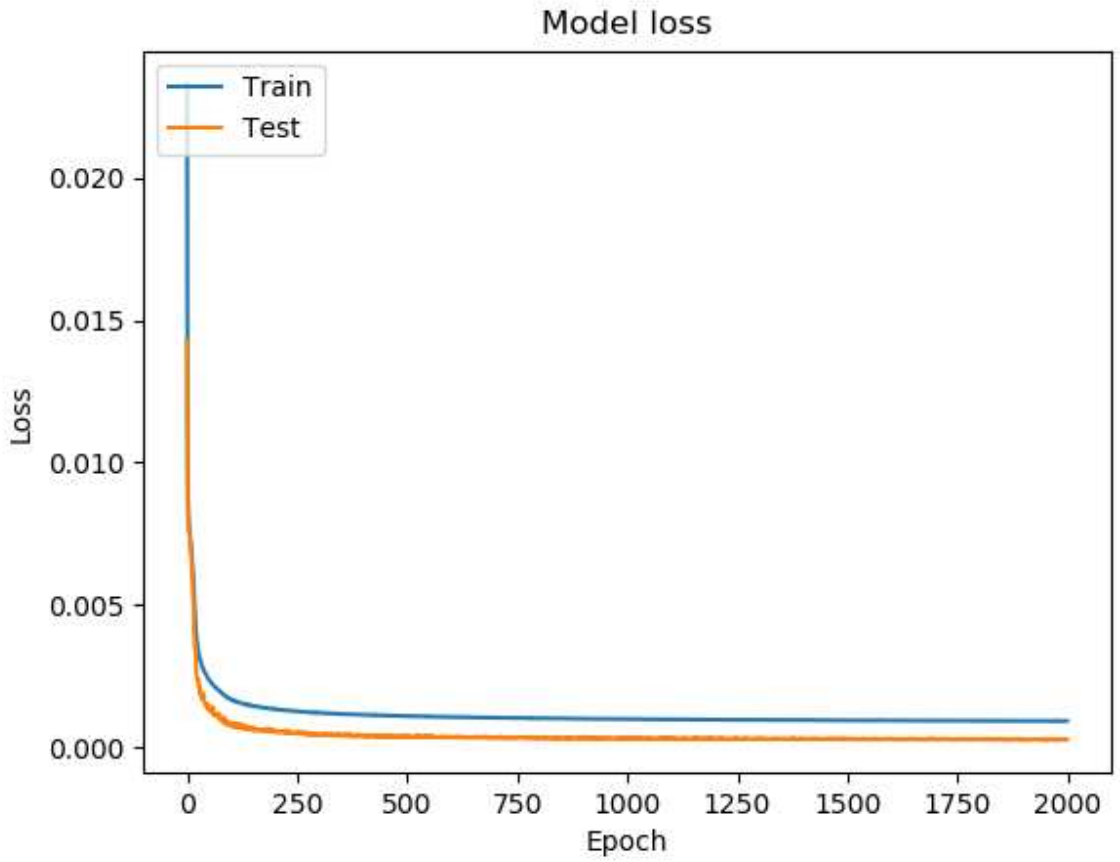


Figure 4.3: Loss plotted against iteration number. The minimum loss achieved was $9.7 * 10^{-4}$.

transform provides, leading to loss of exact inclusion boundary locations. The results in Figures 4.4, 4.5, 4.6 show that given sufficient and varied training data that reconstructions of boundaries can be performed by neural networks.

4.5.2 MULTIPLE SIMULATED INJURIES

While the test data performed very well I also wanted to know how well the network generalized and how well it learned individual features, such as left vs right lung injuries. In order to test this I simulated conductivity distributions with the same parameters as the training data but with two simulated organ injuries instead of one or zero. This is an important diference because the scattering transforms generated from these likely lie outside the space of scattering transforms that the network was trained on.

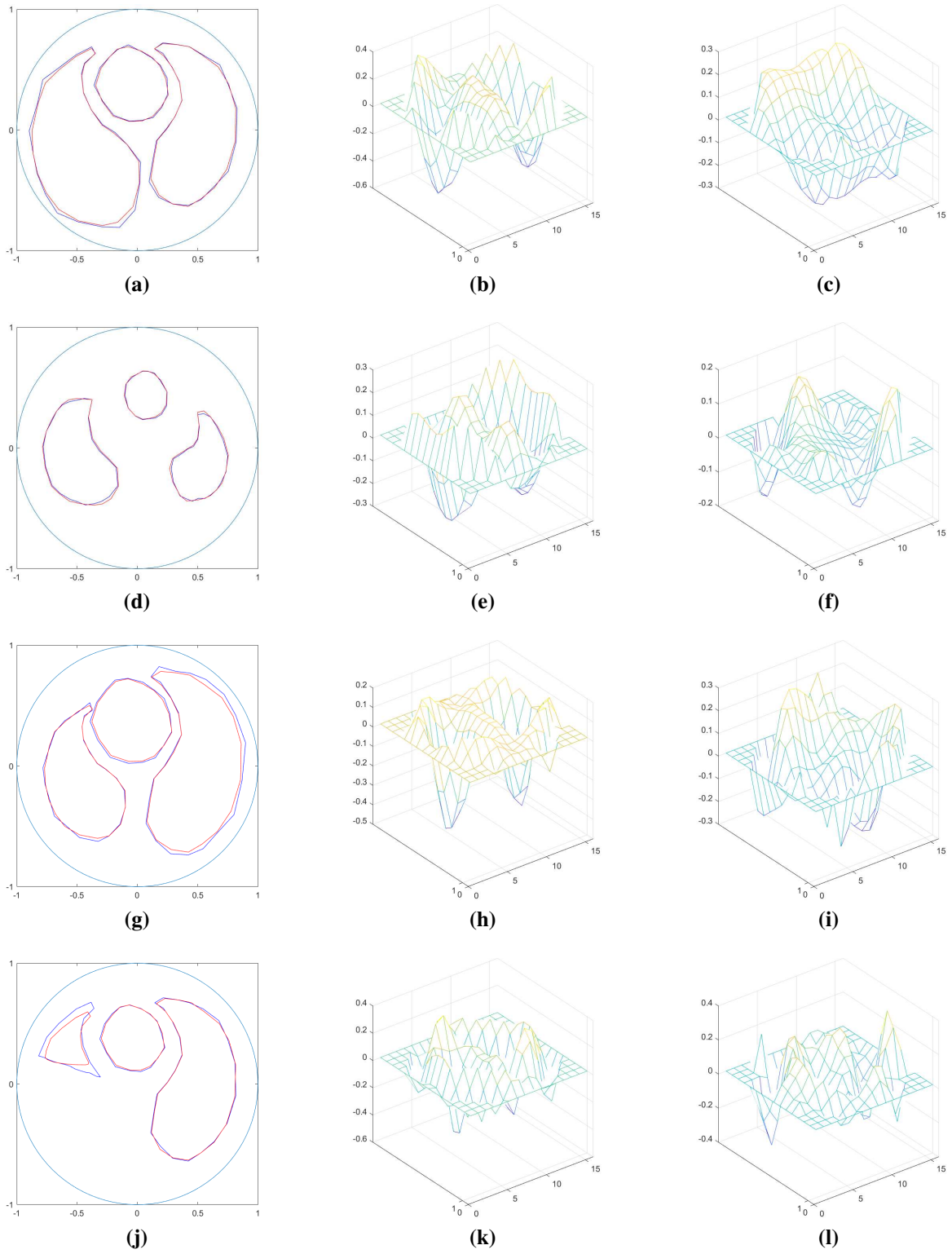


Figure 4.4: These are results from numerically simulated data passed through the network. The first column are outputs of the neural network (red) with the actual boundaries (blue). The middle column is the real part of the input scattering transform $t(k)$ computed using the red boundaries in the first column while the right column is the imaginary part.

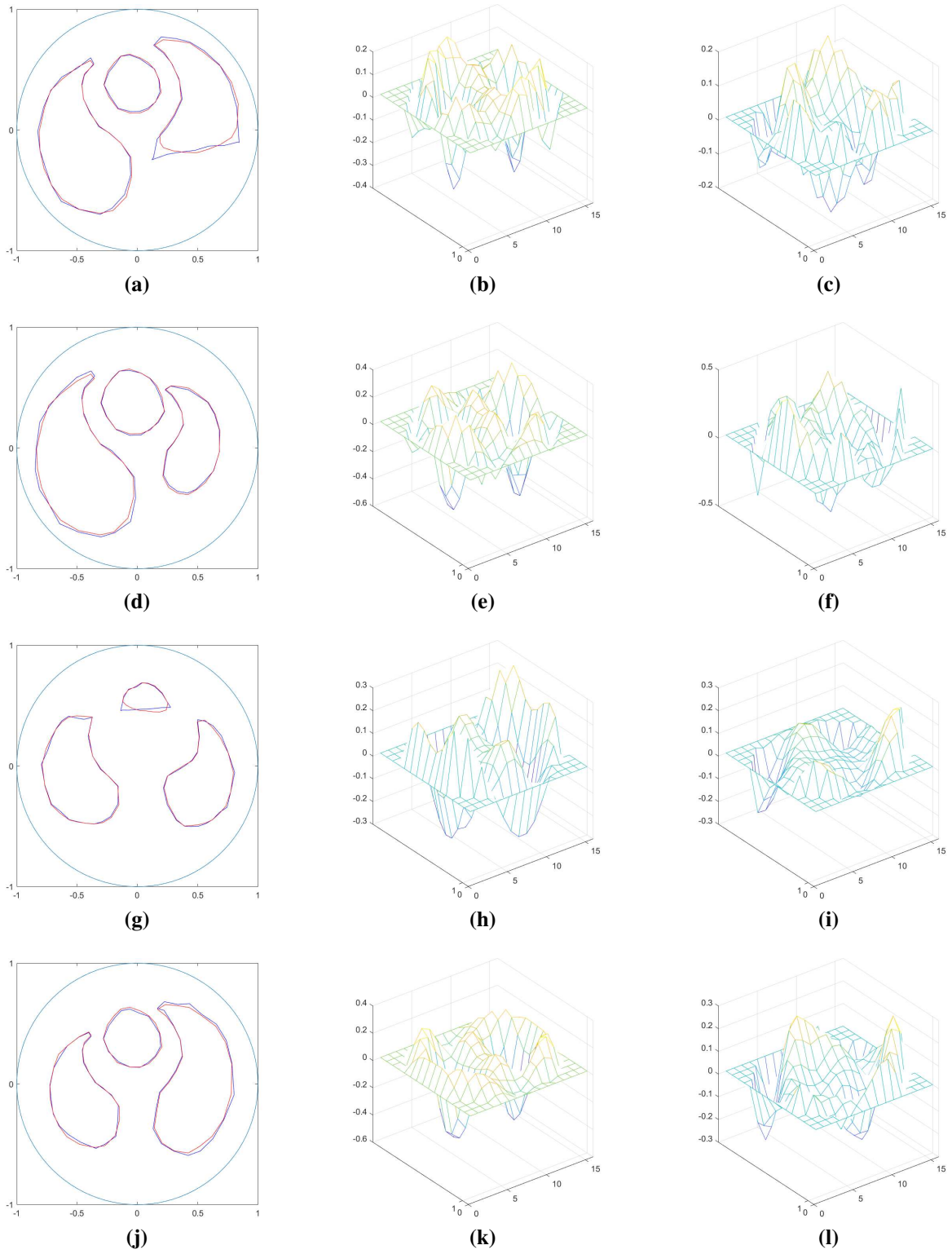


Figure 4.5: These are results from numerically simulated data passed through the network. The first column are outputs of the neural network (red) with the actual boundaries (blue). The middle column is the real part of the input scattering transform $t(k)$ computed using the red boundaries in the first column while the right column is the imaginary part.

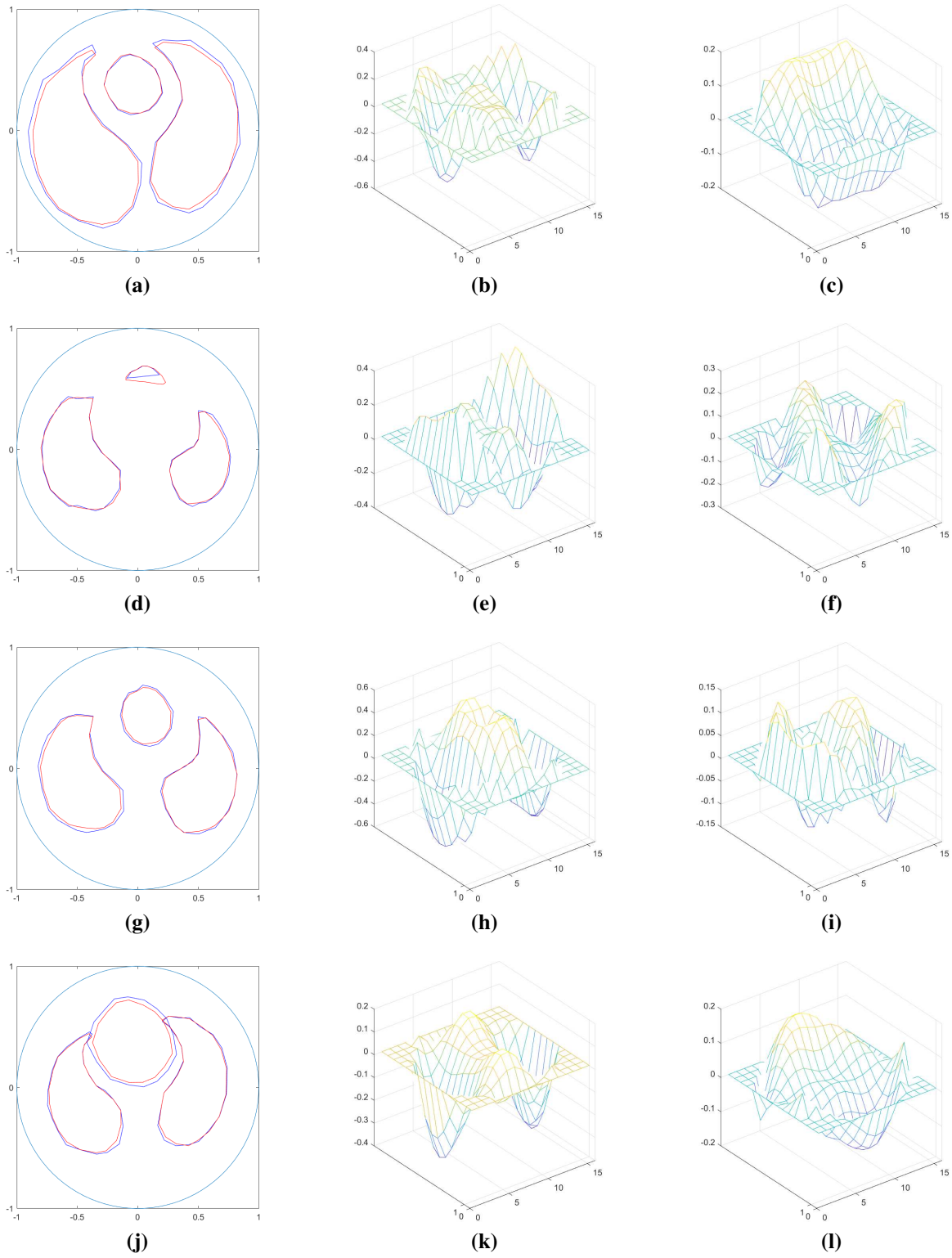


Figure 4.6: These are results from numerically simulated data passed through the network. The first column are outputs of the neural network (red) with the actual boundaries (blue). The middle column is the real part of the input scattering transform $t(k)$ computed using the red boundaries in the first column while the right column is the imaginary part.

As we can see in 4.7 results are mixed. While some boundaries are very close, the network did not predict any distributions with multiple simulated injuries. In some cases it predicted a small boundary when the actual boundary represented a simulated injury, such as 4.7c. Some cases it got completely wrong, predicting no organ injuries at all and not predicting small organs that approximately match the injuries (see Figure 4.7e for example).

This is not surprising since it never encountered double injuries during training, but it does imply that the network did not learn the features of the scattering cooresponding to the injuries in an independent manner. That is, we cannot combine features in a way not present in the training data and expect the network to perform well. Overall the results here are slightly disappointing, but not unexpected.

4.5.3 TANK DATA

I also tested the trained network on data collected on a tank with a heart and lungs simulated with agar (the same tank set used in Chapter 3, see Section 3.3 for details). I generated the network inputs by solving the inverse problem for the truncated scattering transform (using the boundary integral equation), and then forward fed the transform through the network described in Section 4.5 to get a predicted set of organ boundaries. The hope is that the training set covers the part of the space of scattering transforms from real data live in and that the network is able to return an accurate reconstruction of the real boundaries.

The results are mixed. On one set taken on the ACE1 system the results are fairly close, with only the left lung being predicted slightly smaller than the actual, see Figure 4.8.

4.5.4 A MORE GENERAL TRAINING SET

The results in Figures 4.7 clearly indicate that the training set is not robust enough since it does not include multiple organ injuries and rotations of the organs about the origin. The hope is that the more diverse the training set is the higher the chance the network has of generalizing and performing well on data that it has not seen before.

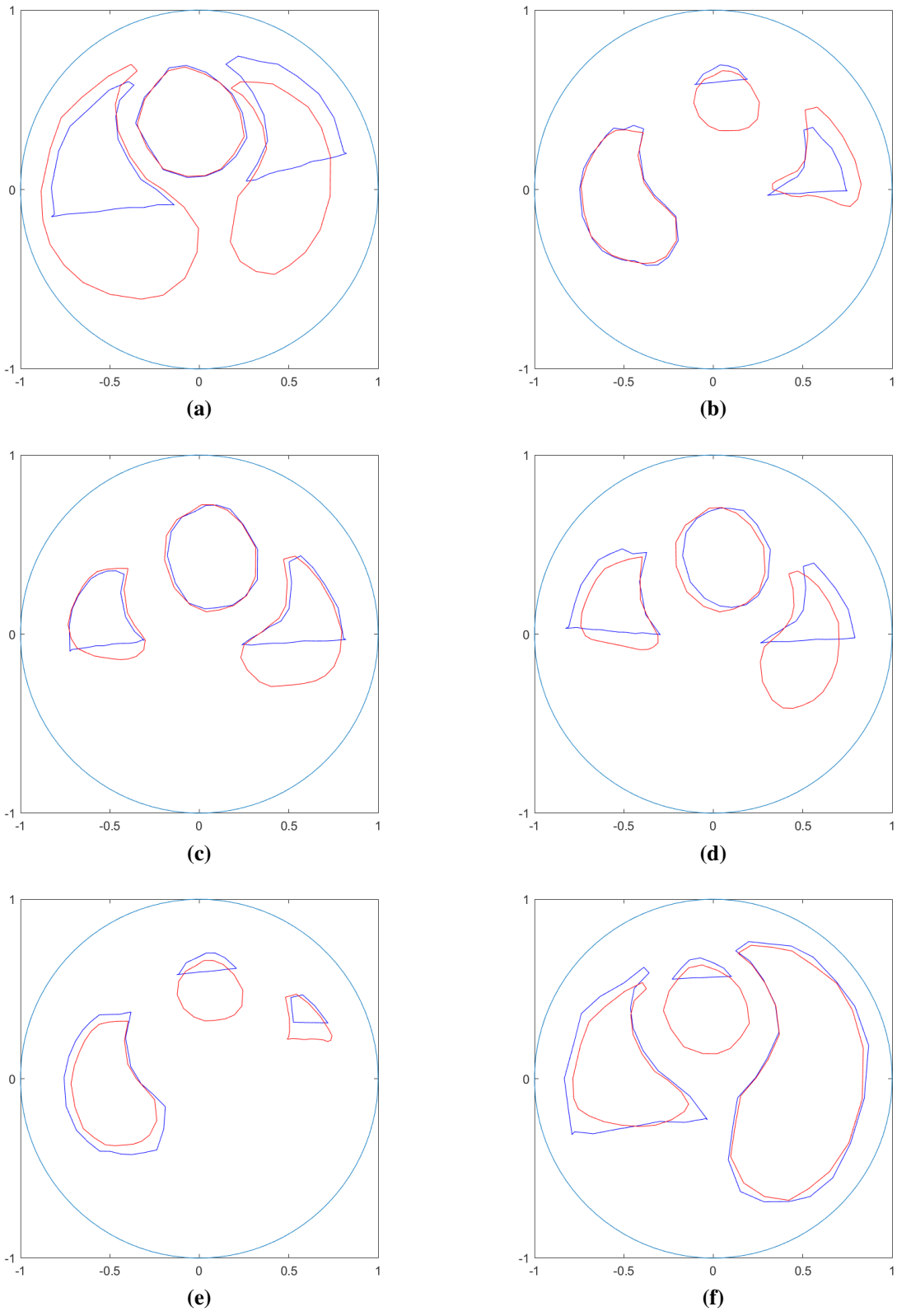


Figure 4.7: Results for multiple simulated injuries. The actual boundaries are in blue, with the network predicted boundaries in red.

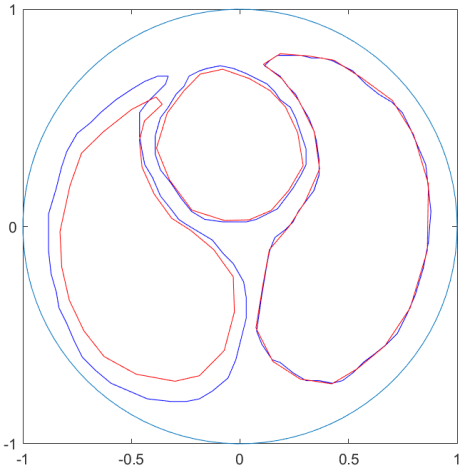


Figure 4.8: Actual vs predicted boundaries on a tank dataset taken with the ACE1 system. Actual boundaries are in blue with the network predicted boundaries in red.

So I created a new trianing data set with the same parameters as before but with the following modifications:

1. The base distribution may be rotated $\pm 30^{\circ}$ around the origin.

2. Each organ independently has a 25% chance of having a simulated injury instead of each sample having a 25% chance of having a single injury.

3. The base conductivities are randomized to up to $\pm 50\%$ instead of $\pm 25\%$.

The same network was again trained on 100,000 samples of numerically simulated data for 2000 epochs using AdaDelta and randomized batches of 100, reaching a loss of $8.9 * 10^{-4}$. The results on the test set are quite good, though not as accurate as the initial network. However, the space of scattering transforms that it can predict boundaries for is much larger than the previous network. See Figures 4.9 and 4.10.

As we can see there are many more configurations that the boundaries can take once we allow rotations and multiple injuries. Doubling the range of the conductivities additionally increases the size of the sample space by 100%. In this situation I believe that 100,000 samples is not enough to adequately represent the sample space and that increasing the number of samples could greatly increase the accuracy.

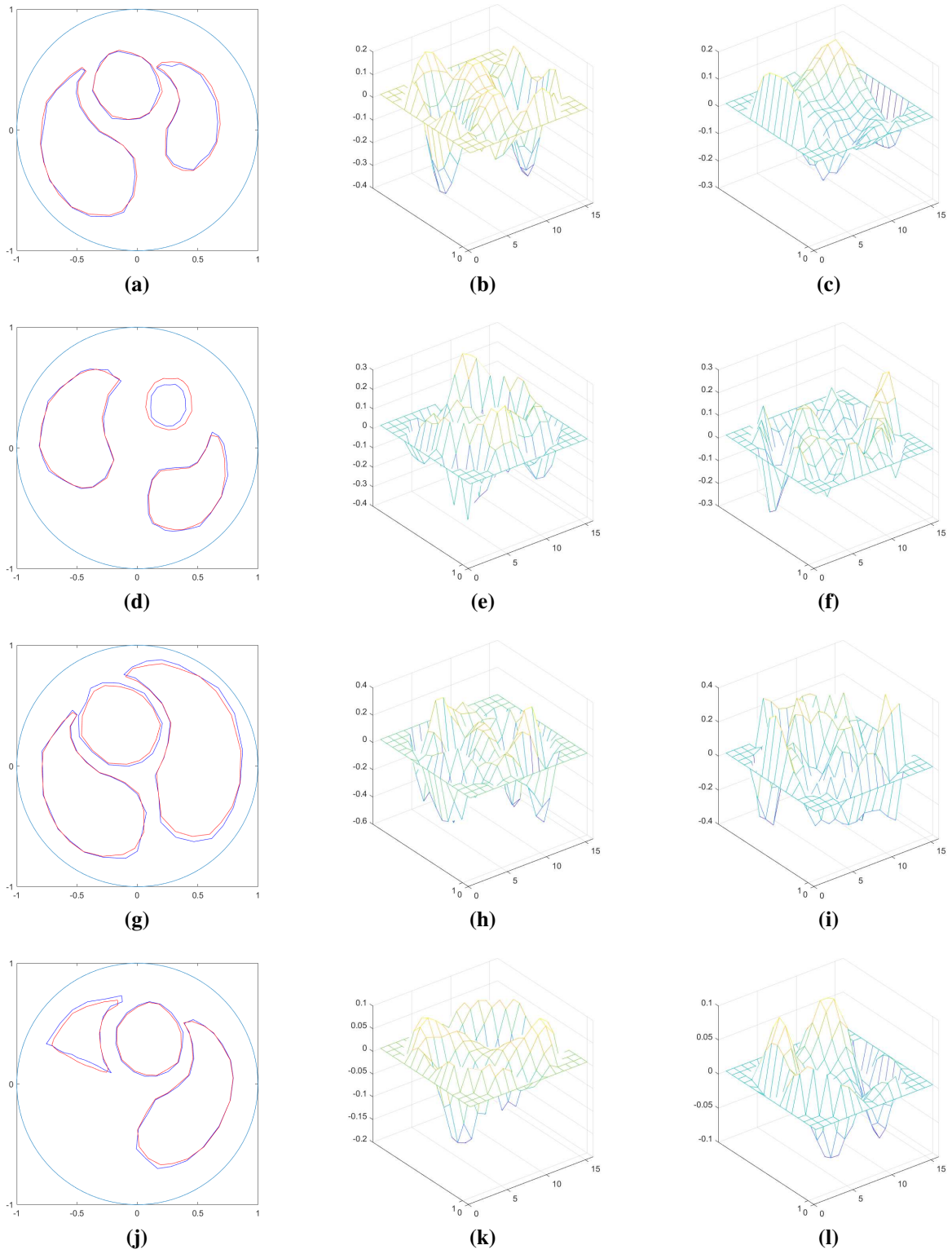


Figure 4.9: Results from the network trained with more general training data. The first column are outputs of the neural network (red) with the actual boundaries (blue). The middle column is the real part of the input scattering transform $t(k)$ computed using the red boundaries in the first column while the right column is the imaginary part.

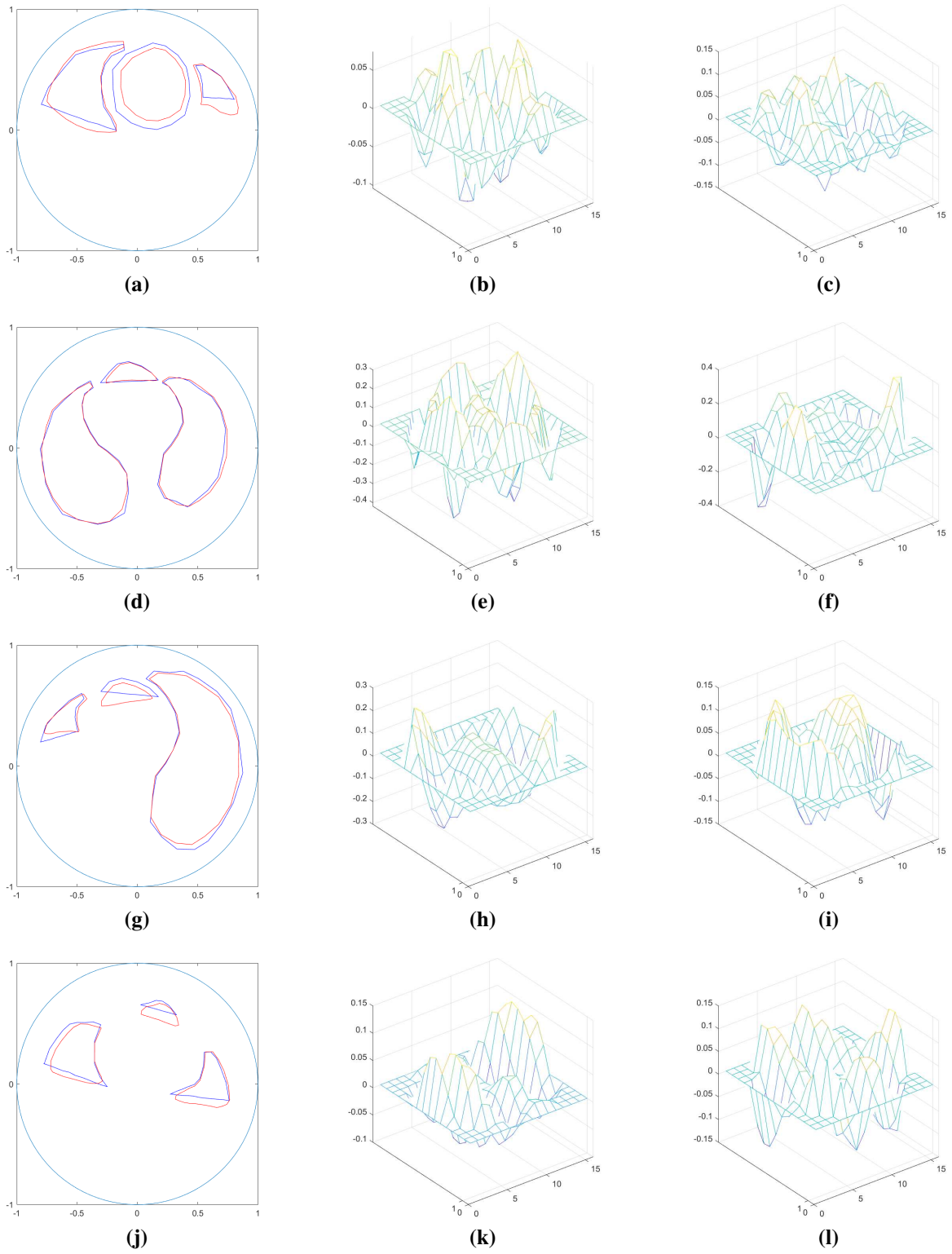


Figure 4.10: Results from the network trained with more general training data. The first column are outputs of the neural network (red) with the actual boundaries (blue). The middle column is the real part of the input scattering transform $t(k)$ computed using the red boundaries in the first column while the right column is the imaginary part.

This network clearly does better on multiple simulated injuries since they were included in the training data. The deficiencies in Figure 4.7 have, for the most part, been remedied in Figures 4.9 and 4.10.

The network prediction for the ACE1 tank data similarly improved and was very accurate (see Figure 4.11). I believe this indicates that the scattering transform from that data set is well-represented in the training data and that the network will produce accurate predictions for similar sets.

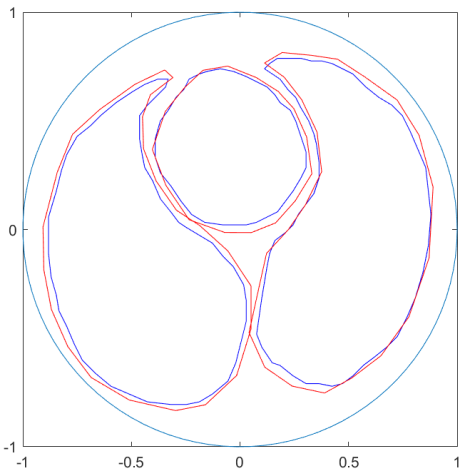


Figure 4.11: Actual vs predicted boundaries on a tank dataset taken with the ACE1 system. Actual boundaries are in blue with the network predicted boundaries in red. The network uses was trained with the expanded data set that includes rotated boundaries.

Training and network hyperparameters may also play a role. Maybe a larger network (or smaller network) is more appropriate, or maybe the level of dropout needs to be tuned. Since these hyperparameters are, in general, ad hoc, it is difficult to find optimal parameters without trial and error. Given that training takes approximately 3 hours and analyzing results isn't trivial, tuning the hyperparameters can be a daunting task.

Overall the results are very promising. They show that convolutional neural networks are well-suited to the task of performing certain types of EIT reconstructions and that they may be applicable to real data sets. There are many possible variations on this particular experiment left to be done as well, such as using functions other than the scattering transform as the input or performing full

reconstructions of more complicated conductivity distributions. The last one in particular I believe to be a very reasonable task, though network training time will play a factor in its practicality.

With sufficient advancement and refinement these networks could one day be used to provide near-instant approximate EIT reconstructions.

Chapter 5

CONCLUSION

We have described three advancements in the field of electrical impedance tomography: application of signal separation techniques to clinical EIT data, recovery of internal boundaries using optimization in the D-bar method, and an application of deep neural networks applied to scattering transforms in the D-bar method to the recovery of internal boundaries.

In Chapter 2 we applied signal separation techniques that leverage principal component analysis to split EIT data sets into two sets, one containing cardiac information and one containing ventilation information. These techniques were applied to tidal breathing sets, breath holding sets, and sets collected during pulmonary function tests. We successfully applied these techniques and were able to leverage them in the process of recovering clinical pulmonary measures and the detection of regions of air trapping.

In Chapter 3 we explored the recovery of the boundaries of internal structures using optimization applied in the D-bar method. Various error functions and optimization routines were used with a wide variety of optimization parameters. The success of the method under simplifying assumptions provides a proof of concept for the method and we believe that the method will be much more successful with a forward solver that more accurately simulates data collected on tanks containing simulated organs.

In Chapter 4 we showed that deep neural networks can successfully learn the mapping between scattering transforms and the internal boundaries of structures. A network was trained on randomly generated numerically simulated data using Keras with a Tensorflow backend. The network was then tested on numerically simulated data not used in the training set and the network successfully reconstructed the internal boundaries. While the network failed to predict boundaries of data collected on a tank with simulated heart and lungs made of agar, we believe that training data generated with a forward solver that accurately simulates tank data could be used to train a network capable of predicting boundaries within data sets collected on a tank.

Bibliography

[1] A. Adler, B. Grychtol, R. Bayford. Why is EIT so hard, and what are we doing about it. *Physiological measurement*, 36(6):1067, 2015.

[2] G. Alessandrini. Stable determination of conductivity by boundary measurements. *Applicable Analysis*, 27(1–3):153–172,1988.

[3] M. Alsaker and J. L. Mueller. A D-Bar Algorithm with A Priori Information for 2-Dimensional Electrical Impedance Tomography *SIAM Journal of Imaging*, Vol 9 No.4 pp. 1619-1654, 2016.

[4] D. Andreuccetti, R. Fossi, and C. Petruccii. Calculation of the dielectric properties of body tissues in the frequency range 10 Hz - 100 GHz. http://niremf.ifac.cnr.it/tissprop/htmlclie/htmlclie.php, Accessed: 2017-11-27

[5] K. Astala and L. Päivärinta. Calderón's inverse conductivity problem in the plane. *Annals of Mathematics*, 163(1):265–299, 2006.

[6] D.C. Barber and B.H. Brown. Applied potential tomography. *Journal of physics. E. Scientific instruments*, 17(9):723–733, 1984.

[7] D. C. Barber and B. H. Brown. Progress in electrical impedance tomography. *Inverse problems in partial differential equations* 151 – 64, 1990.

[8] J. Bikowski and J.L. Mueller. 2D EIT reconstructions using Calderóns method. *Inverse Problems and Imaging*, 2(1):43–61, 2008.

[9] G. T. Bolton, K. M. Primrose. An Overview of electrical tomographic measurements in pharmaceutical and related application areas. *AAPS PharmSciTech* 6(2):E137–E143.

[10] L. Borcea. A nonlinear multigrid for imaging electrical conductivity and permittivity at low frequency. *Inverse Problems*, 17(2):329–359, 2001.

[11] L. Borcea, J. Berryman, G. Papanicolaou. High-contrast impedance tomography *Inverse Problems*, 12(6):835-858, 1996.

[12] G. Boverman, T.J. Kao, D. Isaacson, and G. Saulnier. An implementation of Calderón's method for 3-D limited-view EIT. *Medical Imaging, IEEE Transactions on*, 28(7):1073–1082, 2009.

[13] G. B. Boverman, T. J. Kao, R. Kulkarni, B. S. Kim, D. Isaacson, G. Saulnier, J. C. Newell. Robust linearized image recontruction for multifrequency EIT of the breast. *Medical Imaging, IEEE Transcations on*, 27(10):1439–1448, 2008.

[14] B. H. Brown. Electrical impedance tomography (EIT): a review. *Journal of Medical Engineering & Technology* 27(3):97 –108, 2003.

[15] Uniqueness in the inverse conductivity problem for non-smooth conductivities in two dimensions. *Communications in Partial DifferentialEquations*, 22(5–6):1009–1027, 1997.

[16] A. P. Calderón. On an inverse boundary value problem. In *Seminar on Numerical Analysis and its Applications to Continuum Physics*, pages 65–73. Soc. Brasileira de Matemàtica, 1980

[17] M. Cheney, D. Isaacson, J. C. Newell, S. Simske, and J. Goble. Noser: An algorithm for solving the inverse conductivity problem. *International Journal of Imaging Systems and Technology*, 2(2):66–75, 1990.

[18] M. Cheney, D. Isaacson. Distinguishability in impedance imaging. *Biomedical Engineering, IEEE Transcations on*, 39(8):852–860, 1992.

[19] M. Cheney, D. Isaacson, and E.L. Isaacson. Exact solutions to a linearized inverse boundary value problem. *Inverse Problems*, 6(6):923–934, 1990.

[20] M. Cheney, D. Isaacson. Issues in electrical impedance imaging. *IEEE Computational Science and Engineering*, 2 (1995), pp 85-191.

[21] M. Cheney, D. Isaacson, J. C. Newell. Electrical impedance tomography. *SIAM Review* 41(1):85–101, 1999.

[22] V. Cherepenin, A. Karpov, A. Korjenevsky, V. Kornienko, A. Mazaletskaya, D. Mazourov, D. Meister. A 3D electrical impedance tomography (EIT) system for breast cancer detection. *Physiological measurement* 22(1):9, 2001.

[23] M. H. Choi, T. J. Kao, D. Isaacson, G. Saulnier, J. C. Newell. A reconstruction algorithm for breast cancer imaging with electrical impedance tomography in mammography geometry. *IEEE Transcations on Biomedical Engineering*, 54(4):700-710, 2007.

[24] F. Chollet. Keras. https://keras.io, 2015.

[25] G. Cybenko. Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems*, 2(4) 303-314, 1989.

[26] W. Daily, A. Ramirez. Electrical resistance tomography during in-situ trichloroethylene remediation at the Savannah River Site. *Journal Of Applied Geophysics* 33:239–249, 1995.

[27] W. Daily, A. L. Ramirez. Electrical imaging of engineered hydraulic barriers. *Geophysics*, 65(1):83–94, 2000.

[28] W. Daily, A. Ramirez, R. Johnson. Electrical impedance tomography of a perchloroethylene release. *Journal of Environmental and Engineering Geophysics* 2:189–201, 1998.

[29] W. Daily, A. Ramirez, D. La Brecque, J. Nitao. Electrical resistivity tomography of vadose water movement. *Water Resources Research* 28:1429–1442, 1992.

[30] M. DeAngelo and J.L. Mueller. 2D D-bar reconstructions of human chest and tank data using an improved approximation to the scattering transform. *Physiological measurement*, 31(2):221–232, 2010.

[31] F. J. Dickin, R. A. Williams, M. S. Beck. Determination of composition and motion of multicomponent mixtures in process vessels using electrical impedance tomograhy-i. principles and process engineering applications. *Chemical Engineering Science* 48(10):1883–1897,1993.

[32] J. M. Deibele, H. Luepschen, S. Leonhardt Dynamic separation of pulmonary and cardiac changes in electrical impedance tomography *Physiological Measurement*, 29(6), 2008.

[33] D.C. Dobson. Convergence of a reconstruction method for the inverse conductivity problem. *SIAM Journal on Applied Mathematics*, 52(2):442–458, 1992.

[34] D.C. Dobson and F. Santosa. An image-enhancement technique for electrical impedance tomography. *Inverse Problems*, 10(2):317–334, 1994.

[35] M. Dodd and J.L. Mueller. A real-time D-bar algorithm for 2-D electrical impedance tomography data. *Inverse Probl Imaging (Springfield)*, 8(4): 1013–1031, 2014

[36] O. Dorn, H. Bertete-Aguirre, J.G. Berryman, and G.C. Papanicolaou. A nonlinear inversion method for 3D electromagnetic imaging using adjoint fields. *Inverse Problems*, 15(6):1523–1558, 1999.

[37] M. Eyüboglu, B. Brown, D. Barber. In vivo imaging of cardiac-related impedance changes. *IEEE Eng. Med. Biol. Mag.*, 8 39–45, 1989

[38] L. D. Faddeev. Increasing solutions of the Schrödinger Equation. *Soviet Physics Doklady*, 10:1033-1035, 1966.

[39] Forced Expiration. *http://oac.med.jhmi.edu/res_phys/Encyclopedia/ForcedExpiration/ForcedExpiration.HTML*, Accessed: 2017-11-30

[40] L. M. Fuks, M. Cheney, D. Isaacson, D. G. Gisser, and J. C. Newell. Detection and imaging of electric conductivity and permittivity at low frequency. *Biomedical Engineering, IEEE Transcations on,* 38(11):1106 – 1110, 1991.

[41] P. Gaggero, A. Adler, A. Waldmann, Y. Mamatjan, J. Justiz, V. Koch. Automated robust test framework for electrical impedance tomography. *Physiol. Meas.* 36:1227 -1244.

[42] D. Gilbarg, N. Trudinger Elilptic Partial Differential Equations of Second Order. *Springer-Verlag*. 1983.

[43] A. Greenleaf, Y. Kurylev, M. Lassas, G. Uhlmann. Invisibility and Inverse problems. *Bull. Amer. Math. Soc.* 46:55-97,2009.

[44] M. Hallaji, A. Seppnen, M. Pour-Ghaz. Electrical impedance tomography-based sensing skin for quantitative imaging of damage in concrete. *Smart Materials and Structures*, 23(8):085001, 2014.

[45] R. J. Halter, A. Hartov, K. D. Paulsen. A broadband high-frequency electrical impedance tomography system for breast imaging. *Biomedical Engineering, IEEE Transcations on,* 55(2):650–659, 2008.

[46] P.S. Hamilton, W.J. Tompkins. Adaptive matched filtering for QRS detection. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1988.

[47] S.J. Hamilton, C. N. L. Herrera, J. L. Mueller, and A. Von Herrmann. A direct D-bar reconstruction algorithm for recovering a complex conductivity in 2D. *Inverse Problems*, 28(9):095005, 2012.

[48] S.J. Hamilton, A. Hauptmann. Deep D-Bar: Real time Electrical Impedance Tomography Imaging with Deep Neural Networks. *IEEE Transactions on Medical Imaging*, PP(99), 2017.

[49] S.J. Hamilton, C. N. L. Herrera, J. L. Mueller Direct EIT Reconstructions of Complex Admittivities on a Chest-Shaped Domain in 2-D. *IEEE Transactions on Medical Imaging*, 32(4), 2013

[50] L. M. Heikkinen, J. Kourunen, T. Savolainen, P. J. Vauhkonen, J. P. Kaipio. Real time three-dimensional electrical impedance tomography applied in multiphase flow imaging. *Measurement Science and Technology* 17(8):2083, 2006.

[51] J. Hola, K. Schabowicz. State-of-the-art non-destructive methods for diagnostic testing of building structures anticipated development trends. *Archives of Civil and Mechanical Engineering*, 10(3):5–18, 2010.

[52] T. C. Hou, J. P. Lynch. Electrical impedance tomographic methods for sensing strain fields and crack damage in cementitious structures. *Journal of Intelligent Material Systems and Structures* , 20(11):1363–1379, 2009.

[53] M. Ikehata. Probe method and a Carleman function. *Inverse Problems*, 2007.

[54] M. Ikehata. Reconstruction of the shape of the inclusion by boundary measurements. *Communications in Partial Differential Equations*, 23 1459-74, 1998

[55] M. Ikehata. Reconstruction of the support function for inclusion from boundary measurements. *Journal of Inverse and Ill-posed Problems*, 2000.

[56] M. Ikehata. Recent Development of Probe and Enclosure Methods. *Seminar Notes of Mathematical Sciences*, 10, 2007.

[57] M. Ikehata. Yarmukhamedov's Green function and Mittag-Leffler's function. *Proceedings of the 26th Sapporo Symposium on Partial Differential Equations*, 2001.

[58] D. Isaacson and M. Cheney. Current problems in impedance imaging. *Inverse problems in partial differential equations*, pages 141–149, 1990.

[59] D. Isaacson and M. Cheney. Effects of measurement precision and finite numbers of electrodes on linear impedance imaging algorithms. *SIAM Journal on Applied Mathematics*, 51(6):1705–1731, 1991.

[60] D. Isaacson and E.L. Isaacson. Comment on A.P. Calderóns paper:on an inverse boundary value problem. *mathematics of computation*, 52(186):553–559, 1989.

[61] D. Isaacson, J. L. Mueller, J. C. Newell, S. Siltanen. Imaging cardiac activity by the D-bar method for electrical Impedance tomography. *Physiological Measurement* 27(5):S43–S50, 2006.

[62] D. Isaacson, J.L. Mueller, J.C. Newell, and S. Siltanen. Reconstructions of chest phantoms by the D-bar method for electrical impedance tomography. *Medical Imaging, IEEE Transactions on*, 23(7):821–828, 2004.

[63] H. Jin, S. Yang, M. Wang, R. A. Williams. Measurement of gas holdup profiles in a gas liquid cocurrent bubble column using electrical resistance tomography. *Flow Measurement and Instrumentation*, 18(56):191–196, 2007.

[64] J. Jordana, M. Gasulla, R. Pallás-Areny. Electrical Resistance Tomography to dectect leaks from buried pipes *Measurement Science and Technology* 12:1061–1068, 2001.

[65] J.P. Kaipio, V. Kolehmainen, E. Somersalo, and M. Vauhkonen. Statistical inversion and Monte Carlo sampling methods in electrical impedance tomography. *Inverse Problems*, 16(5):1487–1522, 2000.

[66] A. Kemna, A. Binley, A. Ramirez, W. Daily. Complex resistivity tomography for environmental applications. *Chemical Engineering Journal* 77:11–18, 2000.

[67] A. Kemna, B. Kulessa, H. Vereecken. Imaging and characterization of subsurface solute transport using electrical resistivity tomography (ERT) and equivalent transport models *Journal of Hydrolog*, 267 (2002),pp. 125-146.

[68] N. Kerrouche, C.N. McLeod, and W.R. Lionheart. Time series of eit chest images using singular value decomposition and fourier transfrom. *Phys. Meas.*, 22:147–157, 2001.

[69] S. Kim, A. N. Nkaya T. Dyakowski. Measurement of mixing of two miscible liquids in a stirred vessel with electrical resistance tomography. *International Communications in Heat and Mass Transfer* 33(9):1088–1095, 2006.

[70] K. Knudsen, M. Lassas, J. Mueller, S. Siltanen. D-bar method for electrical impedance tomography with discontinuous conductivities. *SIAM Journal on Applied Mathematics*, 67 (2007),pp. 893-913.

[71] K. Knudsen, M. Lassas, J.L. Mueller, and S. Siltanen. Regularized D-bar method for the inverse conductivity problem. *Inverse Problems and Imaging*, 35(4):599–624, 2009.

[72] R. V. Kohn, M. Vogelius. Determining conductivity by boundary measurements. *Communications on Pure and Applied Mathematics*, 37(3):289–298, 1984.

[73] R. V. Kohn, M. Vogelius. Identification of an unknown conductivity by means of measurements at the boundary. *Laboratory for Numerical Analysis, Technical Note* BN-1005, 1983.

[74] V. Kolehmainen, E. Somersalo, P.J. Vauhkonen, M. Vauhkonen, and J.P. Kaipio. A Bayesian approach and total variation priors in 3D electrical impedance tomography. *In Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE*, volume 2, pages 1028–1031 vol.2, 1998.

[75] G. Kosowski, T. Rymarczyk. Using Neural Networks and Deep Learning Algorithms in Electrical Impedance Tomography. *Informatics Control Measurement in Economy and Environment Protection*, 99-102, 2017.

[76] J. Kourunen, R. Kyhk, J. Matula, J. Kyhk, M. Vauhkonen, L. M. Heikkinen. Imaging of mixing of two miscible liquids using electrical impedance tomography and linear impedance sensor. *Flow Measurement and Instrumentation* 19(6):391–396, 2008.

[77] H. Kwon, J. I. Choi, J. K. Seo. An Electrical impedance monitoring of water-lubricated oil transportation. *Flow Measurement and Instrumentation*, 46, Part B:327–333, 2015

[78] J. Lampinen, A. Vehtari, K. Leinonen. Application of Bayesian Neural Network in Electrical Impedance Tomography. *International Joint Conference on Neural Networks. Proceedings*, 1999.

[79] C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, Y. Bengio Batch Normalized Recurrent Neural Networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*,2016.

[80] A. D. Leathard, B. H. Brown, J. Campbell, F. Zhang, A. H. Morice, D. Tayler. A comparison of ventilatory and cardiac related changes in EIT images of normal human lungs and of lungs with pulmonary emboli. *Physiological Measurement* 15(2A):A137–A.

[81] X. Li, Y. Lu, J. Wang, X. Dang, Q. Wang, X. Duan, Y. Sun. An image reconstruction framework based on deep neural network for electrical impedance tomography. *IEEE International Conference on Image Processing (ICIP)*, 2017.

[82] E. Lin. Radiation Risk From Medical Imaging. *Mayo Clinic Proceedings*. 2010;85(12):1142-1146. doi:10.4065/mcp.2010.0260.

[83] S. Martin, C. Choi. A Post-Processing Method for Three-Dimensional Electrical Impedance Tomography. *Scientific Report*,7(7212),2017.

[84] S. Martin, C. Choi. A novel post-processing scheme for two-dimensional electrical impedance tomography based on artificial neural networks. *PLOS ONE*, 2017.

[85] A. Mary, A. Kumar, A. Chacko. Blind Source Separation Using Wavelets *IEEE International conference on computational Intelligence and Computing Research*, 2010.

[86] MATLAB and Optimization Toolbox Release 2018b. The Mathworks, Inc, Natick, Massachusetts, United States.

[87] M. Mellenthin, J. Mueller, E.D.L. Bueno de Camargo, F. Silva de Moura, T.B.R. Santos, R. Lima ,S. Hamilton, P. Muller, and M. Alsaker. The ACE1 Electrical Impedance Tomography System for Thoracic Imaging. em IEEE Transations on Instrumentation and Measurement, PP(99):1-14, Nov 2018.

[88] M. Michalikova, R. Abed, M. Prauzek, J. Koziorek. Image reconstruction in electrical impedance tomography using neural network. *Cairo International Biomedical Engineering Conference (CIBEC)*, 2014.

[89] J.L. Mueller, P. Muller, M. Mellenthin, R. Murthy, M. Capps, M. Alsaker, R. Deterding, S. Sagel, R. DeBoer. Estimating regions of air trapping from electrical impedance tomography data. *Physiological Measurement*, 2018.

[90] J.L. Mueller, S. Siltanen, and D. Isaacson. A direct reconstruction algorithm for electrical impedance tomography. *IEEE Transactions on Medical Imaging*, 21(6):555–559, 2002.

[91] J.L. Mueller and S. Siltanen. Direct reconstructions of conductivities from boundary measurements. *SIAM Journal on Scientific Computing*, 24(4):1232–1266, 2003.

[92] J. L. Mueller and S. Siltanen. Linear and Nonlinear Inverse Problems with Practical Applications. *SIAM*, 2012.

[93] P. Muller, J. Mueller, M. Mellenthin, R. Murthy, M. Capps, B. Wagner, M. Alsaker, R. Deterding, S. Sagel, J. Hoppe. Evaluation of surrogate measures of pulmonary function derived from electrical impedance tomography data in children with cystic fibrosis. *Physiological Measurement*, 2017.

[94] E. K. Murphy, J. L. Mueller. Effect of domain-shape modeling and measurement errors on the 2-D D-bar method for electrical impedance tomography. *IEEE Trans. Med. Imaging* 28-10: 1567–1584.

[95] A. Nachman, J. Sylvester, and G. Uhlmann. An n-dimensional Borg-Levinson theorem. *Communications in Mathematical Physics*, 115(4):595–605, 1988.

[96] A.I. Nachman. Reconstructions from boundary measurements. *Annals of Mathematics*, 128(3):531–576, 1988.

[97] A.I. Nachman. Global uniqueness for a two-dimensional inverse boundary value problem. em Annals of Mathematics, 143(1):71?96, 1996.

[98] J. C. Newell, P. M. Edic, X. Ren, J. L. Larson-Wiseman, M. D. Danyleiko Assessment of pulmonary edema in dogs by electrical impedance imaging. *IEEE Transactions on Biomedical Engineering*, 43(2):133–138, 1996.

[99] D. T. Nguyen, C. Jin, A. Thiagalingam, A. L. McEwan. A review on electrical impedance tomography for pulmonary perfusion imaging. *Physiological Measurement* 33(5):695, 2012.

[100] A. V. Oppenheim. Digital Signal Processing *Prentice-Hall*, 1975

[101] A. Ramirez, W. Daily, D. La Brecque, E. Owen, D. Chestnut. Monitoring an underground steam injection process using electrical resistance tomography. *Water Resources Research* 29:73–88, 1993.

[102] M. Pal, R. Roy, J. Basu, M. Bepari. Blind source separation: A review and analysis. *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation*, 2013

[103] R. Pikkemaat, S. Leonhardt Separation of ventilation and perfusion related signals within EIT-data streams *J. Phys.: Conf.* Ser. 224 012028, 2010.

[104] T. Rahman, M.M. Hasan, A. Farooq, M.Z. Uddin. Extraction of cardiac and respiration signals in electrical impedance tomography based on independent component analysis. *Journal of Electrical Bioimpedance*, vol 4, 38-44, 2013.

[105] A. Ramirez, W. Daily, D. J. LaBreque, D. Roelant Detection of leaks in underground storage tanks using electrical resistance method. *Journal of Environmental and Engineering Geophysics*, 1:189–203, 1996.

[106] A.L. Ramirez, J.J. Nitao, W.G. Hanley, R. Aines, R.E. Glaser, S.K. Sengupta, K.M. Dyer, T.L. Hickling, and W.D. Daily. Stochastic inversion of electrical resistivity changes using a Markov chain Monte Carlo approach. *Journal of Geophysical Research: Solid Earth*, 110(B2), 2005.

[107] V. Rimpiläinen, L. M. Heikkinen, M. Kuosmanen, A. Lehikoinen, A. Voutilainen, J. Vauhkonen, M. Ketolainen An electrical impedance tomography-based approach to monitor in vitro sodium chloride dissolution from pharmaceutical tablets. *Review of Scientific Instruments* 80(10), 2009.

[108] V. Rimpiläinen, M. Kuosmanen, J. Ketolainen, K. Jrvinen, M. Vauhkonen, L. M. Heikkinen. Electrical impedance tomography for three dimensional drug release monitoring. *European Journal of Pharmaceutical Sciences* 41(2):407–413,2013.

[109] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv:1609.04747v2*, 2017.

[110] T. Rymarczyk, E. Kozłowski, G. Kozłowski. Object Analysis Using Machine Learning to Solve Inverse Problem in Electrical Impedance Tomography. *IEEE International Conference on Imaging Systems and Techniques*, 2018.

[111] A. V. Shahidi, R. Guardo, P. Savard. On the monitoring of pulmonary edema by impedance tomography. In *Engineering in Medicine and Biology Society, 1994. Engineering Advances: New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE* 1994.

[112] S. Siltanen, J. Mueller, and D. Isaacson. An implementation of the reconstruction algorithm of A Nachman for the 2D inverse conductivity problem. *Inverse Problems*, 16(3):681–699, 2000.

[113] E. Somersalo, J.P. Kaipio, M.J. Vauhkonen, D. Baroudi, and S. Jaervenpaeae. *Impedance imaging and Markov chain Monte Carlo methods*, 1997

[114] E. Somersalo, M. Cheney, D. Isaacson, and E. Isaacson. Layer stripping: a direct numerical method for impedance imaging. *Inverse Problems*, 7(6):899–926, 1991.

[115] E. M. Stein, R. Shakarchi. Princeton Lectures in Analysis II: Complex Analysis. em Princeton University Press, 2003.

[116] T. Strauss and T. Khan. Statistical inversion in electrical impedance tomography using mixed total variation and non-convex lp regularization prior. *Journal of Inverse and Ill-posed Problems*, 23(5):529–542, 2015.

[117] X. Sun, O. Jakob, J. Malmivuo, S. Leonhardt. Separation of Cardiac- and Ventilation-related Signals within Electrical Impedance Tomography Data based on Multi-dimensional Ensemble Empirical Mode Decomposition. *IFAC Papers Online*,2017.

[118] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. Software available from https://www.tensorflow.org

[119] A. Vonk. Noordegraaf, T. J. C. Faes, A. Janse, J. T. Marcus, J. G. F. Bronzwaer, P. E. Postmus, P. M. J. M. de. Vries Noninvasive assessment of right ventricular diastolic function by electrical impedance tomography. *CHEST* 111(5):1222–1228, 1997.

[120] R. M. West, D. M. Scott, G. Sunshine, J. Kostuch, L. Heikkinen, M. Vauhkonen, B. S. Hoyle, H. I. Schlaberg, R. Hou, R. A. Williams In situ imaging of paste extrusion using electrical impedance tomography. *Measurement Science and Technology* , 13(12):1890, 2002.

[121] R.M. West, R.G. Aykroyd, S. Meng, and R.A. Williams. Markov chain Monte Carlo techniques and spatialtemporal modelling for medical EIT. *Physiological Measurement*, 25(1):181–194, 2004.

[122] J. Sylvester, G. Uhlmann A uniqueness theorem for an inverse boundary value problem in electrical prospection. *Communications on Pure and Applied Mathematics*, 39(1):91–112, 1986.

[123] J. Sylvester, G. Uhlmann A global uniqueness theorem for an inverse boundary value problem. *Annals of Mathematics*, Second Series, Vol. 125, No. 1 (Jan., 1987), pp. 153-169

[124] J. Sylvester, G. Uhlmann The dirichlet to neumann map and applications. *Proceedings of the Conference Inverse problems in partial differential equations (Arcata, 1989)*, SIAM, Philadelphia, pages 101–139, 1990.

[125] G. Uhlmann Electrical impedance tomography and Calderón's problem. *Inverse Problems*, 25 (2009), 123011

[126] G. Uhlmann Inverse problems: seeing the unseen. *Bulletin of Mathematical Sciences*, 2014, pp 1-71.

[127] A. Vonk-Noordegraaf, P. Kunst, A. Janse, J. Marcus, P. Postmus, T. Faes, P. de Vries. Pulmonary perfusion measured by means of electrical impedance tomography. *Physiol. Meas.*, 19 263–73 , 1998.

[128] H. Wang, B. Raj. On the Origin of Deep Learning. *arXiv*, 2017.

[129] W. Zhang, L. Ge. Application of Adaptive Matched Filter to ECG Signal Detection. *Proceedings of the 7th World Congress on Intelligent Control and Automation*, 2008.