Microwave Imaging for Walk-While-Scan Security Screening

by

K. Parker Trofatter

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

_____
David Smith, Supervisor

_____
Matt Reynolds

_____
Michael Gehm

_____
Jeffrey Krolik

_____
Leslie Collins

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
Duke University

2023

ABSTRACT

Microwave Imaging for Walk-While-Scan Security Screening

by

K. Parker Trofatter

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

_____
David Smith, Supervisor

_____
Matt Reynolds

_____
Michael Gehm

_____
Jeffrey Krolik

_____
Leslie Collins

An abstract of a dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
Duke University

2023

Abstract

Microwave imaging is an important tool for security screening applications. Low-power microwave radiation is used to safely and noninvasively form high resolution images of people to screen for concealed threat objects. This is because microwaves easily pass through clothing and strongly reflect off skin and many materials of interest.

Image resolution improves with large aperture size and bandwidth. Commercial imaging systems realize large apertures two ways; as phased array of antennas, or synthetically by mechanically scanning antennas. In both cases long acquisition times permit only one person to be screened at a time while holding a pose. Security checkpoints employing these systems suffer low screening throughput and present a bottleneck that endangers people. Economically increasing screening throughput requires allowing people to move unimpeded while being imaged.

Computational imaging with a frequency diverse aperture provides a path forward. Frequency diverse apertures are composed of antennas designed to have spatially uncorrelated radiation patterns as a function of frequency. A transceiver drives the antennas with a frequency sweep to rapidly take uncorrelated measurements of a scene. A physical model relating transceiver measurements to scene reflectivity is then numerically solved to form an image. In this way hardware complexity is traded for modeling complexity, leveraging computing technology. The resulting system is inexpensive, modular, flat, and has no moving parts.

An experimental microwave imaging system consisting of a frequency diverse aperture driven by a MIMO transceiver operating from 17.5 GHz to 26.5 GHz is described. The imaging system has 24 Tx antennas, 72 Rx antennas, and samples 100 frequency points giving 172800 possible measurement combinations. The transceiver uses an

orthogonal coding strategy to acquire complete sets of measurements at 7 Hz, enabling a walk-while-scan modality. Depth cameras are integrated to inform image reconstruction and analysis. Several acceleration strategies are pursued to reduce image reconstruction times. A comprehensive simulation platform is used to optimize system configuration. Near real-time imaging of multiple people in motion is demonstrated.

Images of people walking present unique challenges for automated threat detection. A deformable stitching model for combining images is developed, and a framework for applying the stitching model is proposed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

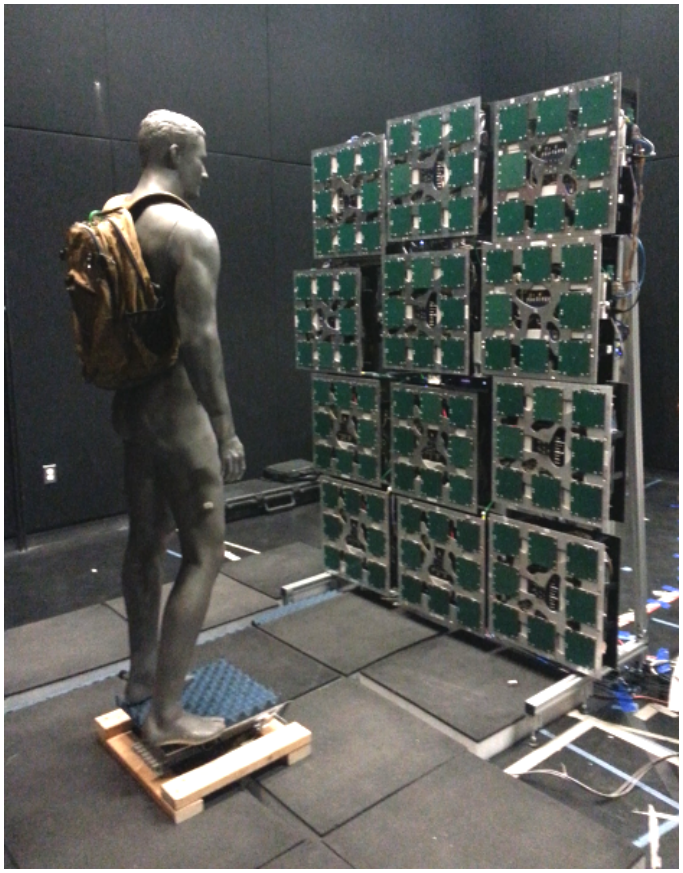

Figure 1.1: Experimental Walk-While-Scan Imaging System — A modular array of stationary antennas images a scene. The antenna radiation patterns have low spatial correlation as a function of frequency. Backscattered transceiver measurements between pairs of antennas encodes scene information related to a physical forward model. Images are reconstructed by inverting the model.

1.1 Background

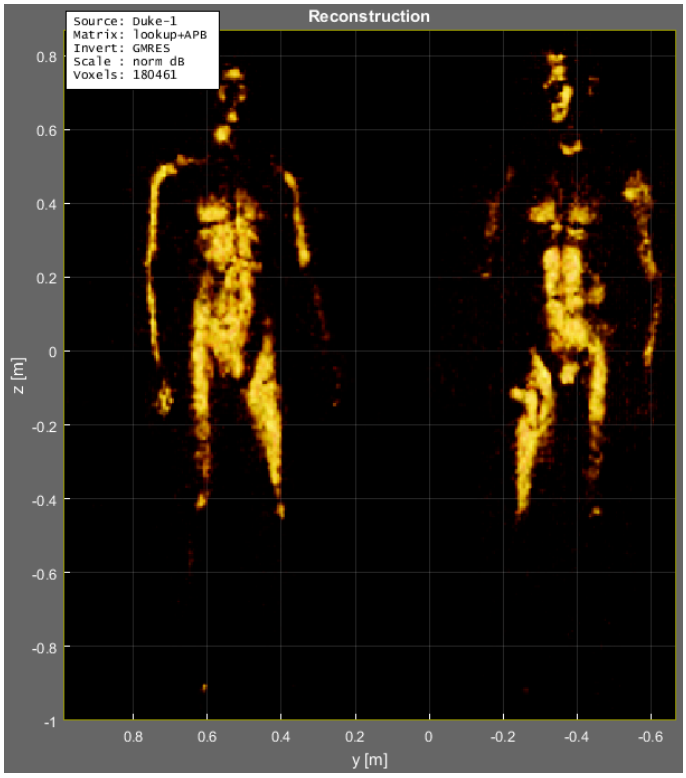

Figure 1.2: Microwave Imaging

An important application at the forefront of 3D imaging and data processing technology is microwave imaging for security screening [SMH01]. Microwaves are a form of non-ionizing electromagnetic waves with a wavelength from about 1mm to 1m. Microwaves around 1cm and less are particularly well suited for security screening people because they safely penetrate clothing, strongly reflect off skin and concealed threat objects, and can form high-resolution images (Fig. 1.2) [SMC+96].

A simplified overview of microwave imaging is presented in Fig. 1.3. A microwave imaging system consists of two core components: a transceiver and antennas. The transceiver (also called a radio and abbreviated xcvr) is an instrument used to take

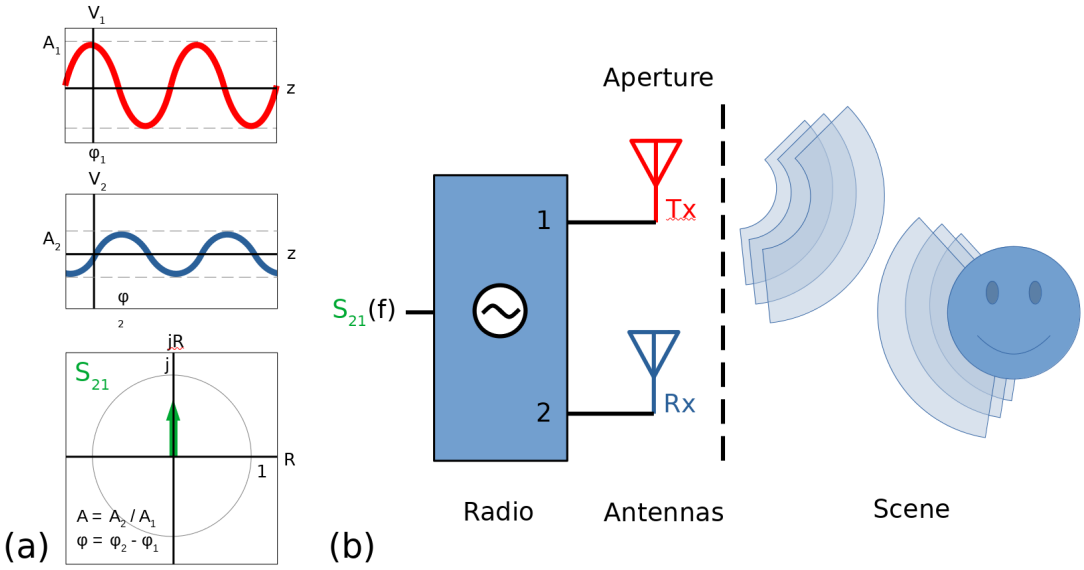

Figure 1.3: Microwave Imaging

radio frequency (RF) measurements (Fig. 1.3.a). Its job is to drive a transmit antenna (tx) with an RF signal at a set frequency, and to compare that signal with the received signal collected by a receive antenna (rx). The received signal depends on the scene in a complex way, but the result is always just a change in amplitude and shift in phase from the drive signal. This is encoded in a single complex number S_{21} whose magnitude is the ratio between the transmitted and received signals, and whose phase is the difference between the signals. The antennas form the aperture. Resolution is closely related to aperture size.

Microwave imagers require large apertures to form images of people with high resolution. Commercially available systems accomplish this one of two ways. One way is with a phased array, a dense array of stationary antennas each with amplitude and phase control (Fig. 1.4.a) [RS16]. Phased arrays set the gold standard for imaging performance. However, phased arrays are expensive because of the large amount of

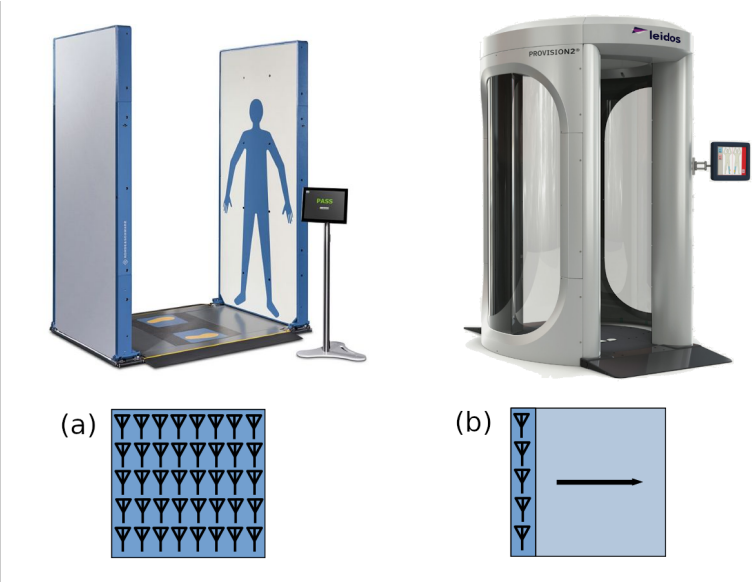

Figure 1.4: QPS and ProVision

RF hardware required to implement them, and0 they are costly to operate because the RF hardware consumes a great deal of power. The second way is to synthesize a large aperture by mechanically scanning a smaller aperture in a technique called synthetic aperture radar (SAR) (Fig. 1.4.b) [SHM+16]. The smaller aperture is more economical, however the mechanical scanning modality forces people to stop and pose while being scanned.

Both types of commercial systems introduce a bottleneck at security checkpoints. Phased array systems cannot be deployed in large numbers due to cost, and SAR systems can only scan one person at a time. In both cases security checkpoints are overwhelmed by long lines outside of a secure area. Such bottlenecks pose security risks to both the people in the lines and the infrastructure where these checkpoints are situated. To alleviate this particular bottleneck, it is necessary to break with existing solutions and aim to implement a walk-while-scan imaging system [GA16].

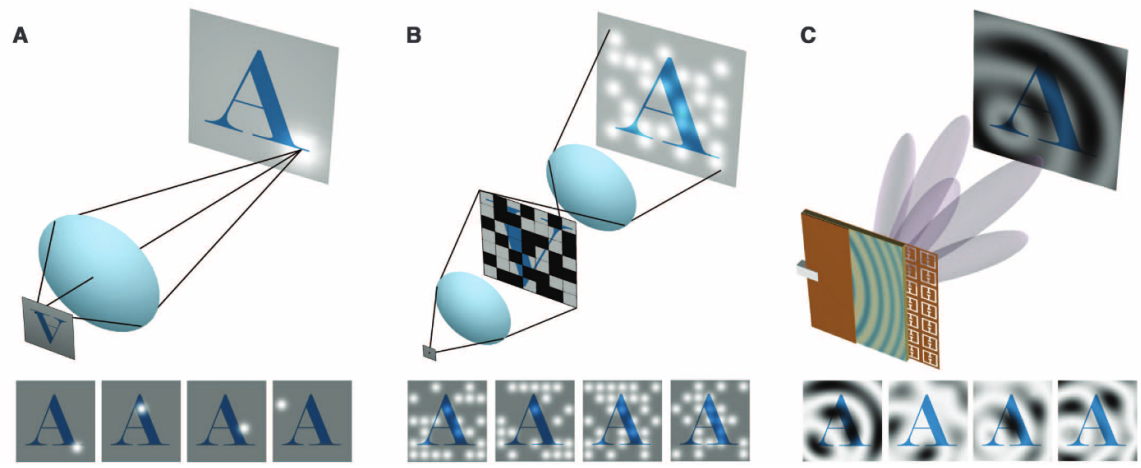

Figure 1.5: Computational Imaging [HDM+13]

Computational imaging provides a path forward. Computational imaging is a generalization of what it means to do imaging. In the past, imaging devices were responsible for both measuring data and forming an image. For example, a camera lens system (1.5.a) focuses a tiny image of a scene onto film with a lens system. The lens system forms the image, while the film records the image. There is no fundamental reason for these tasks to be combined. Indeed, by decoupling measurement from image formation, the design space of imaging sensors is drastically increased, but at the expense of needing a physical model and computer to reconstruct an image from measurement data. An example of this is a coded aperture (Fig. 1.5.b). Coded apertures use a single pixel to measure incoming light that is passed through random masks. If the masks are known and the measurement process can be modeled, then it is possible to reconstruct the same image a normal camera would capture, even though only a single pixel is used. Somehow a small amount of information about the entire scene is encoded in each measurement. A different take on this concept is structured illumination (1.5.c), where instead of passing incoming light through a

mask, a scene is illuminated with light though a mask and the reflection measured. It is important for a computational imaging model to roughly match the number of non-zero image components to the number of measurements taken to reconstruct an image. Compressive sensing is when the number of measurements is less than the number of non-zero image elements, but in this case prior information is required to constrain the image.

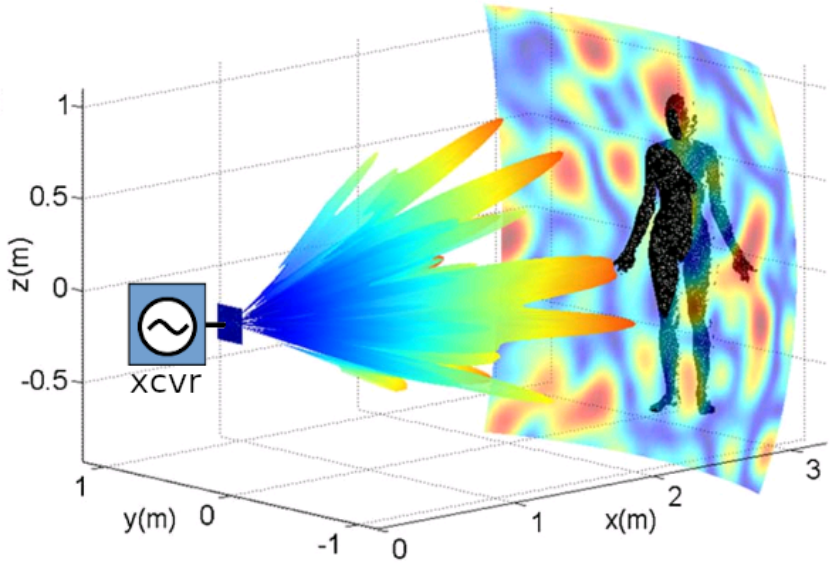

Figure 1.6: Metaimager Concept [LMH+13]

Leaning on computational imaging, a walk-while-scan microwave imaging system based on structured illumination was proposed (Fig. 1.6). The original vision was to use antennas with complicated radiation patterns suitable for structured illumination, design the radiation patterns of these antennas to be sensitive to small changes in frequency, and then to drive such antennas with a rapid frequency sweep. Structured illumination patterns are indexed by frequency, so the antenna doesn't need to be reconfigured. This simple idea was the genesis of the DHS metaimager project.

1.2 Imaging System Overview

An experimental walk-while-scan microwave imaging system was engineered with the goals of being inexpensive, modular, scalable, and flat [PEAG+18] (Fig. 1.1). Broadly speaking, the imager is a structured illumination imaging system operating in K band $[17.5, 26.5]$GHz. The innovative design of the antennas and radio are critical to the operation of the system. The antennas are designed to have spatially uncorrelated radiation patterns w.r.t. frequency to provide structured illumination patterns indexed by frequency. The radio drives a tx antenna at a single frequency to illuminate a scene, and measures the reflected signal with a rx antenna. Frequencies are electronically stepped to rapidly probe a scene with different radiation patterns, and measurements are taken between different tx-rx pairs. The measurement process is physically modeled, and with enough unique measurement information, an image is reconstructed by inverting the measurement model on a computer.

This section highlights key features of the experimental imaging system. The qualities of the antennas, radio, and imaging model are summarized. In-depth information on these topics is found in later chapters or external works.

1.2.1 Frequency Diverse Antennas

Frequency diverse antennas are antennas engineered to have spatially uncorrelated radiation patterns w.r.t. driving frequency f. This type of antenna is suitable for RF structured illumination. The sensitivity of a frequency diverse antenna's radiation pattern to changes in f is of critical design importance to maximize the amount of useful information that can be measured by the antenna. This is because there is only a finite amount of bandwidth B available to sample, and similar radiation

patterns encode redundant information. The sensitivity of an antenna's radiation pattern is related to the antenna's quality factor Q. Q is defined in several related ways, but in general can be thought of characterizing how strongly something resonates. The higher the Q, the more sensitive an antenna's radiation patterns are to changes in f, and the more useful measurements can be packed into the same B. The easiest way to achieve high Q is with a resonant cavity, and the simplest example of a resonant cavity in the RF is a metal box. A resonant cavity can be made into an antenna by poking holes in the cavity's walls that allow radiation to escape.

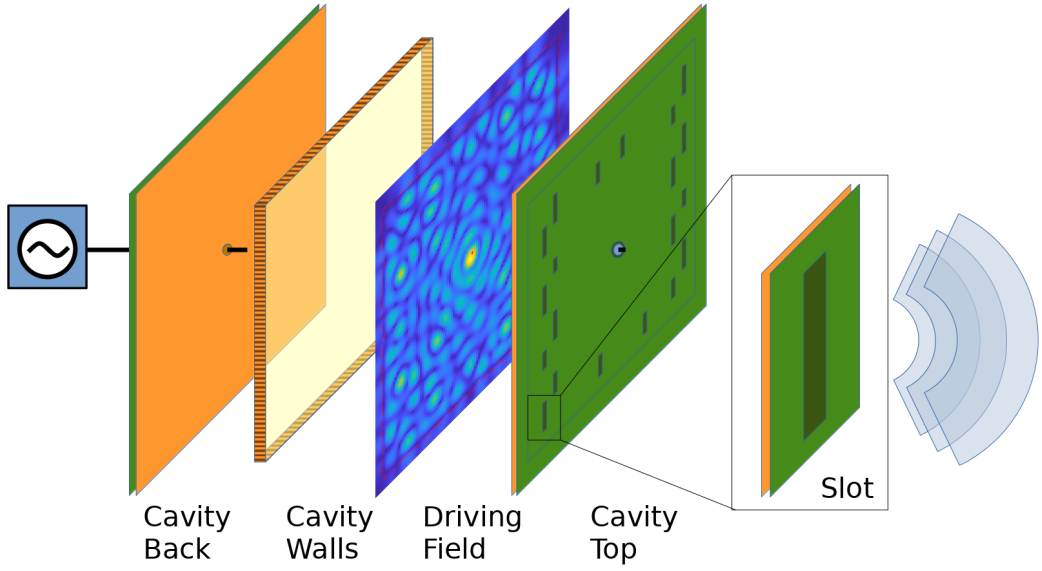

Figure 1.7: Frequency Diverse Antenna Explode

In line with the goals of inexpensive and flat system hardware, cavity-backed metasurface antennas were designed and built using printed circuit board (PCB) technology (Fig. 1.7). PCB is made from a slab of dielectric insulator sandwiched between two copper sheets. The sheets form the top and bottom of the cavity. Vias (holes) are drilled into the PCB and then metalized to electrically connect the top

and bottom of the board. In RF applications, a line of metalized vias at a deeply sub-wavelength pitch act as a metal wall. These via fences are used to realize the walls of the cavity. Weakly resonant slots are cut into the top of the cavity to leak out radiation.

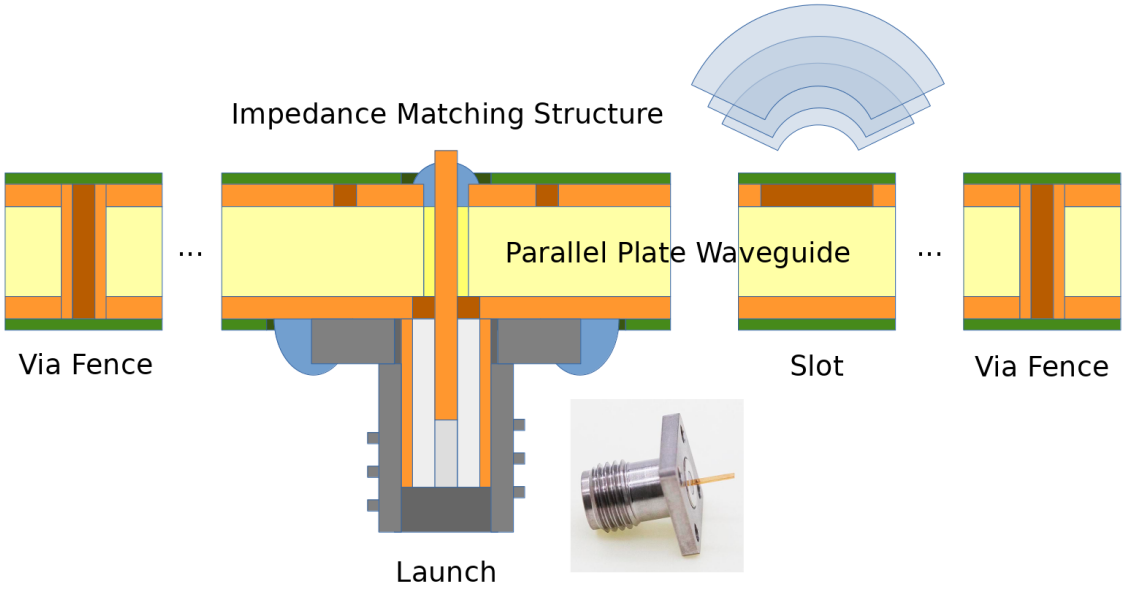

Figure 1.8: Frequency Diverse Antenna Cutaway

The antenna is driven by a coax port installed on the bottom, as seen in Fig. 1.8. The coax pin extends into a via drilled through the PCB cavity, and it is soldered to an impedance matching element on the top of the antenna that is electrically isolated from the surrounding copper. When the pin is driven with an electric current $\mathbf{J}_e$, cylindrical TEM_ρ waves radiate from the pin into the parallel plate waveguide of the PCB (Fig. 1.9.a). The **H** field is in the plane of the PCB, and the **E** field is perpendicular to the plane. The antenna dimensions and driving frequency are such that only the lowest order mode exists. The driving waves propagate away from the coax pin and eventually encounter the via fence. Currents are excited in the

metalized vias, which in turn radiate cylindrical waves in the same fashion as the coax pin. Multiple reflections mix the fields in the cavity in a complicated way that is sensitive to changes in f.

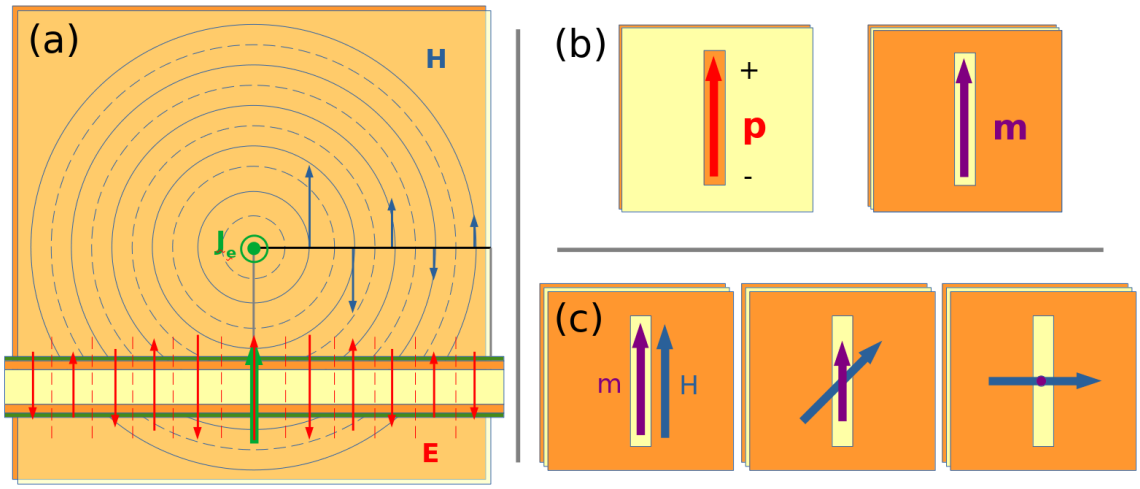

Figure 1.9: Frequency Diverse Antenna Slots

The fields inside the cavity leak out of slots cut into the top of the cavity. A short metal wire will polarize like an electric dipole **p**, so per Babinet's principle a slot will polarize like a magnetic dipole **m** (Fig. 1.9.b). The **H** field in the cavity couples to a slot and induces a magnetic dipole moment that depends on the anisotropic polarizability $\boldsymbol{\alpha}_m$ of the slot (Fig. 1.9.c). A dipole over a metal plane has a simple radiation pattern (Fig. 1.10), so most of the field variation comes from the mode-mixing effect of the cavity randomizing the phase of induced dipoles.

An illustration of an experimental PCB frequency diverse antenna is shown in Fig. 1.11.a. The design of the antenna is the result of design constraints and a long sequence of engineering considerations. Questions concerning antenna size and radiating element shape, polarization, count, and placement had to be answered.

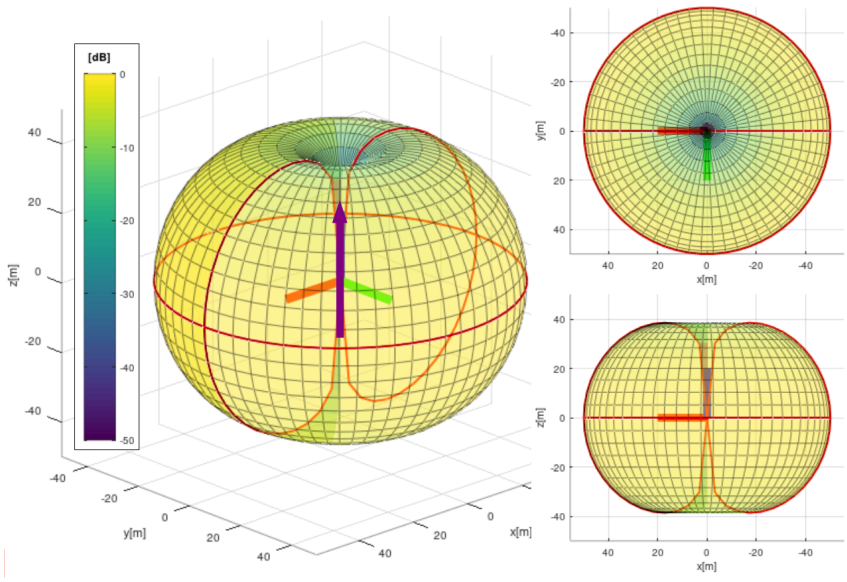

Figure 1.10: Magnetic Dipole

This was accomplished by a combination of qualitatively accurate simulations and rapid prototyping. The design has been well optimized given the design constraints. However given different constraints (i.e. cost being of no concern) higher performance antennas are possible.

Frequency diverse antennas are assembled into an array to form a frequency diverse aperture (Fig. 1.1, Fig. 1.11.b). Like the antenna design, the aperture design is the result of a long sequence of engineering considerations and design constraints. The size of the aperture, the number of tx and rx antennas, and the distribution of antennas all influence imaging performance.

1.2.2 Transceiver

The frequency diverse aperture is driven by a transceiver (radio). The design of the radio is critical in achieving a fast enough a shutter speed with high enough SNR

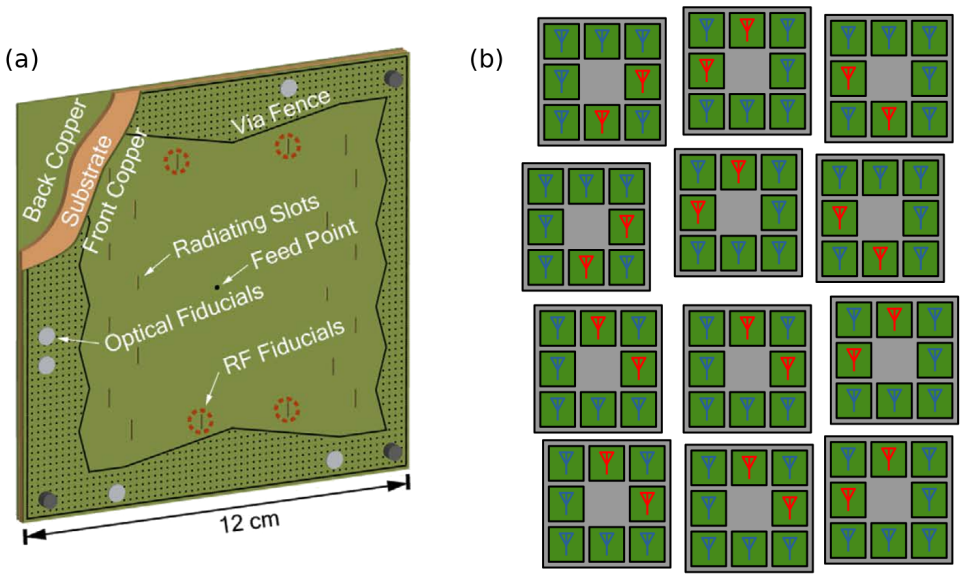

Figure 1.11: Frequency Diverse Aperture

to acquire walk-while-scan data. The demands of the radio are challenging; it must operate over the entirety of K band from $[17.5, 26.5]$GHz, sweep frequency quickly, and be inexpensive. A sequence of radio design generations based on stepped frequency zero IF transceivers satisfied the demands of the experimental imaging system.

An exceptionally simple radio design is a **Zero IF transceiver** (Fig. 1.12). This type of radio has an intermediate frequency at DC, which allows economizing on RF components needed in other RF systems, however the design suffers from self-jamming issues that must be mitigated. At the heart of the zero IF transceiver is a programmable phased locked loop (PLL) that generates a sinusoidal RF signal at a programmed frequency. The PLL is used as the local oscillator (LO) of the radio. The LO is split into a test arm and reference arm. The test arm is amplified and then exits through the tx port, while the reference arm leads to the LO input of a demodulator. The signal from the rx port is amplified and leads into the RF input

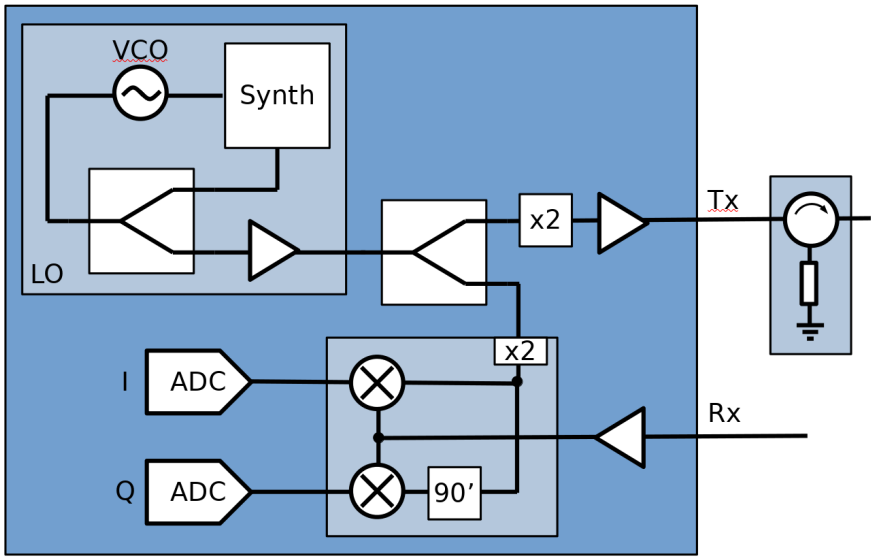

Figure 1.12: Zero IF Transceiver

of the demodulator. Inside the demodulator are a pair of mixers. Mixers function to multiply signals. In this case the LO reference signal and a 90° phase shifted copy are multiplied by the RF signal. In practice the RF signal will be a signal from the rx port excited by the tx port, and therefore will have the same frequency as the LO. The product of two sinusoids at the same frequency results in a sinusoid at twice the frequency with a DC offset. The DC output of the mixers is sampled by a pair of analog to digital converters (ADCs) and labeled I for *in phase* and Q for *quadrature*. The high-frequency term simply averages to zero over the time scale of a single ADC sample conversion. I and Q by themselves don't mean much, but when compared against a calibration measurement, encode the magnitude ratio and phase difference between the transmitted tx port signal and the received rx port signal, the S_{21} parameter!

A two port zero IF transceiver can drive an array if there is a RF switching network

connecting the radio to the aperture. However this arrangement is only suitable for static scenes because of the long acquisition times involved with switching through all tx-rx combinations in a large array. A parallel system is needed for real-time data acquisition.

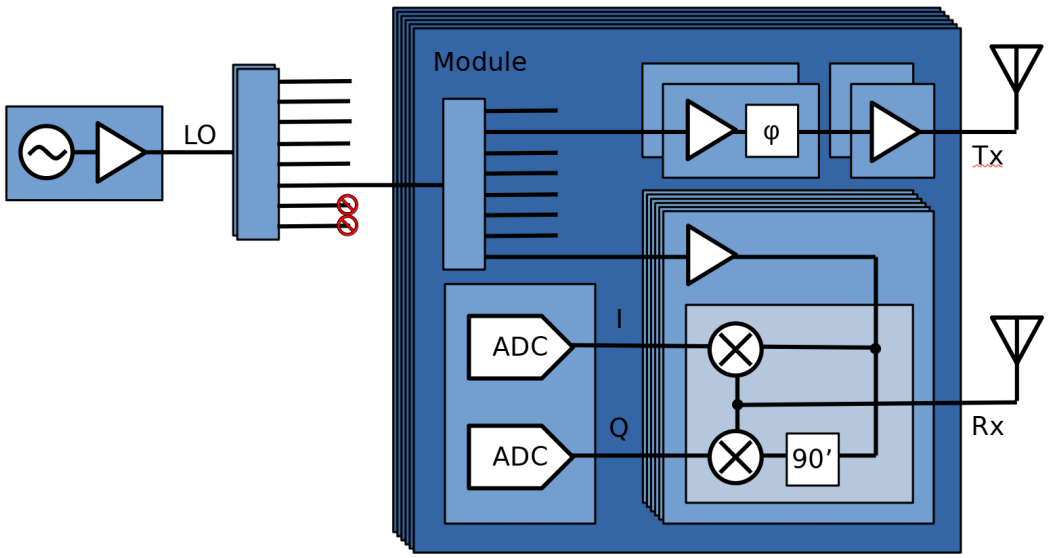

Figure 1.13: OCAI

Orthogonal Coded Active Illumination (OCAI) is a stepped frequency transceiver design capable of massive parallel data acquisition by using a multiplexing strategy (Fig. 1.13). The OCAI radio is a modular multiple-in multiple-out (MIMO) device that simultaneously transmits on 24 tx ports and receives on 72 rx ports when operating. The basic building blocks of the OCAI radio are zero IF radios that are distributed across several RF modules. The key addition to the OCAI radio are phase shifters before tx ports. These phase shifters enable phase shift keying strategies. OCAI adopts a binary phase shift keying (BPSK) strategy. Every tx is assigned a unique 32-bit orthogonal balanced binary code. A measurement at a

single frequency for the entire array takes 32 parallel submeasurements. For the i-th submeasurement, each tx phase shifter is set to the 0° or 180° depending on the i-th bit in its code. Orthogonal codes enable the measurements to be multiplexed, while balanced codes have the special property of mitigating constant self-jamming effects. The submeasurements are demultiplexed using the tx phase codes to recover the individual tx-rx pair measurements. Furthermore, each measurement is effectively averaged 32 times, providing outstanding SNR. In fact, this design does not need an amplifier on the rx ports.

The experimental OCAI system achieved a data acquisition rate of 7Hz in the laboratory with 24 tx and 72 rx antennas sampled at 100 frequencies for 172800 measurements. This was fast enough to image people walking through the system. However if a person moves too fast, the scene will evolve an appreciable amount over the finite acquisition time and break the image. A path to 100Hz possible.

1.2.3 Imaging Model

Computational imaging requires a measurement model that connects the physical property being imaged to the sensor measurement. Given a measurement model and a measurement, the model is inverted on a computer to reconstruct an image of the property. Together the measurement model and inversion algorithm constitute the imaging model.

The relevant elements of the measurement model are illustrated in Fig. 1.14. Let the i-th measurement identify a tx, rx, and frequency combination. The physical measurement process is as follows. An electric current at frequency f_i drives the i-th tx antenna, exciting currents on the surface of the antenna which radiate electromag-

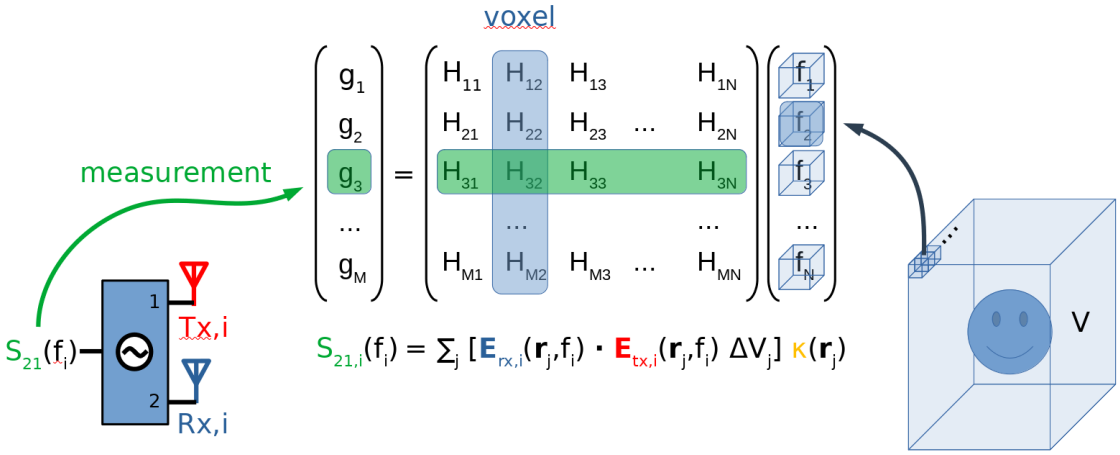

Figure 1.14: g = Hf

netic waves that illuminate a scene. The incident electric field $\mathbf{E}_{tx,i}$ excites currents in the scene per Ohm's law $\mathbf{J}_e = \kappa_e \mathbf{E}$ for electric conductivity distribution κ_e. The scene currents in turn radiate their own waves. Waves from the scene excite currents on the i-th rx antenna. Finally, the radio compares the amplitude and phase of the excitation signal and received signal. Advanced electromagnetic theorems reveal that the coupling of a point in space to any given measurement is proportional to the dot product of the tx and rx electric fields multiplied by κ_e (the rx field $\mathbf{E}_{rx,i}$ is the same field as if the rx were a tx). To make the model linear, the Born approximation is invoked to ignore reflections when considering Ohm's law for the scene such that $\mathbf{J}_e \approx \kappa_e \mathbf{E}_{tx,i}$. The problem is discretized into a linear system by subdividing the imaging volume into voxels, resulting in

$$\mathbf{g} = \mathbf{H}\mathbf{f} \tag{1.1}$$

where $\mathbf{f}$ are voxel conductivities organized into a vector, $\mathbf{g}$ are the radio measurements organized into a vector, and $\mathbf{H}$ is the **measurement matrix** that encodes the

physical measurement model connecting $\mathbf{f}$ to $\mathbf{g}$.

The linear measurement model is used to reconstruct images. However the measurement model is the wrong way around; it maps a scene vector to a measurement vector. To solve for a scene vector given a measurement vector, the model must be inverted. $\mathbf{H}$ can be very large, on the order of 1×10^5 elements on a side, so the only real options for real-time reconstruction are applying a matched filter or iterative linear solver. The simplest is matched filter, which only requires a single application of the conjugate transpose of the forward model. This works because it is essentially correlating the fields with the measurements. Matched filter is effectively the first step in more sophisticated iterative solvers. Two solvers, conjugate gradient and generalized minimum residual (GMRES), produce the highest quality images with a relatively small number of iterations.

Having the imaging model in hand is still not enough to reconstruct experimental images. Two additional tasks must first be complete prior to imaging. The first tasks is antenna registration, which is the process of estimating the spatial relationship between all of the antennas in an imaging system. This process is crucial to correctly model the fields inside an imaging volume. Registration is actually a deeper task than just registering the antennas. Frequency diverse antennas are experimentally measured by near field scan (NFS) to model their fields. It is necessary to also register the NFS with the antenna.

The second task is calibration. Calibration aims to measure and remove the internal response of the radio hardware itself from measurements. This includes the effects of any hardware not already considered in the model (i.e. cables). A variety of calibration measurement techniques and algorithms were researched to address this

issue. The final experimental system used a calibration object, where a calibration object was simulated and experimentally measured, and the discrepancies between simulation and measurement are attributed to the sought for calibration. Algorithms to clean up the measured calibration worked to varying effect.

As mentioned, the measurement matrix $\mathbf{H}$ can be very large. This is partially out of necessity because it is not feasible to compute $\mathbf{H}$ in real-time. However this also makes real-time image reconstruction a daunting task. The measurement model as formulated can be reduced in size if voxels are known to be empty *a priori*. A clever and economical idea is to incorporate depth cameras into the RF imaging system to identify a region of interest (ROI) that includes the location of foreground voxels. This idea requires the depth cameras be registered with the imaging volume, a method to generate a ROI, and a way to apply the ROI during image reconstruction. Using a ROI, along with several other acceleration strategies, made it possible to reconstruct images almost in real-time.

This high level components of the experimental walk-while-scan imaging system are summarized in Fig. 1.15. A frequency diverse aperture is driven by the OCAI radio in a massively parallel mode of operation way. The aperture probes an imaging volume, over which a measurement model is built. Depth cameras identify the location or foreground objects and supplies a ROI to the reconstruction algorithms. Reconstruction uses the depth prior to prune the measurement matrix to a manageable size, and then numerically solves the reduced system on a workstation to arrive at an estimate for the scene vector $\mathbf{f}$.

An experimental image produced by the imaging system is illustrated in Fig. 1.2. This image demonstrates the unique capability of imaging two people at once. This

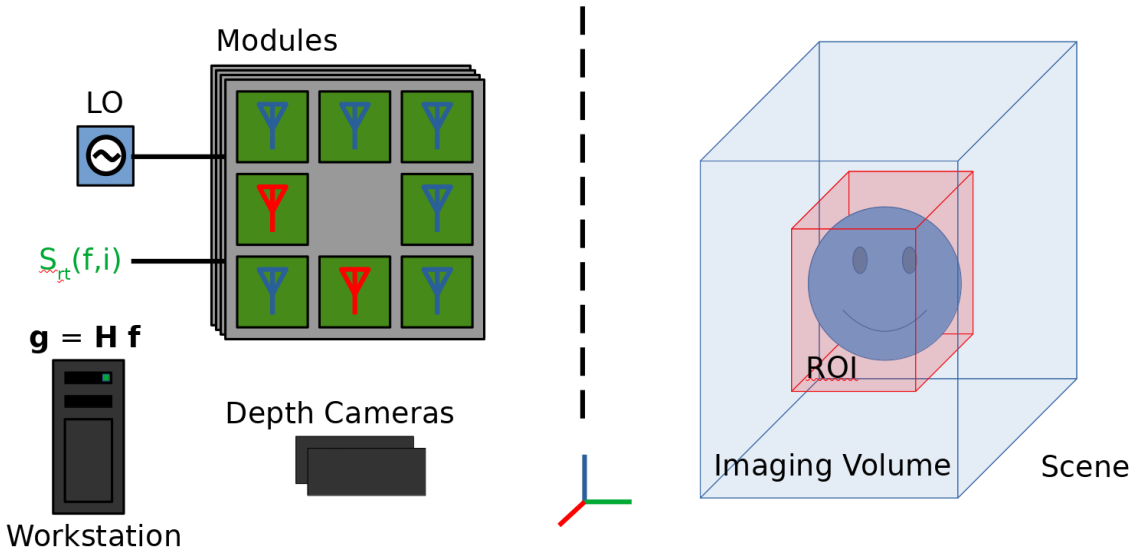

Figure 1.15: Metaimager System

is not possible with commercial imaging systems. This image also illustrates an issue common to all microwave imaging systems; specularity. Many objects of interest are conductive and smooth on the scale of microwaves, and therefore reflect like a mirror. Effectively images are limited to the specular highlights of surfaces that reflect waves directly back at the aperture. Commercial systems mitigate this issue with a favorable imaging environment or by taking measurement data from all directions. However a walk-while-scan system doesn't have this luxury. Specularity and deformation of the scene complicates image analysis and automated threat detection (ATD). Post processing of image data is likely necessary before ATD can be applied.

The success of the imaging system design critically depended on a comprehensive simulation platform. The ability to create and evaluate virtual imaging systems was an invaluable tool for rapid and cost-effective research. This extends to the future image analysis endeavors. The prospect of generating synthetic training data for

machine learning enabled tasks like ATD is an exciting avenue of future research.

1.2.4 Specifications

The development of the imaging system was a precision engineering effort informed by careful investigation of quantitative metrics. For instance, image resolution is of central concern for any imaging application, and it was important that theoretical understanding matched reality. Resolution targets were useful in this respect (Fig. 1.16). A short list of other important system metrics is collected in Tab. 1.1.

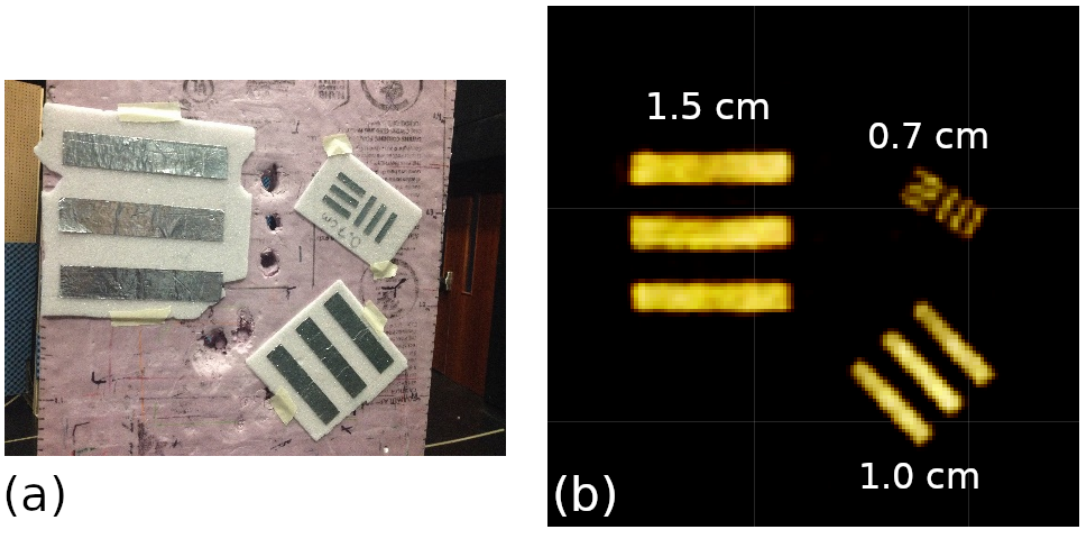

Figure 1.16: Resolution Target

1.3 Contributions

The experimental imaging system was the product of a massive group research effort. At the peak of the project about 50 people were laboring to make the imaging system a

Table 1.1: Imager Specifications

Parameter	Demo B
Band	$[17.5, 26.5]$GHz
Radio	OCAI
Number of Tx	24
Number of Rx	72
Number of Frequencies	100
Aperture Size	$2.1 \times 2.0\text{m}^2$
Field of View	$\pm 60°, \pm 60°$
Resolution FWHM	$< 0.5°$
Resolution @ 1m	0.7cm
Range	$0 - 1.6$m static
Range Resolution	1.6cm
Frame Rate	7Hz
Image Acquisition Time	0.14s
Image Reconstruction Time	0.24s (FAMI)
Total Modes	172800
Antenna Q	330
Radio Carrier Rejection	40dB
Radio Output Power	13dBm

reality. I had the privilege of being the experimental system integrator, and therefore was at the intersection of all the work being done on the project. In this role I was both directly and indirectly involved with many aspects of the project. My major contributions and collaborations that were critical to the success of the project are as follows:

- Integrated experimental imaging system
- Developed constellation registration (Ch. 4)
- Collaborated on NFS registration (Ch. 4)
- Developed array registration (Ch. 4)
- Collaborated on calibration
- Developed depth camera registration (Ch. 5)
- Developed ROI (Ch. 5)
- Developed rigid linear stitching (Ch. 6)
- Developed rigid rotation stitching (Ch. 6)
- Developed rigid skeleton stitching (Ch. 6)
- Developed skeleton stitching model (Ch. 6)

Although not intentional, registration was a common theme in my work.

Chapter 2

Forward Model

A sensor **forward model** is a physical model of a sensor's measurement process. Given an input scene configuration, a forward model predicts the output of a sensor. Possessing a forward model is key for posing and solving the image reconstruction problem (chapter 3), network calibration , and simulation .

Conceptually, the basic metaimager measurement illuminates a scene with a transmit antenna and measures the reflection with a receive antenna (figure 2.1). A transceiver generates a signal at frequency ω and compares the the magnitude and phase of the transmitted and received signals, represented by S-parameter S_{21}. This process is rapidly repeated with spatially uncorrelated radiation patterns to encode scene information. With enough measurements, a computer can form an image.

Several electromagnetic theorems and approximations are used in formulating a tractable forward model for the metaimager. This chapter assembles a forward model from first principles, starting with Maxwell's equations. Electromagnetic reciprocity is then derived to model the coupling between antennas and a scene in terms of sources and fields. A scene model and interpretation is given. An antenna model and associated radiation model are developed. Finally the above concepts are combined to derive a linear forward model.

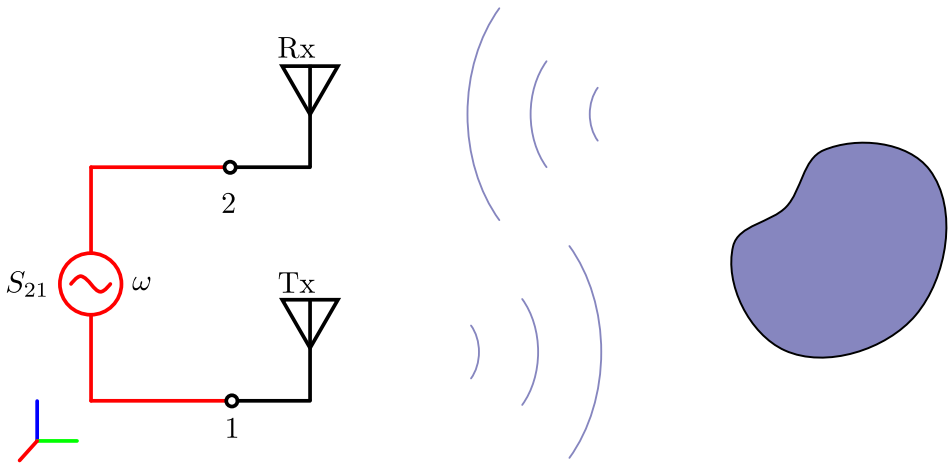

Figure 2.1: Metaimager - General idea

2.1 Maxwell's Equations

Consider Maxwell's equations in matter in differential form (appendix A)

$$\begin{cases} \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \times \mathbf{H} = \mathbf{J}_f + \frac{\partial \mathbf{D}}{\partial t} \\ \nabla \cdot \mathbf{D} = \rho_f \\ \nabla \cdot \mathbf{B} = 0 \end{cases} \qquad \text{BCs:} \begin{cases} \hat{\mathbf{n}} \times \Delta \mathbf{E} = \mathbf{0} \\ \hat{\mathbf{n}} \times \Delta \mathbf{H} = \mathbf{K}_f \\ \hat{\mathbf{n}} \cdot \Delta \mathbf{D} = \sigma_f \\ \hat{\mathbf{n}} \cdot \Delta \mathbf{B} = 0 \end{cases} \tag{2.1}$$

with field definitions

$$\mathbf{D} \equiv \epsilon_0 \mathbf{E} + \mathbf{P} \tag{2.2}$$

$$\mathbf{H} \equiv \frac{1}{\mu_0} \mathbf{B} - \mathbf{M} \tag{2.3}$$

and the electromagnetic (Lorentz) force law

$$\mathbf{f}_e = \mathbf{E} + \mathbf{v} \times \mathbf{B}. \tag{2.4}$$

This formulation of Maxwell's equations incorporates dipolar material effects at the expense of introducing definitions 2.2 and 2.3, which must be supplemented by constitutive equations for $\mathbf{P}$ and $\mathbf{M}$. Consequently electric sources are classified as either free or bound

$$\rho = \rho_f + \rho_b = \rho_f - \nabla \cdot \mathbf{P} \tag{2.5}$$

$$\mathbf{J} = \mathbf{J}_f + \mathbf{J}_b + \mathbf{J}_p = \mathbf{J}_f + \nabla \times \mathbf{M} + \frac{\partial \mathbf{P}}{\partial t}. \tag{2.6}$$

Bound sources arise in matter from constituent electric charges responding to forces and establishing electric and magnetic dipole moment densities $\mathbf{P}$ and $\mathbf{M}$. Free sources simply are not bound sources. Higher-order multipole effects may be isolated, but dipole effects dominate in most natural materials.

Maxwell's equations can be made symmetric in electric and magnetic quantities by introducing magnetic source charges and currents. There is no observational evidence for magnetic sources, but the concept is theoretically valuable. Based on dimensional analysis a consistent way to introduce magnetic charge is in terms of the SI derived unit of magnetic flux, the Weber, defined as $\mathrm{Wb} \equiv \mathrm{V} \cdot \mathrm{s}$. Equations 2.1 then become

$$\begin{cases} \nabla \times \mathbf{E} = -\mathbf{J}_{mf} - \frac{\partial \mathbf{B}}{\partial t} \\ \nabla \times \mathbf{H} = \mathbf{J}_{ef} + \frac{\partial \mathbf{D}}{\partial t} \\ \nabla \cdot \mathbf{D} = \rho_{ef} \\ \nabla \cdot \mathbf{B} = \rho_{mf} \end{cases} \qquad \text{BCs:} \begin{cases} \hat{\mathbf{n}} \times \Delta \mathbf{E} = -\mathbf{K}_{mf} \\ \hat{\mathbf{n}} \times \Delta \mathbf{H} = \mathbf{K}_{ef} \\ \hat{\mathbf{n}} \cdot \Delta \mathbf{D} = \sigma_{ef} \\ \hat{\mathbf{n}} \cdot \Delta \mathbf{B} = \sigma_{mf} \end{cases} \tag{2.7}$$

where subscripts e and m differentiate electric and magnetic sources respectively. The symmetry of these equations gives rise to the principle of duality (appendix A), where the electric and magnetic quantities of any derived equation can be exchanged in a prescribed way to yield a valid dual equation.

For linear media, $\mathbf{P}$ and $\mathbf{M}$ are in general related to $\mathbf{E}$ and $\mathbf{H}$ by electric and magnetic susceptibility tensors $\boldsymbol{\chi}_e$ and $\boldsymbol{\chi}_m$ and magnetoelectric tensors $\boldsymbol{\chi}_{em}$ and $\boldsymbol{\chi}_{me}$

$$\mathbf{P} = \epsilon_0 \boldsymbol{\chi}_e \mathbf{E} + \boldsymbol{\chi}_{em} \mathbf{H} \tag{2.8}$$

$$\mu_0 \mathbf{M} = \mu_0 \boldsymbol{\chi}_m \mathbf{H} + \boldsymbol{\chi}_{me} \mathbf{E}. \tag{2.9}$$

Most natural materials are adequately described as non-magnetoelectric where $\boldsymbol{\chi}_{em} = \mathbf{0}$ and $\boldsymbol{\chi}_{me} = \mathbf{0}$, and equations 2.8 and 2.9 reduce to

$$\mathbf{P} = \epsilon_0 \boldsymbol{\chi}_e \mathbf{E} \tag{2.10}$$

$$\mathbf{M} = \boldsymbol{\chi}_m \mathbf{H}. \tag{2.11}$$

Furthermore at millimeter wave frequencies human skin and many other materials of interest for security screening are isotropic, so the tensors can be written as scalars

$$\mathbf{P} = \epsilon_0 \chi_e \mathbf{E} \tag{2.12}$$

$$\mathbf{M} = \chi_m \mathbf{H}. \tag{2.13}$$

Under these assumptions, definitions 2.2 and 2.3 for $\mathbf{D}$ and $\mathbf{H}$ simplify to

$$\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P} = \epsilon_0 \mathbf{E} + [\epsilon_0 \chi_e \mathbf{E}] = \epsilon_0 [1 + \chi_e] \mathbf{E} = \epsilon_0 \epsilon_r \mathbf{E} = \epsilon \mathbf{E} \tag{2.14}$$

$$\mathbf{B} = \mu_0 [\mathbf{H} + \mathbf{M}] = \mu_0 [\mathbf{H} + [\chi_m \mathbf{H}]] = \mu_0 [1 + \chi_m] \mathbf{H} = \mu_0 \mu_r \mathbf{H} = \mu \mathbf{H} \tag{2.15}$$

where the permittivity ϵ and permeability μ are

$$\epsilon = \epsilon_0 \epsilon_r = \epsilon_0 [1 + \chi_e] \tag{2.16}$$

$$\mu = \mu_0 \mu_r = \mu_0 [1 + \chi_m]. \tag{2.17}$$

Eliminate $\mathbf{D}$ and $\mathbf{B}$ from the symmetric Maxwell's equations 2.7 by inserting constitutive equations 2.14 and 2.15 to obtain

$$\begin{cases} \nabla \times \mathbf{E} = -\mathbf{J}_{mf} - \frac{\partial}{\partial t}[\mu \mathbf{H}] \\ \nabla \times \mathbf{H} = \mathbf{J}_{ef} + \frac{\partial}{\partial t}[\epsilon \mathbf{E}] \\ \nabla \cdot [\epsilon \mathbf{E}] = \rho_{ef} \\ \nabla \cdot [\mu \mathbf{H}] = \rho_{mf} \end{cases} \quad \text{BCs:} \begin{cases} \hat{\mathbf{n}} \times \Delta \mathbf{E} = -\mathbf{K}_{mf} \\ \hat{\mathbf{n}} \times \Delta \mathbf{H} = \mathbf{K}_{ef} \\ \hat{\mathbf{n}} \cdot \Delta[\epsilon \mathbf{E}] = \sigma_{ef} \\ \hat{\mathbf{n}} \cdot \Delta[\mu \mathbf{H}] = \sigma_{mf} \end{cases} . \tag{2.18}$$

Substantial analytic simplification can be made for linear time-invariant (LTI) systems where material properties are both linear and constant in time

$$\frac{\partial \epsilon}{\partial t} = 0 \tag{2.19}$$

$$\frac{\partial \mu}{\partial t} = 0. \tag{2.20}$$

While a primary goal of the metaimager is the ability to image moving scenes, scenes of walking people evolve on time scales much longer than the propagation of light across the scene and measurement process, and can be modeled as time-invariant. LTI system eigenfunctions are complex exponentials with angular frequency $\omega = 2\pi f$. Field quantities of these solutions vary sinusoidally at the given frequency, and are described as time-harmonic. For example, a time-harmonic electric field has the form

$$\mathbf{E}(\mathbf{r}, t) = \hat{\mathbf{x}} E_x(\mathbf{r}) \cos(\phi_x(\mathbf{r}) + \omega t) + \hat{\mathbf{y}} E_y(\mathbf{r}) \cos(\phi_y(\mathbf{r}) + \omega t) + \hat{\mathbf{z}} E_z(\mathbf{r}) \cos(\phi_z(\mathbf{r}) + \omega t) \tag{2.21}$$

where $E_i(\mathbf{r})$ and $\phi_i(\mathbf{r})$ are the amplitude and phase functions of the i-th electric field component. Using Euler's identity

$$e^{j\phi} = \cos\phi + j \sin\phi \tag{2.22}$$

and the definition for the real part of a complex number

$$\mathfrak{Re}(\tilde{z}) \equiv \frac{1}{2}[\tilde{z} + \tilde{z}^*] \tag{2.23}$$

the time-harmonic electric field can be written in terms of complex-valued phasors

$$\mathbf{E}(\mathbf{r}, t) = \mathfrak{Re}(\tilde{\mathbf{E}}(\mathbf{r})\tilde{T}(t)) \tag{2.24}$$

where the spatial part is defined as

$$\tilde{\mathbf{E}}(\mathbf{r}) \equiv \hat{\mathbf{x}}E_x(\mathbf{r})e^{j\phi_x(\mathbf{r})} + \hat{\mathbf{y}}E_y(\mathbf{r})e^{j\phi_y(\mathbf{r})} + \hat{\mathbf{z}}E_z(\mathbf{r})e^{j\phi_z(\mathbf{r})} \tag{2.25}$$

and the temporal part is defined as

$$\tilde{T}(t) \equiv e^{j\omega t}. \tag{2.26}$$

It is worth noting the sign of the temporal exponent is a convention, and this work asserts the positive temporal phase convention of electrical engineering. Maxwell's equations for LTI systems are then cast in terms of phasors. Consider Faraday's Law

$$\nabla \times \mathbf{E} = -\mathbf{J}_{mf} - \frac{\partial}{\partial t}[\mu \mathbf{H}] \tag{2.27}$$

written in terms of phasors

$$\nabla \times \mathfrak{Re}(\tilde{\mathbf{E}}\tilde{T}) = -\mathfrak{Re}(\tilde{\mathbf{J}}_{mf}\tilde{T}) - \frac{\partial}{\partial t}[\mu \mathfrak{Re}(\tilde{\mathbf{H}}\tilde{T})]. \tag{2.28}$$

Derivatives with respect to real variables commute with complex conjugation

$$\frac{d}{dx}[\tilde{f}(x)^*] = \lim_{\Delta x \to 0} \frac{\tilde{f}(x + \Delta x)^* - \tilde{f}(x)^*}{\Delta x} = \left[\lim_{\Delta x \to 0} \frac{\tilde{f}(x + \Delta x) - \tilde{f}(x)}{\Delta x}\right]^* = \left[\frac{d\tilde{f}}{dx}\right]^*. \tag{2.29}$$

This implies $\mathfrak{Re}$ commutes with real derivatives, and by extension linear combinations of real differential operators, including vector differential operators. Therefore

$$\mathfrak{Re}([\nabla \times \tilde{\mathbf{E}}]\tilde{T}) = -\mathfrak{Re}(\tilde{\mathbf{J}}_{mf}\tilde{T}) - \mu \mathfrak{Re}\left(\tilde{\mathbf{H}}\frac{d\tilde{T}}{dt}\right) \tag{2.30}$$

where the time independence of the materials has been used and derivatives applied. The special form of time-harmonic fields allows the time derivative to be evaluated

$$\frac{d\tilde{T}}{dt} = \frac{d}{dt}e^{j\omega t} = j\omega e^{j\omega t} = j\omega\tilde{T}. \tag{2.31}$$

Using this, and $\mathfrak{Re}(\alpha\tilde{z} + \beta\tilde{z}') = \alpha\mathfrak{Re}(\tilde{z}) + \beta\mathfrak{Re}(\tilde{z}')$ for $\alpha, \beta \in \mathbb{R}$ and $\tilde{z}, \tilde{z}' \in \mathbb{C}$, write

$$\mathfrak{Re}([\nabla \times \tilde{\mathbf{E}}]\tilde{T}) = \mathfrak{Re}(-\tilde{\mathbf{J}}_{mf}\tilde{T} - j\omega\mu\tilde{\mathbf{H}}\tilde{T}). \tag{2.32}$$

Relax this equation to include the imaginary part and divide out the temporal part

$$\nabla \times \tilde{\mathbf{E}} = -\tilde{\mathbf{J}}_{mf} - j\omega\mu\tilde{\mathbf{H}}. \tag{2.33}$$

Comparing equations 2.33 and 2.27, the time-dependence has been eliminated in exchange for complex-valued fields, and the time derivative is effectively replaced by a scalar multiplication

$$\frac{\partial}{\partial t} \to j\omega \tag{2.34}$$

which is a general result. The same logic can be applied to all the equations of 2.18. The time-harmonic Maxwell's equations in isotropic matter are then

$$\begin{cases} \nabla \times \tilde{\mathbf{E}} = -\tilde{\mathbf{J}}_{mf} - j\omega\mu\tilde{\mathbf{H}} \\ \nabla \times \tilde{\mathbf{H}} = \tilde{\mathbf{J}}_{ef} + j\omega\epsilon\tilde{\mathbf{E}} \\ \nabla \cdot [\epsilon\tilde{\mathbf{E}}] = \tilde{\rho}_{ef} \\ \nabla \cdot [\mu\tilde{\mathbf{H}}] = \tilde{\rho}_{mf} \end{cases} \quad \text{BCs:} \begin{cases} \hat{\mathbf{n}} \times \Delta\tilde{\mathbf{E}} = -\tilde{\mathbf{K}}_{mf} \\ \hat{\mathbf{n}} \times \Delta\tilde{\mathbf{H}} = \tilde{\mathbf{K}}_{ef} \\ \hat{\mathbf{n}} \cdot \Delta[\epsilon\tilde{\mathbf{E}}] = \tilde{\sigma}_{ef} \\ \hat{\mathbf{n}} \cdot \Delta[\mu\tilde{\mathbf{H}}] = \tilde{\sigma}_{mf} \end{cases} . \tag{2.35}$$

Field solutions can be converted back to the time domain by equations like 2.24. Note, the dot product is no longer an inner product because it is not conjugate symmetric, but retains its significance as an instruction on how to combine quantities.

The time-harmonic formalism naturally accommodates conduction currents with only slight modification. Ohm's law is the statement that electric current is excited by a force per electric charge

$$\mathbf{J}_{ec} = \kappa_e \mathbf{f}_e. \tag{2.36}$$

For the electromagnetic force equation 2.4, often $\underline{\|\mathbf{E}| \gg \|\mathbf{v} \times \mathbf{B}\|}$, which implies

$$\mathbf{J}_{ec} \approx \kappa_e \mathbf{E}. \tag{2.37}$$

Free electric current can be further classified as impressed or conduction

$$\tilde{\mathbf{J}}_{ef} = \tilde{\mathbf{J}}_{ei} + \tilde{\mathbf{J}}_{ec} \tag{2.38}$$

where the conduction current phasor is

$$\tilde{\mathbf{J}}_{ec} = \kappa_e \tilde{\mathbf{E}}. \tag{2.39}$$

Insert equations 2.38 and 2.39 into Ampère's law

$$\nabla \times \tilde{\mathbf{H}} = [\tilde{\mathbf{J}}_{ei} + [\kappa_e \tilde{\mathbf{E}}]] + j\omega\epsilon\tilde{\mathbf{E}} \tag{2.40}$$

and collect terms

$$\nabla \times \tilde{\mathbf{H}} = \tilde{\mathbf{J}}_{ei} + j\omega \left[\epsilon + \frac{\kappa_e}{j\omega}\right] \tilde{\mathbf{E}}. \tag{2.41}$$

Free electric charge density can also be further classified as impressed or conduction

$$\tilde{\rho}_{ef} = \tilde{\rho}_{ei} + \tilde{\rho}_{ec} \tag{2.42}$$

where conduction charge is related to conduction current by the continuity equation

$$\nabla \cdot \tilde{\mathbf{J}}_{ec} = -j\omega\tilde{\rho}_{ec}. \tag{2.43}$$

Insert equations 2.42, 2.43, and 2.39 into Gauss's law

$$\nabla \cdot [\epsilon\tilde{\mathbf{E}}] = \left[\tilde{\rho}_{ei} + \left[-\frac{1}{j\omega}\nabla \cdot [\kappa_e \tilde{\mathbf{E}}]\right]\right] \tag{2.44}$$

and collect terms

$$\nabla \cdot \left[\left[\epsilon + \frac{\kappa_e}{j\omega} \right] \tilde{\mathbf{E}} \right] = \tilde{\rho}_{ei}. \tag{2.45}$$

Equations 2.41 and 2.45 motivate the definition of complex permittivity

$$\tilde{\epsilon} \equiv \epsilon + \frac{\kappa_e}{j\omega}. \tag{2.46}$$

Similar logic for magnetic sources motivates the definition of complex permeability

$$\tilde{\mu} \equiv \mu + \frac{\kappa_m}{j\omega}. \tag{2.47}$$

The time-harmonic Maxwell's equations in isotropic complex matter are then

$$\begin{cases} \nabla \times \tilde{\mathbf{E}} = -\tilde{\mathbf{J}}_{mi} - j\omega\tilde{\mu}\tilde{\mathbf{H}} \\ \nabla \times \tilde{\mathbf{H}} = \tilde{\mathbf{J}}_{ei} + j\omega\tilde{\epsilon}\tilde{\mathbf{E}} \\ \nabla \cdot [\tilde{\epsilon}\tilde{\mathbf{E}}] = \tilde{\rho}_{ei} \\ \nabla \cdot [\tilde{\mu}\tilde{\mathbf{H}}] = \tilde{\rho}_{mi} \end{cases} \quad \text{BCs:} \quad \begin{cases} \hat{\mathbf{n}} \times \Delta\tilde{\mathbf{E}} = -\tilde{\mathbf{K}}_{mi} \\ \hat{\mathbf{n}} \times \Delta\tilde{\mathbf{H}} = \tilde{\mathbf{K}}_{ei} \\ \hat{\mathbf{n}} \cdot \Delta[\tilde{\epsilon}\tilde{\mathbf{E}}] = \tilde{\sigma}_{ei} \\ \hat{\mathbf{n}} \cdot \Delta[\tilde{\mu}\tilde{\mathbf{H}}] = \tilde{\sigma}_{mi} \end{cases} \tag{2.48}$$

This form of Maxwell's equations are the foundation for theory presented in this work. Several assumptions were made to derive these equations from their general form. Specifically, these equations apply to linear non-magnetoelectric isotropic time-invariant media when $\|\tilde{\mathbf{E}}\| \gg \|\mathbf{v} \times \tilde{\mathbf{B}}\|$. From now on all electromagnetic quantities are assumed to be complex-valued unless otherwise stated. The complex notation will be suppressed when convenient.

2.2 Reciprocity

The most basic function of the metaimager is to illuminate a scene with a transmit antenna and measure the reflection with a receive antenna. It is therefore important

to model the coupling between the antennas and scene when formulating the forward model. A key theorem in understanding this coupling is electromagnetic reciprocity.

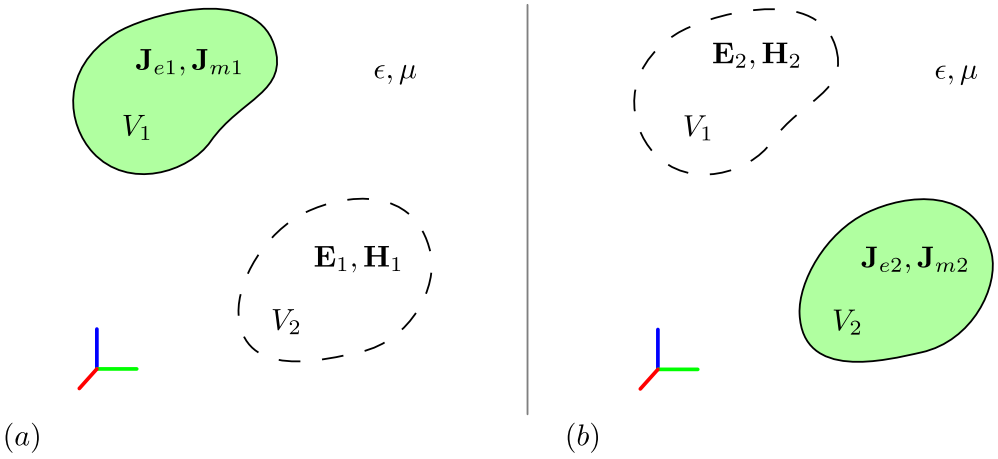

Figure 2.2: Reciprocity - A scene is defined by $\epsilon(\mathbf{r})$ and $\mu(\mathbf{r})$. (a) A set of source currents in V_1 excite fields in V_2. (b) A set of source currents in V_2 excite fields in V_1.

Consider a scene with two different configurations involving the same media but different sources and fields (figure 2.2). The sources and fields of configuration 1 are $\mathbf{J}_{e1}$, $\mathbf{J}_{m1}$, $\mathbf{E}_1$, $\mathbf{H}_1$, and configuration 2 are $\mathbf{J}_{e2}$, $\mathbf{J}_{m2}$, $\mathbf{E}_2$, $\mathbf{H}_2$. Both configurations satisfy Maxwell's curl equations 2.48

$$\begin{cases} \nabla \times \mathbf{E}_1 = -\mathbf{J}_{m1} - j\omega\mu\mathbf{H}_1 \\ \nabla \times \mathbf{H}_1 = \mathbf{J}_{e1} + j\omega\epsilon\mathbf{E}_1 \end{cases} \tag{2.49}$$

$$\begin{cases} \nabla \times \mathbf{E}_2 = -\mathbf{J}_{m2} - j\omega\mu\mathbf{H}_2 \\ \nabla \times \mathbf{H}_2 = \mathbf{J}_{e2} + j\omega\epsilon\mathbf{E}_2 \end{cases} . \tag{2.50}$$

These two systems of equations can be combined with a vector calculus product rule

$$\nabla \cdot [\mathbf{A} \times \mathbf{B}] = \mathbf{B} \cdot \nabla \times \mathbf{A} - \mathbf{A} \cdot \nabla \times \mathbf{B} \tag{2.51}$$

by taking the appropriate dot products

$$\begin{cases} \mathbf{H}_2 \cdot \nabla \times \mathbf{E}_1 = -\mathbf{H}_2 \cdot \mathbf{J}_{m1} - j\omega \mathbf{H}_2 \cdot \mu \mathbf{H}_1 \\ \mathbf{E}_2 \cdot \nabla \times \mathbf{H}_1 = \mathbf{E}_2 \cdot \mathbf{J}_{e1} + j\omega \mathbf{E}_2 \cdot \epsilon \mathbf{E}_1 \\ \mathbf{H}_1 \cdot \nabla \times \mathbf{E}_2 = -\mathbf{H}_1 \cdot \mathbf{J}_{m2} - j\omega \mathbf{H}_1 \cdot \mu \mathbf{H}_2 \\ \mathbf{E}_1 \cdot \nabla \times \mathbf{H}_2 = \mathbf{E}_1 \cdot \mathbf{J}_{e2} + j\omega \mathbf{E}_1 \cdot \epsilon \mathbf{E}_2 \end{cases} \tag{2.52}$$

resulting in

$$\begin{cases} \nabla \cdot [\mathbf{E}_1 \times \mathbf{H}_2] = -\mathbf{H}_2 \cdot \mathbf{J}_{m1} - j\omega \mathbf{H}_2 \cdot \mu \mathbf{H}_1 - \mathbf{E}_1 \cdot \mathbf{J}_{e2} - j\omega \mathbf{E}_1 \cdot \epsilon \mathbf{E}_2 \\ \nabla \cdot [\mathbf{E}_2 \times \mathbf{H}_1] = -\mathbf{H}_1 \cdot \mathbf{J}_{m2} - j\omega \mathbf{H}_1 \cdot \mu \mathbf{H}_2 - \mathbf{E}_2 \cdot \mathbf{J}_{e1} - j\omega \mathbf{E}_2 \cdot \epsilon \mathbf{E}_1 \end{cases} . \tag{2.53}$$

Combine the remaining equations by taking the difference

$$-\nabla \cdot [\mathbf{E}_1 \times \mathbf{H}_2 - \mathbf{E}_2 \times \mathbf{H}_1] = [\mathbf{E}_1 \cdot \mathbf{J}_{e2} - \mathbf{H}_1 \cdot \mathbf{J}_{m2}] - [\mathbf{E}_2 \cdot \mathbf{J}_{e1} - \mathbf{H}_2 \cdot \mathbf{J}_{m1}]. \tag{2.54}$$

Integrate over volume V and apply the divergence theorem to yield the general time-harmonic reciprocity theorem in integral form

$$-\oint_S [\mathbf{E}_1 \times \mathbf{H}_2 - \mathbf{E}_2 \times \mathbf{H}_1] \cdot d\mathbf{S} = \int_V [[\mathbf{E}_1 \cdot \mathbf{J}_{e2} - \mathbf{H}_1 \cdot \mathbf{J}_{m2}] - [\mathbf{E}_2 \cdot \mathbf{J}_{e1} - \mathbf{H}_2 \cdot \mathbf{J}_{m1}]] dV. \tag{2.55}$$

Fields of sources with finite support obey the Sommerfeld radiation conditions

$$\lim_{r \to \infty} r[\mathbf{H} - \eta^{-1} \hat{\mathbf{r}} \times \mathbf{E}] = 0 \tag{2.56}$$

$$\lim_{r \to \infty} r[\mathbf{E} + \eta \hat{\mathbf{r}} \times \mathbf{H}] = 0 \tag{2.57}$$

which implies the surface integral vanishes when $V = \mathbb{R}^3$. Furthermore, the finite support of the sources limits the integration domains. Therefore

$$\boxed{\int_{V_2} [\mathbf{E}_1 \cdot \mathbf{J}_{e2} - \mathbf{H}_1 \cdot \mathbf{J}_{m2}] dV = \int_{V_1} [\mathbf{E}_2 \cdot \mathbf{J}_{e1} - \mathbf{H}_2 \cdot \mathbf{J}_{m1}] dV.} \tag{2.58}$$

This is reciprocity for sources with finite support. The sources do not have to simultaneously exist, so this equation cannot be interpreted as a power. However it can be used to compute the coupling between an antenna and scene as will be demonstrated.

2.3 Scene Model

In general the material properties of a scene can be very complicated. Reciprocity indicates a need to model currents in a scene. Fortunately several simplifying assumptions can be made. First, the assumptions built into the time-harmonic Maxwell's equations for complex media help considerably. Secondly, (after background subtraction) all non-physical magnetic currents can be ignored.

Total electric volume current density is expanded from equations 2.6 and 2.38 as

$$\mathbf{J}_e = \mathbf{J}_{ei} + \mathbf{J}_{ec} + \mathbf{J}_{eb} + \mathbf{J}_{ep} \tag{2.59}$$

where the subscripts are for impressed, conduction, bound, and polarization currents. Assume the scene has no impressed electric volume current density

$$\mathbf{J}_{ei} = \mathbf{0} \tag{2.60}$$

Conduction current was discussed in the process of deriving the time-harmonic Maxwell's equations. Ohm's law was approximated as

$$\mathbf{J}_{ec} \approx \kappa_e \mathbf{E} \tag{2.61}$$

Materials give rise to additional types of current. These currents are embedded in the definitions of $\mathbf{D}$ and $\mathbf{H}$. The bound electric volume current density is

$$\mathbf{J}_{eb} = \nabla \times \mathbf{M}. \tag{2.62}$$

For isotropic media

$$\mathbf{M} = \chi_m \mathbf{H}. \tag{2.63}$$

This work assumes non-magnetic scenes with $\chi_m = 0$ and thus $\mathbf{M} = \mathbf{0}$ and $\mathbf{J}_{eb} = \mathbf{0}$. The polarization electric volume current density is

$$\mathbf{J}_{ep} = j\omega \mathbf{P}. \tag{2.64}$$

For isotropic media

$$\mathbf{P} = \epsilon_0 \chi_e \mathbf{E}. \tag{2.65}$$

Combining the above equations and assumptions, the total scene current is

$$\mathbf{J}_e = \mathbf{J}_{ec} + \mathbf{J}_{ep} = [\kappa_e \mathbf{E}] + [j\omega[\epsilon_0 \chi_e \mathbf{E}]] = [\kappa_e + j\omega\epsilon_0\chi_e]\mathbf{E} \tag{2.66}$$

Define the complex conductivity such that (complex notation emphasized)

$$\boxed{\tilde{\mathbf{J}}_e = \tilde{\kappa}_e \tilde{\mathbf{E}}} \tag{2.67}$$

$$\boxed{\tilde{\kappa}_e \equiv \kappa_e + j\omega\epsilon_0\chi_e} \tag{2.68}$$

This is similar but distinct from the complex permittivity $\tilde{\epsilon}$. The real part of $\tilde{\kappa}_e$ represents electric conduction currents that move in phase with the fields, and the imaginary part represents dielectric polarization currents that move a quarter turn out of phase.

2.4 Antenna Model

Modeling the measurement process requires knowledge of the fields radiated by antennas at arbitrary positions in space. When an antenna port is driven with a time-harmonic reference potential V_0, electric currents are driven on the surface of the antenna that radiate electromagnetic fields into space. The details of these physical currents are difficult to predict and measure. However it is easy to measure the radiating fields on a plane and relate them to fictitious sources that radiate equivalent fields. This is an application of the equivalence principle.

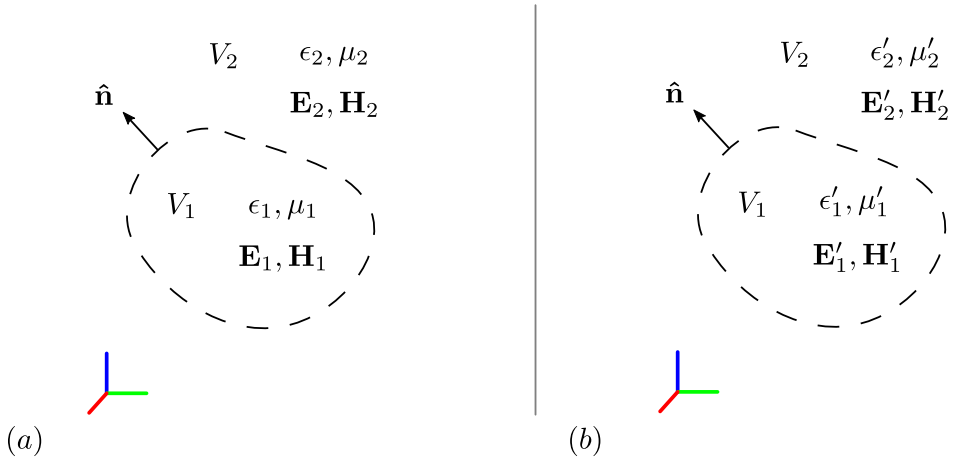

Figure 2.3: Equivalence Principle - (a) Original problem. (b) Equivalent problem.

The general idea of the equivalence principle is to transform a problem into an equivalent problem that preserves desired field quantities and facilitates theoretical development. There are many useful transformations and applications. Consider the equivalence principle when space is partitioned into volumes V_1 and V_2 as illustrated

in figure 2.3. The original problem is transformed into an equivalent problem

$$(\epsilon_1, \mu_1, \mathbf{E}_1, \mathbf{H}_1, \epsilon_2, \mu_2, \mathbf{E}_2, \mathbf{H}_2) \Rightarrow (\epsilon_1', \mu_1', \mathbf{E}_1', \mathbf{H}_1', \epsilon_2', \mu_2', \mathbf{E}_2', \mathbf{H}_2') \tag{2.69}$$

and the boundary conditions are necessarily transformed as well

$$\begin{cases} \hat{\mathbf{n}} \times \Delta\mathbf{E} = -\mathbf{K}_{mi} \\ \hat{\mathbf{n}} \times \Delta\mathbf{H} = \mathbf{K}_{ei} \end{cases} \Rightarrow \begin{cases} \hat{\mathbf{n}} \times \Delta\mathbf{E}' = -\mathbf{K}'_{mi} \\ \hat{\mathbf{n}} \times \Delta\mathbf{H}' = \mathbf{K}'_{ei} \end{cases} . \tag{2.70}$$

Usually the quantities in one volume are changed and the other unchanged.

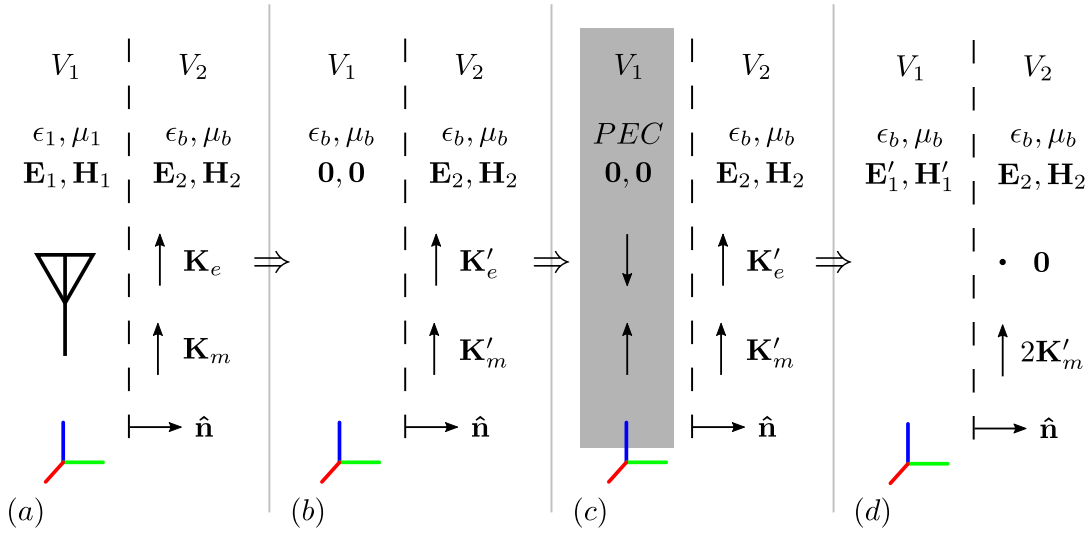

Figure 2.4: Antenna Model - (a) physical situation, (b) Love's equivalence, (c) Love's equivalence for a PEC plane, (d) equivalent source currents.

The antenna model is derived in figure 2.4. First consider the physical situation of an antenna radiating fields into a homogeneous background medium ϵ_b, μ_b. Partition space with a planar surface into volume V_1 that fully contains the antenna and volume V_2. Suppose only fields in V_2 are needed. Apply Love's equivalence to set the fields

in V_1 to zero

$$(\epsilon_1, \mu_1, \mathbf{E}_1, \mathbf{H}_1, \epsilon_b, \mu_b, \mathbf{E}_2, \mathbf{H}_2) \Rightarrow (\epsilon_b, \mu_b, \mathbf{0}, \mathbf{0}, \epsilon_b, \mu_b, \mathbf{E}_2, \mathbf{H}_2) \tag{2.71}$$

which is possible by satisfying the boundary conditions with fictitious surface currents

$$\begin{cases} \hat{\mathbf{n}} \times \mathbf{E}_2 = -\mathbf{K}'_{mi} \\ \hat{\mathbf{n}} \times \mathbf{H}_2 = \mathbf{K}'_{ei} \end{cases}. \tag{2.72}$$

Because the fields are zero in V_1, the material properties of V_1 are arbitrary. Consider Love's equivalence for a perfect electric conductor (PEC) plane in which V_1 is filled with a PEC plane set back a small distance from the boundary plane. In the limit the distance goes to zero, image theory (appendix A) predicts electric currents won't radiate while magnetic currents radiate double. Image theory makes this exact when used to replace the PEC with the background medium

$$\begin{cases} 2\hat{\mathbf{n}} \times \mathbf{E}_2 = -\mathbf{K}''_{mi} \\ \hat{\mathbf{n}} \times \mathbf{H}_2 = \mathbf{0} \end{cases}. \tag{2.73}$$

Thus, if an antenna's electric field is known on a plane, the antenna can be modeled as an equivalent magnetic surface current density radiating in a homogeneous background medium

$$\boxed{\mathbf{K}_{mi}(\mathbf{r}) = -2\hat{\mathbf{n}} \times \mathbf{E}(\mathbf{r}).} \tag{2.74}$$

The electric field of an antenna is estimated by near-field scan (NFS) using a vector network analyzer (VNA) to measure S-parameters in terms of reference voltage V_0

$$\boxed{\mathbf{E}_{NFS}(\mathbf{r}) = E_0 \mathbf{S}_{NFS}(\mathbf{r}) = V_0 \Gamma \mathbf{S}_{NFS}(\mathbf{r}).} \tag{2.75}$$

The quality of this model partially depends on the size of the scan.

2.5 Radiation Model

The antenna model results in source currents radiating in a homogeneous isotropic background material. An analytic field solution exists for this configuration, but the derivation is complex and left to appendix A. Potentials are used to simplify the treatment. The result is summarized here.

Fields of arbitrary volume sources in homogeneous isotropic media are

$$\mathbf{E}_e(\mathbf{r}) = \frac{1}{j\omega\epsilon}\int_V [G_2\mathbf{J}_e(\mathbf{r}') + G_1\hat{\mathbf{R}}[\hat{\mathbf{R}}\cdot\mathbf{J}_e(\mathbf{r}')]]dV' - \frac{1}{j\omega\epsilon}\mathbf{J}_e(\mathbf{r}) \tag{2.76a}$$

$$\mathbf{E}_m(\mathbf{r}) = \int_V G_0\hat{\mathbf{R}}\times\mathbf{J}_m(\mathbf{r}')dV' \tag{2.76b}$$

$$\mathbf{H}_e(\mathbf{r}) = -\int_V G_0\hat{\mathbf{R}}\times\mathbf{J}_e(\mathbf{r}')dV' \tag{2.76c}$$

$$\mathbf{H}_m(\mathbf{r}) = \frac{1}{j\omega\mu}\int_V [G_2\mathbf{J}_m(\mathbf{r}') + G_1\hat{\mathbf{R}}[\hat{\mathbf{R}}\cdot\mathbf{J}_m(\mathbf{r}')]]dV' - \frac{1}{j\omega\mu}\mathbf{J}_m(\mathbf{r}) \tag{2.76d}$$

where

$$G_0 \equiv [4\pi]^{-1}[jkR^{-1} + R^{-2}]e^{-jkR} \tag{2.77a}$$

$$G_1 \equiv [4\pi]^{-1}[-k^2R^{-1} + j3kR^{-2} + 3R^{-3}]e^{-jkR} \tag{2.77b}$$

$$G_2 \equiv [4\pi]^{-1}[k^2R^{-1} - jkR^{-2} - R^{-3}]e^{-jkR} \tag{2.77c}$$

.

The antenna model only has magnetic current sources, and the scene model is excited only by electric fields, so only one radiation equation is required

$$\mathbf{E}_m(\mathbf{r}) = \int_V \frac{1}{4\pi}\left[\frac{jk}{R} + \frac{1}{R^2}\right]e^{-jkR}\hat{\mathbf{R}}\times\mathbf{J}_m(\mathbf{r}')dV' \tag{2.78}$$

or in terms of the magnetic surface currents of the antenna model

$$\mathbf{E}_m(\mathbf{r}) = \int_S \frac{1}{4\pi}\left[\frac{jk}{R} + \frac{1}{R^2}\right]e^{-jkR}\hat{\mathbf{R}}\times\mathbf{K}_m(\mathbf{r}')dS'. \tag{2.79}$$

It is sufficient to keep only the lowest order term

$$\boxed{\mathbf{E}_m(\mathbf{r}) \approx \frac{jk}{4\pi}\int_S \frac{\mathbf{R}}{R^2} e^{-jkR} \times \mathbf{K}_m(\mathbf{r}')dS'.} \tag{2.80}$$

Practically the radiation model is in terms of S-parameters

$$\boxed{\mathbf{S}(\mathbf{r}) \approx \frac{jk}{4\pi}\int_S \frac{\mathbf{R}}{R^2} e^{-jkR} \times \left[-2\hat{\mathbf{n}} \times \mathbf{S}_{NFS}(\mathbf{r}')\right] dS'.} \tag{2.81}$$

If the radiation patterns are linearly polarized and the scene doesn't appreciably rotate polarization, the model can be approximated by a single component which can save on computational time.

2.6 Imaging Model

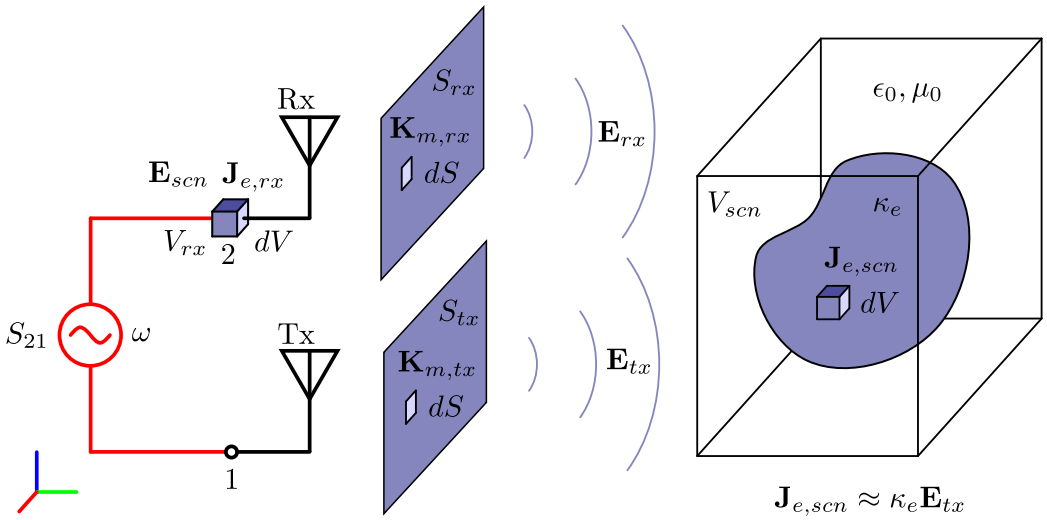

Figure 2.5: Metaimager Forward Model

All of the tools needed to derive the metaimager forward model (figure 2.5) have been assembled. The basic idea is a of form scattering problem, which can be formu-

lated in terms of total, incident, and scattered fields

$$\mathbf{E} = \mathbf{E}_{inc} + \mathbf{E}_{sct} \tag{2.82}$$

where the incident field is the field when only background is present. In the case of the metaimager, the incident field is the measured transmit field

$$\boxed{\mathbf{E}_{inc} = \mathbf{E}_{tx}.} \tag{2.83}$$

The total and scattered fields are difficult to compute because the incident field excites scene currents which radiate and cause higher-order scattering. A great simplification can be made for scenes with weak scattering by approximating the excited currents as depending only on the incident field and not the total field

$$\boxed{\mathbf{J}_{e,scn}(\mathbf{E}) \approx \mathbf{J}_{e,scn}(\mathbf{E}_{inc}).} \tag{2.84}$$

This is called the first Born approximation for being the first term in the Born series. Ironically this approximation also works for strong reflections so long as multiple reflections are minimized. This important approximation makes the subsequent forward model linear, and thus amiable to solution.

Consider reciprocity for finite sources

$$\int_{V_2} [\mathbf{E}_1 \cdot \mathbf{J}_{e2} - \mathbf{H}_1 \cdot \mathbf{J}_{m2}] dV = \int_{V_1} [\mathbf{E}_2 \cdot \mathbf{J}_{e1} - \mathbf{H}_2 \cdot \mathbf{J}_{m1}] dV \tag{2.85}$$

where V_1 is scene volume V_{scn}, and V_2 is receive volume V_{rx} enclosing a thin slice of transmission line at receive port 2. After background subtraction , the only sources to consider are electric sources

$$\int_{V_{rx}} \tilde{\mathbf{E}}_{scn} \cdot \tilde{\mathbf{J}}_{e,rx} dV = \int_{V_{scn}} \tilde{\mathbf{E}}_{rx} \cdot \tilde{\mathbf{J}}_{e,scn} dV. \tag{2.86}$$

The left integral encloses a plane of coax transmission line and can be reduced to a circuit-theoretic form, while equations 2.67 and 2.84 can be inserted into the right integral

$$\tilde{V}_{scn}\tilde{I}_{e,rx} = \int_{V_{scn}} \tilde{\mathbf{E}}_{rx} \cdot [\tilde{\kappa}_e \tilde{\mathbf{E}}_{tx}] dV. \tag{2.87}$$

Isolate the voltage at the recieve port due to the scene sources

$$\boxed{\tilde{V}_{scn} = \frac{1}{\tilde{I}_{e,rx}} \int_{V_{scn}} \tilde{\mathbf{E}}_{rx}(\mathbf{r}) \cdot \tilde{\kappa_e}(\mathbf{r}) \tilde{\mathbf{E}}_{tx}(\mathbf{r}) dV.} \tag{2.88}$$

This equation is similar to the model proposed in. However a more appropriate form incorporates equation 2.75

$$\tilde{V}_{scn} = \frac{1}{\tilde{I}_{e,rx}} \int_{V_{scn}} [\tilde{V}_{rx}\Gamma_{rx}\tilde{\mathbf{S}}_{rx}] \cdot \tilde{\kappa}_e [\tilde{V}_{tx}\Gamma_{tx}\tilde{\mathbf{S}}_{tx}] dV. \tag{2.89}$$

Use the identity $\tilde{Z}_{rx} = \tilde{V}_{rx}/\tilde{I}_{rx}$ and factor out the reference transmit voltage $\tilde{V}_{tx}$

$$\tilde{V}_{scn} = \left[\int_{V_{scn}} [\tilde{Z}_{rx}\Gamma_{rx}\tilde{\mathbf{S}}_{rx}] \cdot \tilde{\kappa}_e [\Gamma_{tx}\tilde{\mathbf{S}}_{tx}] dV \right] \tilde{V}_{tx}. \tag{2.90}$$

This equation looks like an S-parameter measurement between the transceiver ports

$$\tilde{V}_{scn} = \tilde{S}_{21}\tilde{V}_{tx} \tag{2.91}$$

where port 1 is the transmit antenna, port 2 is the receive antenna, and $\tilde{S}_{21}$ is

$$\boxed{\tilde{S}_{21} = \int_{V_{scn}} [\tilde{Z}_{rx}\Gamma_{rx}\tilde{\mathbf{S}}_{rx}(\mathbf{r})] \cdot \tilde{\kappa}_e(\mathbf{r}) [\Gamma_{tx}\tilde{\mathbf{S}}_{tx}(\mathbf{r})] dV.} \tag{2.92}$$

Z_{rx} is the receive antenna impedance and can be measured with a VNA using reflection measurements. Γ_{rx} and Γ_{tx} are real-valued geometric factors based on the NFS probe. From experience, the forward model is still valid even when $\tilde{Z}_{rx}$, Γ_{rx} and Γ_{tx} are unaccounted for, although the interpretation of images is somewhat ambiguous. Using slightly different notation to indicate this case, the forward model becomes

$$g_i = \int_{V_{scn}} \mathbf{S}_{rx,i}(\mathbf{r}) \cdot f(\mathbf{r}) \mathbf{S}_{tx,i}(\mathbf{r}) dV. \tag{2.93}$$

Approximate this integral by a discrete sum

$$g_i = \sum_j \mathbf{S}_{rx,i}(\mathbf{r}_j) \cdot f(\mathbf{r}_j)\mathbf{S}_{tx,i}(\mathbf{r}_j)\Delta V_j. \tag{2.94}$$

For isotropic media this sum can be written as

$$\boxed{g_i = \sum_j H_{ij} f(\mathbf{r}_j)} \tag{2.95}$$

where

$$\boxed{H_{ij} \equiv \mathbf{S}_{rx,i}(\mathbf{r}_j) \cdot \mathbf{S}_{tx,i}(\mathbf{r}_j)\Delta V_j} \tag{2.96}$$

or in matrix notation

$$\boxed{\mathbf{g} = \mathbf{H}\mathbf{f}.} \tag{2.97}$$

Thus, the metaimager forward model can be reduced to a linear equation.

Chapter 3

Image Reconstruction

The metaimager forward model results in a complex-valued linear system

$$\mathbf{Hf} = \mathbf{g} \tag{3.1}$$

where sensing matrix $\mathbf{H} \in \mathbb{C}^{m \times n} : \mathbb{C}^n \to \mathbb{C}^m$ relates a vector of scene voxel scattering coefficients $\mathbf{f} \in \mathbb{C}^n$ to a vector of transceiver S_{21} measurements $\mathbf{g} \in \mathbb{C}^m$. The **image reconstruction problem** is the task of, given a measurement vector $\mathbf{g}$, estimating the associated scene vector $\mathbf{f}$ by inverting $\mathbf{H}$. In general this is an ill-posed problem, and $\mathbf{H}^{-1}$ does not usually exist.

This chapter discusses a variety of methods and algorithms used to meaningfully estimate a solution to the image reconstruction problem, even when no exact solution exists. Basic space and time complexity analysis for the various methods are discussed.

Matrix inversion is a general problem in linear algebra, and a notation different from the forward model is adopted in the literature. Let $\mathbf{A} = \mathbf{H}$, $\mathbf{x} = \mathbf{f}$, and $\mathbf{b} = \mathbf{g}$ such that

$$\mathbf{Ax} = \mathbf{b}. \tag{3.2}$$

The inverse problem then becomes given matrix $\mathbf{A}$ and vector $\mathbf{b}$, find vector $\mathbf{x}$. A necessary but insufficient condition for $\mathbf{A}^{-1}$ to exist is that $\mathbf{A}$ is square with $m = n$. This almost never happens in practice. More often $m \neq n$. When $n < m$ (more measurements than voxels), then the system is over-determined (tall). If there $m > n$

(more voxels than measurements), then the system under-determined (wide).

An under-determined image reconstruction problem corresponds to compressed sensing (CS). Such a linear system has an infinite number of solutions. To pick a unique solution, prior information must be incorporated into the reconstruction model. For instance, the number of non-zero voxels can be penalized for sparse scenes.

For general computational imaging applications it is desirable to approximately match the number of measurements with the number of reconstruction voxels. This is to mitigate the need for a prior, and to minimize the size of the inverse problem.

This chapter begins with the SVD pseudoinverse, which is theoretically interesting, but computationally expensive to compute. Then the very simple matched filter is introduced, which relies on the statistical properties of a random system with a random matrix. Next complex least squares with regularization is derived to cast any ill-posed linear inverse problem into a well-posed invertible inverse problem with a unique solution. Three iterative methods are then derived to solve the least-squares problem with better computationally complexity than more direct inverse methods.

3.1 SVD Pseudoinverse

Every matrix has a singular value decomposition (SVD) given by

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \tag{3.3}$$

where $\mathbf{U} \in \mathbb{C}^{m\times m}$ and $\mathbf{V} \in \mathbb{C}^{n\times n}$ are unitary matrices and $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is a diagonal matrix. The elements $\sigma_i \in \mathbb{R}^{+0}$ on the diagonal of $\mathbf{\Sigma}$ are called singular values, the whole collection of singular values is called the singular value spectrum, and

the spectrum is customarily arranged in descending order. The SVD indicates that any linear transformation can be thought of as norm-preserving unitary "rotation", followed by a per-component scaling, and then one more "rotation".

The matrix factors of can be used to form an **SVD pseudoinverse**. Unitary matrices are defined by the property $\mathbf{Q}^{-1} = \mathbf{Q}^H$, which apply to $\mathbf{U}$ and $\mathbf{V}$. Slightly more care must be taken to define the inverse of $\mathbf{\Sigma}$ because singular values may be 0. Define the matrix $\mathbf{\Sigma}^+ \in \mathbb{R}^{n \times m}$ to be the diagonal matrix where the diagonal elements are the reciprocals of non-zero singular values or zero otherwise. Then the SVD pseudoinverse is defined as

$$\mathbf{A}^+ \equiv \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^H. \tag{3.4}$$

Strictly speaking, $\mathbf{A}^{-1}$ does not exist, but $\mathbf{A}^+$ shares features that are desired in an inverse. These are codified in the properties of the Moore-Penrose pseudoinverse. An estimate for the solution to the linear system is then found to be

$$\boxed{\mathbf{x}_{pinv} = \mathbf{A}^+\mathbf{b}}. \tag{3.5}$$

It is often useful to modify the pseudoinverse by truncating the singular value spectrum below a threshold. This can help prevent extremely small singular values and nullspaces from magnifying noise and numeric errors to problematic levels. This has the side benefit of making the inverse problem smaller. The singular value spectrum also reveals important details about the information content of a linear system, and informs the design process in several ways, from minimizing correlation between measurements, to maximizing total number of useful elements, etc.

Algorithms for computing the SVD of a matrix are outside the scope of this text. State of the art algorithms have time complexity of $O(km^2n+k'n^3)$ for some constants

$k, k' \in \mathbb{R}$. This is prohibitively expensive for very large linear systems that have real time constraints, and an alternative reconstruction strategies must be found.

3.2 Matched Filter

The simplest image reconstruction method is **matched filter**. Matched filter reconstruction exploits the statistical nature of a random matrix $\mathbf{A}$ to formulate a computationally inexpensive approximate inverse. The key idea is that large random vectors with elements pulled from a zero-mean distribution are on average orthogonal to each other and the same length. This applies to the columns of $\mathbf{A}$. Premultiplying $\mathbf{A}$ by its conjugate transpose then should very nearly equal the identity matrix scaled by some real constant $k \in \mathbb{R}$

$$\mathbf{A}^H\mathbf{A} \approx k\mathbf{I} \tag{3.6}$$

and thus the solution is approximated by

$$\mathbf{x} \approx \frac{1}{k}\mathbf{A}^H\mathbf{b}. \tag{3.7}$$

The behavior of matched filter reconstruction is understood by its statistical properties. The real case will be analyzed first and then extended to the complex case.

Some basic statistical theorems are needed to study the matched filter. A continuous random variable $x \in \mathbb{R}$ is associated with a continuous probability distribution $p : \mathbb{R} \to \mathbb{R}$ for which the integral over all values must equal 1

$$\int_{-\infty}^{\infty} p(x)dx = 1. \tag{3.8}$$

The average of a real function $f : \mathbb{R} \to \mathbb{R}$ of a real random variable $x \in \mathbb{R}$ is then

$$\langle f(x) \rangle = \int_{-\infty}^{\infty} f(x)p(x)dx. \tag{3.9}$$

For functions of two random variables, a joint probability function $p_{xy} : \mathbb{R}^2 \to \mathbb{R}$ must be specified, and this distribution must also integrate over all values to 1

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{xy}(x,y)dxdy = 1. \tag{3.10}$$

The average of a real function $f : \mathbb{R}^2 \to \mathbb{R}$ of two real random variables $x, y \in \mathbb{R}$ is then

$$\langle f(x,y)\rangle = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)p_{xy}(x,y)dxdy. \tag{3.11}$$

When random variables are independent, their joint probability has the special form

$$p_{xy}(x,y) = p_x(x)p_y(y). \tag{3.12}$$

The average value of independent random variables have some nice properties. First, the average of the sum or independent random variables is the sum of the average of the random variables

$$\begin{aligned}\langle x+y\rangle &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[x+y][p_x(x)p_y(y)]dxdy \\ &= \int_{-\infty}^{\infty} xp_x(x)dx\int_{-\infty}^{\infty} p_y(y)dy + \int_{-\infty}^{\infty} p_x(x)dx\int_{-\infty}^{\infty} yp_y(y)dy \\ &= \langle x\rangle + \langle y\rangle .\end{aligned} \tag{3.13}$$

Secondly, the average value of the product of independent random variables is the product of the average value of those variables

$$\langle xy\rangle = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[xy][p_x(x)p_y(y)]dxdy = \int_{-\infty}^{\infty} xp_x(x)dx\int_{-\infty}^{\infty} yp_y(y)dy = \langle x\rangle\langle y\rangle . \tag{3.14}$$

Now consider the real random matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with independent and identically distributed elements drawn from a zero-mean normal distribution with variance σ^2

$$p(x) = N(x;0,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{[x-0]^2}{2\sigma^2}}. \tag{3.15}$$

Elements of matrix $\mathbf{A}^T\mathbf{A}$ can be classified as either on or off the main diagonal. On-diagonal elements are the inner products of columns with themselves, and off-diagonal elements are the inner products between different columns. First consider the on-diagonal elements. A column of $\mathbf{A}$ is a random vector $\mathbf{x} \in \mathbb{R}^m$. Therefore the average of an on-diagonal element is the average of the inner product of $\mathbf{x}$ with itself

$$\left\langle \mathbf{x}^T\mathbf{x} \right\rangle = \left\langle \sum_{i=1}^{m} x_i^2 \right\rangle = \sum_{i=1}^{m} \left\langle x_i^2 \right\rangle = \left\langle x^2 \right\rangle \sum_{i=1}^{m} 1 = m \left\langle x^2 \right\rangle . \tag{3.16}$$

The average of x^2 is

$$\left\langle x^2 \right\rangle = \int_{-\infty}^{\infty} \left[x^2\right] \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}\right] dx = \frac{2}{\sqrt{2\pi}\sigma} \int_0^{\infty} x^2 e^{-\frac{x^2}{2\sigma^2}} dx = \frac{2}{\sqrt{2\pi}\sigma} \frac{2\sigma^2}{4} \sqrt{2\pi\sigma^2} = \sigma^2 \tag{3.17}$$

where the following identity is used

$$\int_0^{\infty} x^{2n} e^{-ax^2} dx = \frac{[2n-1]!!}{a^n 2^{n+1}} \sqrt{\frac{\pi}{a}}. \tag{3.18}$$

Combining these results it is found that the average of on-diagonal elements is

$$\left\langle \mathbf{x}^T\mathbf{x} \right\rangle = m\sigma^2. \tag{3.19}$$

Now consider off-diagonal elements. In this case the inner product of two independent column vectors $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{x}' \in \mathbb{R}^m$ is formed. Taking the average it is found

$$\left\langle \mathbf{x}^T\mathbf{x}' \right\rangle = \left\langle \sum_{i=1}^{m} x_i x_i' \right\rangle = \sum_{i=1}^{m} \left\langle x_i \right\rangle \left\langle x_i' \right\rangle = \left\langle x \right\rangle \left\langle x' \right\rangle \sum_{i=1}^{m} 1 = m \left\langle x \right\rangle \left\langle x' \right\rangle . \tag{3.20}$$

By assumption the elements are drawn from a zero-mean distribution

$$\left\langle x \right\rangle = \int_{-\infty}^{\infty} [x] \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}\right] dx = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} x e^{-\frac{x^2}{2\sigma^2}} dx = 0 \tag{3.21}$$

the last step following by integrating an odd function over an even interval. This means that the average off-diagonal element is 0

$$\left\langle \mathbf{x}^T\mathbf{x}' \right\rangle = 0. \tag{3.22}$$

Putting the two cases together, it is found that $k = m\sigma^2$ and

$$\left\langle \mathbf{A}^T\mathbf{A} \right\rangle = m\sigma^2\mathbf{I}. \tag{3.23}$$

Extend this result to the complex case by considering complex random matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ with elements in rectangular form

$$z = x + jy \tag{3.24}$$

where the continuous probability distributions for the real and imaginary parts are given as $p_x(x) = n(x; 0, \sigma/\sqrt{2})$ and $p_y(y) = n(y; 0, \sigma/\sqrt{2})$. Consider the on-diagonal elements of $\mathbf{A}^H\mathbf{A}$. The square in the real case is replaced with a complex modulus

$$\left\langle \mathbf{z}^H\mathbf{z} \right\rangle = m\left\langle x^2 + y^2 \right\rangle = m\left[\left\langle x^2 \right\rangle + \left\langle y^2 \right\rangle\right] = m\left[\frac{\sigma^2}{2} + \frac{\sigma^2}{2}\right] = m\sigma^2. \tag{3.25}$$

The motivation for multiplying the variance a factor of $1/\sqrt{2}$ is now apparent. Because $\langle z \rangle = 0$, the average of off-diagonal elements is

$$\left\langle \mathbf{z}^H\mathbf{z}' \right\rangle = 0. \tag{3.26}$$

Thus in the complex case the average of $\mathbf{A}^H\mathbf{A}$ is

$$\left\langle \mathbf{A}^H\mathbf{A} \right\rangle = m\sigma^2\mathbf{I} \tag{3.27}$$

which is the same as the real result save for the conjugate transpose.

The results of this section motivates the definition of the matched filter approximate solution

$$\mathbf{x}_{mf} \equiv \frac{1}{m\sigma^2}\mathbf{A}^H\mathbf{b}. \tag{3.28}$$

For imaging, often the scale of the image is unimportant. In this case the scalar factor can be ignored

$$\boxed{\mathbf{x}_{mf} = \mathbf{A}^H\mathbf{b}.} \tag{3.29}$$

MF works surprisingly well with minimal effort. it can be interpreted as correlating signals in the time domain, and this works because the columns of measurement matrix $\mathbf{H}$ are designed to be (ideally) uncorrelated. Assuming the inputs are already loaded in memory, the space complexity of matched filter is $O_s(n)$ because an output scene vector is needed. The time complexity is $O_t(mn)$ for the single matrix multiplication.

While matched filter does not produce the highest quality estimate for a solution, for the amount of computation it has adequate results. The first step of more sophisticated algorithms are often equivalent to matched filter. Note, the analysis presented in this section assumes all the elements of $\mathbf{A}$ are independent and identically distributed. This is only an approximation that is unlikely to be exactly realized in practice, but the key ideas remain mostly valid.

3.3 Complex Least Squares

Given matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$ and vector $\mathbf{b} \in \mathbb{C}^m$, find vector $\mathbf{x} \in \mathbb{C}^n$ such that

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3.30}$$

Usually this is an ill-posed problem having no solutions or infinite solutions.

While it is not always possible to satisfy Eq. 3.30, it is always possible to take the difference of both sides. Define the **residual** as

$$\boxed{\mathbf{r}(\mathbf{x}) \equiv \mathbf{b} - \mathbf{A}\mathbf{x}}. \tag{3.31}$$

A reasonable, although somewhat arbitrary, way to compare the fitness of any $\mathbf{x}$ as an approximate solution to Eq. 3.30 is by the residual norm (squared)

$$f(\mathbf{x}) = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 \tag{3.32}$$

where the norm is taken to be that induced by the standard complex inner product

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^H \mathbf{y} \quad \Rightarrow \quad \|\mathbf{x}\|^2 = \langle \mathbf{x}, \mathbf{x} \rangle = \mathbf{x}^H \mathbf{x}. \tag{3.33}$$

A smaller residual norm is interpreted as indicating a better approximation to Eq. 3.30, and zero residual norm an exact solution. Function $f : \mathbb{C}^n \to \mathbb{R}$ is a **cost function** for which a global minimum is sought. This is the **complex least squares problem**. A **solution vector** $\mathbf{x}_s \in \mathbb{C}^n$ is defined to minimize the cost function

$$\mathbf{x}_s = \underset{\mathbf{x} \in \mathbb{C}^n}{\arg\min} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 . \tag{3.34}$$

Gradient-based methods are often used to solve optimization problems. For a real-valued function of real variables $f : \mathbb{R}^n \to \mathbb{R}$ the gradient points in the direction of greatest change, and extrema are necessarily at critical points with zero gradient

$$\nabla f(\mathbf{x}_s) = \mathbf{0}. \tag{3.35}$$

It is desirable to adapt these ideas to complex functions. Recall for $\mathbb{R}^3$ the gradient is motivated by writing the total derivative of f in terms of a dot product (a kind of inner product)

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy + \frac{\partial f}{\partial z} dz = \left[\hat{\mathbf{x}} \frac{\partial f}{\partial x} + \hat{\mathbf{y}} \frac{\partial f}{\partial y} + \hat{\mathbf{z}} \frac{\partial f}{\partial z} \right] \cdot [\hat{\mathbf{x}} dx + \hat{\mathbf{y}} dy + \hat{\mathbf{z}} dz] = \nabla f \cdot d\mathbf{r}. \tag{3.36}$$

It would seem a **complex gradient** could be defined in terms of complex derivatives and the standard complex inner product, and this is true. However the complex least squares cost function is not complex-differentiable despite being continuous. Additional tools of calculus are needed to generalize the gradient to this case.

3.3.1 CR Calculus

Consider a complex-valued function $f : \mathbb{C} \to \mathbb{C}$ of a complex variable $z \in \mathbb{C}$

$$f(z) \in \mathbb{C}. \tag{3.37}$$

Complex variable z can be represented in rectangular form as a linear combination of two real variables $x, y \in \mathbb{R}$

$$\boxed{z = x + jy}. \tag{3.38}$$

Likewise function f can be represented in rectangular form as a linear combination of two real-valued functions $u, v : \mathbb{R}^2 \to \mathbb{R}$ of two real variables $x, y \in \mathbb{R}$

$$\boxed{f(x, y) = u(x, y) + jv(x, y)} \tag{3.39}$$

so f can be represented as a complex-valued function of two real variables $f : \mathbb{R}^2 \to \mathbb{C}$

$$f(z) = f(x, y) \in \mathbb{C}. \tag{3.40}$$

The **complex derivative** is defined analogous to the real derivative

$$\boxed{\frac{df}{dz} \equiv \lim_{\Delta z \to 0} \frac{f(z + \Delta z) - f(z)}{\Delta z}}. \tag{3.41}$$

Complex differentiability imposes strong constraints on f. For the complex derivative to exist the definition must evaluate to the same number for any direction of Δz in the complex plane. This is similar to the condition that for a real derivative to exist the left and right limits must be equal. Represent Δz in rectangular form as

$$\Delta z = \Delta x + j\Delta y. \tag{3.42}$$

One way to evaluate the complex derivative is along the real axis

$$\frac{df}{dz} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} = \frac{\partial f}{\partial x}. \tag{3.43}$$

Another way to evaluate the complex derivative is along the imaginary axis

$$\frac{df}{dz} = \lim_{\Delta y \to 0} \frac{f(x, y + \Delta y) - f(x, y)}{j\Delta y} = \frac{1}{j}\frac{\partial f}{\partial y}. \tag{3.44}$$

For the complex derivative to exist, Eq. 3.43 and Eq. 3.44 must be equal

$$\frac{\partial f}{\partial x} = \frac{1}{j}\frac{\partial f}{\partial y}. \tag{3.45}$$

Insert the rectangular form of f into this equation

$$\left[\frac{\partial u}{\partial x} + j\frac{\partial v}{\partial x}\right] = \frac{1}{j}\left[\frac{\partial u}{\partial y} + j\frac{\partial v}{\partial y}\right] \tag{3.46}$$

and isolate the real and imaginary parts to form a linear system of equations

$$\boxed{\begin{cases} \frac{\partial u}{\partial x} &= \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} &= -\frac{\partial v}{\partial x} \end{cases}}. \tag{3.47}$$

These are the **Cauchy-Riemann equations**. A function's real and imaginary parts must satisfy these equations to be complex-differentiable, or **holomorphic**.

Some important functions are not holomorphic, such as complex conjugation

$$f(x, y) = x - jy = z^* \tag{3.48}$$

and complex modulus

$$f(x, y) = x^2 + y^2 = [x - jy][x + jy] = z^* z = |z|^2 . \tag{3.49}$$

These functions are used in the complex least squares cost function via the norm. Thus *the complex derivative can't be applied to complex least squares optimization.* In fact the only holomorphic real-valued function is a constant real function. However the rectangular form of a non-holomorphic complex function can still have well-defined real derivatives which can be exploited. This is the key idea of **CR calculus**.

Consider the complex differential dz and its conjugate dz^* in rectangular form

$$\boxed{dz = dx + jdy \qquad dz^* = dx - jdy}. \tag{3.50}$$

Rewrite the complex differentials as the real part and imaginary part differentials

$$dx = \frac{1}{2}[dz + dz^*] \qquad dy = \frac{1}{2j}[dz - dz^*]. \tag{3.51}$$

The total derivative of function f in rectangular form is

$$df = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy. \tag{3.52}$$

Use Eq. 3.51 to express the total derivative in terms of the complex differentials

$$df = \frac{1}{2}\left[\frac{\partial f}{\partial x} + \frac{1}{j}\frac{\partial f}{\partial y}\right]dz + \frac{1}{2}\left[\frac{\partial f}{\partial x} - \frac{1}{j}\frac{\partial f}{\partial y}\right]dz^*. \tag{3.53}$$

Define the **CR (Wirtinger) derivatives** as

$$\boxed{\frac{\partial}{\partial z} \equiv \frac{1}{2}\left[\frac{\partial}{\partial x} + \frac{1}{j}\frac{\partial}{\partial y}\right] \qquad \frac{\partial}{\partial z^*} \equiv \frac{1}{2}\left[\frac{\partial}{\partial x} - \frac{1}{j}\frac{\partial}{\partial y}\right]} \tag{3.54}$$

so the total derivative becomes

$$\boxed{df = \frac{\partial f}{\partial z}dz + \frac{\partial f}{\partial z^*}dz^*}. \tag{3.55}$$

These definitions suggest that $f(z)$ can be parameterized in terms of $f(z, z^*)$

$$f(z) = f(x, y) = f(z, z^*) \in \mathbb{C} \tag{3.56}$$

which is further supported by z and z^* behaving like independent variables with respect to the CR derivatives

$$\frac{\partial z}{\partial z} = \frac{1}{2}\left[\frac{\partial}{\partial x} + \frac{1}{j}\frac{\partial}{\partial y}\right][x + jy] = 1 \qquad \frac{\partial z^*}{\partial z} = \frac{1}{2}\left[\frac{\partial}{\partial x} + \frac{1}{j}\frac{\partial}{\partial y}\right][x - jy] = 0 \tag{3.57}$$

$$\frac{\partial z}{\partial z^*} = \frac{1}{2}\left[\frac{\partial}{\partial x} - \frac{1}{j}\frac{\partial}{\partial y}\right][x + jy] = 0 \qquad \frac{\partial z^*}{\partial z^*} = \frac{1}{2}\left[\frac{\partial}{\partial x} - \frac{1}{j}\frac{\partial}{\partial y}\right][x - jy] = 1. \tag{3.58}$$

In addition, a function $f(z, z^*)$ is holomorphic exactly when $\partial f/\partial z^* = 0$

$$\frac{\partial f}{\partial z^*} = \frac{1}{2}\left[\frac{\partial}{\partial x} - \frac{1}{j}\frac{\partial}{\partial y}\right][u + jv] = \frac{1}{2}\left[\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right] + \frac{j}{2}\left[\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right] = 0. \tag{3.59}$$

The last step follows from the Cauchy-Riemann equations, Eq. 3.47. Therefore a concise statement of the Cauchy-Riemann equations is

$$\boxed{\frac{\partial f}{\partial z^*} = 0 \quad \Leftrightarrow \quad \text{holomorphic}}. \tag{3.60}$$

In the language of CR calculus it is now clear why the least squares cost function is not holomorphic; it depends on conjugated variables via the norm, so $\partial f/\partial z^* \neq 0$.

The CR derivatives behave similar to real derivatives, in that they are linear operators with a product rule and chain rule. Linearity follows from the CR derivative definitions Eq. 3.54 because they are linear combinations of linear operators, thus

$$\boxed{\frac{\partial}{\partial z}[\alpha f(z, z^*) + \beta g(z, z^*)] = \alpha\frac{\partial f}{\partial z} + \beta\frac{\partial g}{\partial z}}. \tag{3.61}$$

The CR product rule follows the same pattern as the real product rule

$$\boxed{\frac{\partial}{\partial z}[f(z, z^*)g(z, z^*)] = \frac{\partial f}{\partial z}g + f\frac{\partial g}{\partial z}} \tag{3.62}$$

and is derived by evaluating in rectangular form

$$\begin{aligned}
\frac{\partial}{\partial z}[f(z, z^*)g(z, z^*)] &= \frac{1}{2}\left[\frac{\partial}{\partial x} + \frac{1}{j}\frac{\partial}{\partial y}\right][f(x, y)g(x, y)] \\
&= \frac{1}{2}\left[\frac{\partial}{\partial x}[fg] + \frac{1}{j}\frac{\partial}{\partial y}[fg]\right] \\
&= \frac{1}{2}\left[\frac{\partial f}{\partial x}g + f\frac{\partial g}{\partial x} + \frac{1}{j}\left[\frac{\partial f}{\partial y}g + f\frac{\partial g}{\partial y}\right]\right] \\
&= \left[\frac{1}{2}\left[\frac{\partial}{\partial x} + \frac{1}{j}\frac{\partial}{\partial y}\right]f\right]g + f\left[\frac{1}{2}\left[\frac{\partial}{\partial x} + \frac{1}{j}\frac{\partial}{\partial y}\right]g\right] \\
&= \frac{\partial f}{\partial z}g + f\frac{\partial g}{\partial z}.
\end{aligned}$$

The CR chain rule is more complicated than the real chain rule, but the pattern follows the derivative of a function of two variables that depend on the same variable

$$\boxed{\frac{\partial}{\partial z}g(f(z,z^*),f^*(z,z^*)) = \frac{\partial g}{\partial f}\frac{\partial f}{\partial z} + \frac{\partial g}{\partial f^*}\frac{\partial f^*}{\partial z}} \tag{3.63}$$

and is also derived by evaluating in rectangular form

$$\begin{aligned}
&= \frac{\partial}{\partial z}g(f(z,z^*),f^*(z,z^*)) \\
&= \frac{1}{2}\left[\frac{\partial}{\partial x} + \frac{1}{j}\frac{\partial}{\partial y}\right] g(u(x,y),v(x,y)) \\
&= \frac{1}{2}\left[\frac{\partial g}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial g}{\partial v}\frac{\partial v}{\partial x} + \frac{1}{j}\frac{\partial g}{\partial u}\frac{\partial u}{\partial y} + \frac{1}{j}\frac{\partial g}{\partial v}\frac{\partial v}{\partial y}\right] \\
&= \frac{1}{4}\left[\frac{\partial g}{\partial u}\frac{\partial u}{\partial x} + j\frac{\partial g}{\partial u}\frac{\partial v}{\partial x} + \frac{1}{j}\frac{\partial g}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial g}{\partial u}\frac{\partial v}{\partial y} + \frac{1}{j}\frac{\partial g}{\partial v}\frac{\partial u}{\partial x} + \frac{\partial g}{\partial v}\frac{\partial v}{\partial x} - \frac{\partial g}{\partial v}\frac{\partial u}{\partial y} + \frac{1}{j}\frac{\partial g}{\partial v}\frac{\partial v}{\partial y}\right] \\
&+ \frac{1}{4}\left[\frac{\partial g}{\partial u}\frac{\partial u}{\partial x} - j\frac{\partial g}{\partial u}\frac{\partial v}{\partial x} + \frac{1}{j}\frac{\partial g}{\partial u}\frac{\partial u}{\partial y} - \frac{\partial g}{\partial u}\frac{\partial v}{\partial y} - \frac{1}{j}\frac{\partial g}{\partial v}\frac{\partial u}{\partial x} + \frac{\partial g}{\partial v}\frac{\partial v}{\partial x} + \frac{\partial g}{\partial v}\frac{\partial u}{\partial y} + \frac{1}{j}\frac{\partial g}{\partial v}\frac{\partial v}{\partial y}\right] \\
&= \frac{1}{4}\left[\frac{\partial g}{\partial u} + \frac{1}{j}\frac{\partial g}{\partial v}\right]\left[\frac{\partial u}{\partial x} + j\frac{\partial v}{\partial x} + \frac{1}{j}\frac{\partial u}{\partial y} + \frac{\partial v}{\partial y}\right] \\
&+ \frac{1}{4}\left[\frac{\partial g}{\partial u} - \frac{1}{j}\frac{\partial g}{\partial v}\right]\left[\frac{\partial u}{\partial x} - j\frac{\partial v}{\partial x} + \frac{1}{j}\frac{\partial u}{\partial y} - \frac{\partial v}{\partial y}\right] \\
&= \left[\frac{1}{2}\left[\frac{\partial g}{\partial u} + \frac{1}{j}\frac{\partial g}{\partial v}\right]\right]\left[\frac{1}{2}\left[\frac{\partial}{\partial x} + \frac{1}{j}\frac{\partial}{\partial y}\right][u+jv]\right] \\
&+ \left[\frac{1}{2}\left[\frac{\partial g}{\partial u} - \frac{1}{j}\frac{\partial g}{\partial v}\right]\right]\left[\frac{1}{2}\left[\frac{\partial}{\partial x} + \frac{1}{j}\frac{\partial}{\partial y}\right][u-jv]\right] \\
&= \frac{\partial g}{\partial f}\frac{\partial f}{\partial z} + \frac{\partial g}{\partial f^*}\frac{\partial f^*}{\partial z}.
\end{aligned}$$

In addition, conjugation rules follow directly from the definitions Eq. 3.54

$$\boxed{\left[\frac{\partial f}{\partial z}\right]^* = \frac{\partial f^*}{\partial z^*} \qquad \left[\frac{\partial f}{\partial z^*}\right]^* = \frac{\partial f^*}{\partial z}}. \tag{3.64}$$

Extending to multivariate functions is straight forward. Consider a complex-valued vector function $\mathbf{f}:\mathbb{C}^n \to \mathbb{C}^m$ of a complex-valued vector $\mathbf{z} \in \mathbb{C}^n$

$$\mathbf{f}(\mathbf{z}) = \mathbf{f}(\mathbf{z},\mathbf{z}^*) = \mathbf{f}(\mathbf{x},\mathbf{y}) \in \mathbb{C}^m. \tag{3.65}$$

The total derivative written in matrix notation is

$$\boxed{d\mathbf{f} = \frac{\partial \mathbf{f}}{\partial \mathbf{z}} d\mathbf{z} + \frac{\partial \mathbf{f}}{\partial \mathbf{z}^*} d\mathbf{z}^*} \tag{3.66}$$

where the differential vectors

$$\boxed{d\mathbf{z} \equiv \begin{bmatrix} dz_1 \\ \vdots \\ dz_n \end{bmatrix} \qquad d\mathbf{z}^* \equiv \begin{bmatrix} dz_1^* \\ \vdots \\ dz_n^* \end{bmatrix}} \tag{3.67}$$

are transformed by the matrices

$$\boxed{\frac{\partial \mathbf{f}}{\partial \mathbf{z}} \equiv \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \cdots & \frac{\partial f_1}{\partial z_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial z_1} & \cdots & \frac{\partial f_m}{\partial z_n} \end{bmatrix} \qquad \frac{\partial \mathbf{f}}{\partial \mathbf{z}^*} \equiv \begin{bmatrix} \frac{\partial f_1}{\partial z_1^*} & \cdots & \frac{\partial f_1}{\partial z_n^*} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial z_1^*} & \cdots & \frac{\partial f_m}{\partial z_n^*} \end{bmatrix}}. \tag{3.68}$$

This is simply a system of equations describing the total derivative of each component of $\mathbf{f}$ with respect to complex differential vectors $d\mathbf{z}$ and $d\mathbf{z}^*$. Conjugation generalizes in the expected way

$$\boxed{\left[\frac{\partial \mathbf{f}}{\partial \mathbf{z}}\right]^* = \frac{\partial \mathbf{f}^*}{\partial \mathbf{z}^*} \qquad \left[\frac{\partial \mathbf{f}}{\partial \mathbf{z}^*}\right]^* = \frac{\partial \mathbf{f}^*}{\partial \mathbf{z}}}. \tag{3.69}$$

The chain rule also generalizes in the expected way

$$\boxed{\frac{\partial}{\partial \mathbf{z}} \mathbf{g}(\mathbf{f}(\mathbf{z})) = \frac{\partial \mathbf{g}}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial \mathbf{z}} + \frac{\partial \mathbf{g}}{\partial \mathbf{f}^*} \frac{\partial \mathbf{f}^*}{\partial \mathbf{z}} \qquad \frac{\partial}{\partial \mathbf{z}^*} \mathbf{g}(\mathbf{f}(\mathbf{z})) = \frac{\partial \mathbf{g}}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial \mathbf{z}^*} + \frac{\partial \mathbf{g}}{\partial \mathbf{f}^*} \frac{\partial \mathbf{f}^*}{\partial \mathbf{z}^*}}. \tag{3.70}$$

There is no higher-dimensional analog of the product rule.

Now consider the special case of the total derivative of a real-valued function of complex variables $f : \mathbb{C}^n \to \mathbb{R}$. Because f is real, the total derivative can be written in a special form

$$df = \frac{\partial f}{\partial \mathbf{z}} d\mathbf{z} + \frac{\partial f}{\partial \mathbf{z}^*} d\mathbf{z}^* = \frac{\partial f}{\partial \mathbf{z}} d\mathbf{z} + \left[\frac{\partial f}{\partial \mathbf{z}} d\mathbf{z}\right]^* = 2\mathfrak{Re}\left(\frac{\partial f}{\partial \mathbf{z}} d\mathbf{z}\right). \tag{3.71}$$

As with Eq. 3.36, rewrite Eq. 3.71 using a (standard complex) inner product

$$\frac{\partial f}{\partial \mathbf{z}} d\mathbf{z} = \left[\left[\frac{\partial f}{\partial \mathbf{z}} \right]^H \right]^H d\mathbf{z} = \nabla f^H d\mathbf{z} = \langle \nabla f, d\mathbf{z} \rangle \tag{3.72}$$

such that

$$\boxed{df = 2\mathfrak{Re}(\langle \nabla f, d\mathbf{z} \rangle)} \tag{3.73}$$

where the **CR gradient** is defined as

$$\boxed{\nabla f(\mathbf{z}) \equiv \left[\frac{\partial f}{\partial \mathbf{z}} \right]^H}. \tag{3.74}$$

The factor of 2 from Eq. 3.73 is not included in the definition of the CR gradient to make it compatible with the complex gradient. Annoyingly this makes the CR gradient 1/2 the real gradient when applied to real-valued functions of real variables, so there isn't perfect symmetry between real and complex gradient theories. However both theories will agree on df.

To verify that the CR gradient shares qualities with the real gradient consider the Schwarz inequality applied to the CR gradient of function f and unit vector $\hat{\mathbf{n}} \in \mathbb{C}^n$

$$|\langle \nabla f, \hat{\mathbf{n}} \rangle| \leq \|\nabla f\| \, \|\hat{\mathbf{n}}\| = \|\nabla f\| \,. \tag{3.75}$$

The inequality is an equality either when $\nabla f = \mathbf{0}$, or when the vectors are colinear such that $\nabla f = \alpha \hat{\mathbf{n}}$ for a scalar $\alpha \in \mathbb{C} - \{0\}$. To maximize df in Eq. 3.73 the real part of $\langle \nabla f, \hat{\mathbf{n}} \rangle$ must be maximized, which happens when $\hat{\mathbf{n}} = \nabla f / \|\nabla f\|$. In other words, the CR gradient points in the direction of maximal change like the real gradient, so the gradient terminology is warranted. It follows then that extrema for function f are necessarily at critical points where the CR gradient is zero

$$\nabla f(\mathbf{z}_s) = \mathbf{0}. \tag{3.76}$$

These facts taken together solidify the interpretation of the CR gradient as a gradient.

In addition to basic CR calculus, an identity for the CR derivative of the modulus of a holomorphic function $f : \mathbb{C}^n \to \mathbb{C}$ will be theoretically useful

$$\frac{\partial}{\partial z} |f(z; z^*)|^2 = \frac{\partial}{\partial z}[f^* f] = \frac{\partial f^*}{\partial z} f + f^* \frac{\partial f}{\partial z} = \left[\frac{\partial f}{\partial z^*} f^*\right]^* + f^* \frac{\partial f}{\partial z} = f^* \frac{\partial f}{\partial z} \tag{3.77}$$

where the semicolon in $f(z; z^*)$ indicates the function is technically parameterized in terms of z and z^* but is independent of z^*.

3.3.2 Solution

Recall the complex least squares cost function to be minimized is the residual norm

$$f(\mathbf{x}) = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 . \tag{3.78}$$

Take the CR gradient of the cost function

$$\nabla f(\mathbf{x}) = \nabla \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 \tag{3.79}$$

convert to summation notation to evaluate

$$\begin{aligned}
\left[\frac{\partial f}{\partial x_k}\right]^* &= \left[\frac{\partial}{\partial x_k}\sum_{j=1}^{m}\left|b_j - \sum_{i=1}^{n} A_{ji}x_i\right|^2\right]^* \\
&= \left[\sum_{j=1}^{m}\frac{\partial}{\partial x_k}\left|b_j - \sum_{i=1}^{n} A_{ji}x_i\right|^2\right]^* \\
&= \left[\sum_{j=1}^{m}\left[b_j - \sum_{i=1}^{n} A_{ji}x_i\right]^*\frac{\partial}{\partial x_k}\left[b_j - \sum_{i=1}^{n} A_{ji}x_i\right]\right]^* \quad \text{(Eq. 3.77)} \\
&= \left[\sum_{j=1}^{m}\left[b_j - \sum_{i=1}^{n} A_{ji}x_i\right]^*\left[\frac{\partial b_j}{\partial x_k} - \sum_{i=1}^{n} A_{ji}\frac{\partial x_i}{\partial x_k}\right]\right]^* \\
&= \left[\sum_{j=1}^{m}\left[b_j - \sum_{i=1}^{n} A_{ji}x_i\right]^*\left[0 - \sum_{i=1}^{n} A_{ji}\delta_{ik}\right]\right]^* \\
&= \left[\sum_{j=1}^{m}\left[b_j - \sum_{i=1}^{n} A_{ji}x_i\right]^*\left[-A_{jk}\right]\right]^* \\
&= -\sum_{j=1}^{m} {A_{jk}}^*\left[b_j - \sum_{i=1}^{n} A_{ji}x_i\right]
\end{aligned} \tag{3.80}$$

and convert back to matrix notation to be concise

$$\nabla f(\mathbf{x}) = -\mathbf{A}^H\left[\mathbf{b} - \mathbf{A}\mathbf{x}\right]. \tag{3.81}$$

Optimize the cost function f by setting its CR gradient to zero

$$\nabla f(\mathbf{x}_s) = -\mathbf{A}^H[\mathbf{b} - \mathbf{A}\mathbf{x}_s] = \mathbf{0} \tag{3.82}$$

to derive the **complex least squares equation**

$$\boxed{\mathbf{A}^H\mathbf{A}\mathbf{x}_s = \mathbf{A}^H\mathbf{b}}. \tag{3.83}$$

This is equivalent to the modified linear system

$$\mathbf{A}'\mathbf{x}_s = \mathbf{b}' \tag{3.84}$$

where $\mathbf{A}' = \mathbf{A}^H\mathbf{A}$ and $\mathbf{b}' = \mathbf{A}^H\mathbf{b}$. These are also known as normal equations. Geometrically a complex least squares solution vector $\mathbf{x}_s$ makes the solution residual vector $\mathbf{r}_s$ orthogonal to the columnspace (output space) of $\mathbf{A}$

$$\mathbf{A}^H[\mathbf{b} - \mathbf{A}\mathbf{x}_s] = \mathbf{A}^H\mathbf{r}_s = \mathbf{0} \tag{3.85}$$

meaning the solution residual only has components in the space inaccessible to $\mathbf{A}$.

Of passing interest is the real least squares cost function $f : \mathbb{R}^n \to \mathbb{R}$

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 \tag{3.86}$$

where a factor of $1/2$ is introduced to compensate for differences between the real and CR gradients. The real gradient is computed in a similar way as Eq. 3.80 except instead of Eq. 3.77 the real chain rule is used to differentiate the square and cancel the $1/2$, resulting in

$$\nabla f(\mathbf{x}) = -\mathbf{A}^T[\mathbf{b} - \mathbf{A}\mathbf{x}]. \tag{3.87}$$

Optimize function f by setting the gradient to zero

$$\nabla f(\mathbf{x}_s) = -\mathbf{A}^T[\mathbf{b} - \mathbf{A}\mathbf{x}_s] = \mathbf{0} \tag{3.88}$$

to form the real least squares equation

$$\mathbf{A}^T\mathbf{A}\mathbf{x}_s = \mathbf{A}^T\mathbf{b}. \tag{3.89}$$

The only difference between Eq. 3.83 and Eq. 3.89 is that the conjugate transposes are replaced by transposes. The similarity between these results belies the important theoretical differences in their derivation.

3.3.3 Regularization

A problem with the complex least squares cost function Eq. 3.32 is that infinite solution vectors exist if $\mathbf{A}$ has a non-trivial nullspace. To see this, evaluate the cost function f at a solution vector $\mathbf{x}_s$ displaced by any non-zero nullspace vector $\mathbf{e} \in \text{nullspace}(\mathbf{A}) - \{\mathbf{0}\} \subset \mathbb{C}^n$ such that

$$f(\mathbf{x}_s + \mathbf{e}) = \|\mathbf{b} - \mathbf{A}[\mathbf{x}_s + \mathbf{e}]\|^2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}_s - \mathbf{0}\|^2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}_s\|^2 = f(\mathbf{x}_s). \quad (3.90)$$

Thus any non-zero nullspace vector can be added to a solution vector to yield a different solution vector with the same minimal cost function value.

Some additional condition must be imposed to guarantee a unique solution. This process is called **regularization**. In general the cost function is modified by adding a regularization function $R : \mathbb{C}^n \to \mathbb{R}$ that depends on $\mathbf{x}$

$$f(\mathbf{x}) = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 + R(\mathbf{x}). \quad (3.91)$$

The regularization function R is somewhat arbitrary and many choices exist. Since components of non-trivial nullspaces can be arbitrarily large, it is reasonable to penalize the norm of solution vectors to prevent nullspaces from superfluously contributing to solution vectors. As with regularization functions, several norms exist.

An obvious choice is the norm induced by the standard complex inner product

$$\|\mathbf{x}\|_2 = \langle \mathbf{x}, \mathbf{x} \rangle^{\frac{1}{2}} = [\mathbf{x}^H \mathbf{x}]^{\frac{1}{2}}. \quad (3.92)$$

This is the L^2 norm, and a subscript 2 is used here to distinguish it from other norms. Setting R to the L^2 norm squared results in **$\mathbf{L^2}$ (Tikhonov) regularization**

$$R(\mathbf{x}) = \|\mathbf{x}\|_2^2 . \quad (3.93)$$

Prior information about the solution can be incorporated into the L^2 regularizer by penalizing the difference between an expected solution vector $\mathbf{x}_p \in \mathbb{C}^n$ and vector $\mathbf{x}$

$$R(\mathbf{x}) = \|\mathbf{x}_p - \mathbf{x}\|_2^2 . \tag{3.94}$$

Weighting can also be incorporated into the L^2 regularizer by multiplying the vectors in the norm by invertible matrix $\mathbf{\Gamma} \in \mathbb{C}^{n \times n}$

$$R(\mathbf{x}) = \|\mathbf{\Gamma}\left[\mathbf{x}_p - \mathbf{x}\right]\|_2^2 . \tag{3.95}$$

A common weighting choice is $\mathbf{\Gamma} = \gamma^{\frac{1}{2}}\mathbf{I}$ where $\gamma \in \mathbb{R}^+$ such that

$$R(\mathbf{x}) = \gamma \|\mathbf{x}_p - \mathbf{x}\|_2^2 \tag{3.96}$$

which provides a way to adjust the relative importance of the residual norm versus the regularization function in the cost function. Unfortunately choosing the optimal γ is problem-specific and usually done empirically.

Other choices of regularization function exist. The generalization of the L^2 norm is the L^p norm, which changes the shape of the unit sphere

$$\|\mathbf{x}\|_p \equiv \left[\sum_{i=1}^{n} |x_i|^p\right]^{\frac{1}{p}} \quad \text{where} \quad p \in [1, \infty] \subset \mathbb{R}. \tag{3.97}$$

Inserting $p = 0$ into Eq. 3.97 results in the L^0 "norm" which counts the number of non-zero elements in a vector. L^0 is not a true norm (e.g. $\|\alpha\mathbf{x}\|_0 \neq |\alpha| \|\mathbf{x}\|_0$), but it is an attractive regularizer for enforcing sparsity. Unfortunately L^0 minimization is NP-hard. While it is difficult to use L^0 regularization directly, Ch. 5 will discuss how to indirectly implement an L^0 prior using depth cameras to reduce the size of the inverse problem considerably. More exotic regularization methods include gradient-based and total-variation-based schemes, but these are computationally prohibitive for local real-time applications like security screening.

3.3.4 Left Preconditioning

To **precondition** a linear system means to transform it into a system with a better condition number before solving it. While this doesn't change the solution, it may make numeric computation more accurate and iterative algorithms converge faster. Left preconditioning can be incorporated into complex least squares by left multiplying the original linear system Eq. 3.30 by invertible matrix $\mathbf{P} \in \mathbb{C}^{m \times m}$

$$\mathbf{PAx} = \mathbf{Pb} \tag{3.98}$$

which results in a cost function similar to the weighted L^2 regularizer with prior

$$f(\mathbf{x}) = \|\mathbf{P}\left[\mathbf{b} - \mathbf{Ax}\right]\|^2 . \tag{3.99}$$

In the presence of measurement noise, the covariance matrix

$$\mathbf{K_{xy}} \equiv [\mathrm{cov}(x_i, y_j)] \tag{3.100}$$

can be used to weight measurement channels based on statistical considerations. Specifically, the inverse of the auto-covariance matrix, the precision matrix, of data vector $\mathbf{b}$ can be chosen as a preconditioner

$$\mathbf{P} = \mathbf{K_{bb}^{-1}}. \tag{3.101}$$

Preconditioning is best incorporated directly into iterative algorithms to take advantage of clever simplifications, as will be shown, but is included here for generality.

3.3.5 Solution Revisited

Regularization and preconditioning are now applied to formulate a well-posed least squares problem with a unique solution. The original complex least squares cost

function Eq. 3.32 is the residual norm squared

$$f(\mathbf{x}) = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2. \tag{3.102}$$

This cost function is modified by incorporating left preconditioning and weighted L^2 regularization with a prior

$$f(\mathbf{x}) = \|\mathbf{P}\left[\mathbf{b} - \mathbf{A}\mathbf{x}\right]\|^2 + \|\mathbf{\Gamma}\left[\mathbf{x}_p - \mathbf{x}\right]\|^2. \tag{3.103}$$

Optimize f by taking the CR gradient using Eq. 3.81 as a template and set to zero

$$\nabla f(\mathbf{x}_s) = -\mathbf{A}^H\mathbf{P}^H\mathbf{P}\left[\mathbf{b} - \mathbf{A}\mathbf{x}_s\right] - \mathbf{\Gamma}^H\mathbf{\Gamma}\left[\mathbf{x}_p - \mathbf{x}_s\right] = \mathbf{0} \tag{3.104}$$

to derive the regularized complex least squares equation

$$\boxed{\left[\mathbf{A}^H\mathbf{P}^H\mathbf{P}\mathbf{A} + \mathbf{\Gamma}^H\mathbf{\Gamma}\right]\mathbf{x}_s = \left[\mathbf{A}^H\mathbf{P}^H\mathbf{P}\mathbf{b} + \mathbf{\Gamma}^H\mathbf{\Gamma}\mathbf{x}_p\right].} \tag{3.105}$$

While this looks complicated, this is equivalent to the modified linear system

$$\mathbf{A}'\mathbf{x}_s = \mathbf{b}' \tag{3.106}$$

where

$$\mathbf{A}' = \mathbf{A}^H\mathbf{P}^H\mathbf{P}\mathbf{A} + \mathbf{\Gamma}^H\mathbf{\Gamma} \tag{3.107}$$

and

$$\mathbf{b}' = \mathbf{A}^H\mathbf{P}^H\mathbf{P}\mathbf{b} + \mathbf{\Gamma}^H\mathbf{\Gamma}\mathbf{x}_p. \tag{3.108}$$

Matrix $\mathbf{A}'$ is conjugate-symmetric

$$\mathbf{A}'^H = [\mathbf{A}^H\mathbf{P}^H\mathbf{P}\mathbf{A} + \mathbf{\Gamma}^H\mathbf{\Gamma}]^H = \mathbf{A}^H\mathbf{P}^H\mathbf{P}\mathbf{A} + \mathbf{\Gamma}^H\mathbf{\Gamma} = \mathbf{A}' \tag{3.109}$$

and positive-definite

$$\begin{aligned}\langle \mathbf{x}, \mathbf{A}'\mathbf{x} \rangle &= \left\langle \mathbf{x}, [\mathbf{A}^H\mathbf{P}^H\mathbf{P}\mathbf{A} + \boldsymbol{\Gamma}^H\boldsymbol{\Gamma}]\mathbf{x} \right\rangle \\ &= \left\langle \mathbf{x}, \mathbf{A}^H\mathbf{P}^H\mathbf{P}\mathbf{A}\mathbf{x} \right\rangle + \left\langle \mathbf{x}, \boldsymbol{\Gamma}^H\boldsymbol{\Gamma}\mathbf{x} \right\rangle \\ &= \langle \mathbf{P}\mathbf{A}\mathbf{x}, \mathbf{P}\mathbf{A}\mathbf{x} \rangle + \langle \boldsymbol{\Gamma}\mathbf{x}, \boldsymbol{\Gamma}\mathbf{x} \rangle \\ &= \|\mathbf{P}\mathbf{A}\mathbf{x}\|^2 + \|\boldsymbol{\Gamma}\mathbf{x}\|^2 \\ &\geq \|\boldsymbol{\Gamma}\mathbf{x}\|^2 \\ &> 0 \quad \text{when} \quad \mathbf{x} \neq \mathbf{0}\end{aligned} \tag{3.110}$$

and thus invertible

$$\mathbf{x}_s = \mathbf{A}'^{-1}\mathbf{b} \tag{3.111}$$

because a square matrix with a trivial nullspace is full rank by the rank-nullity theorem.

For the special case of $\mathbf{P} = \mathbf{I}$, $\boldsymbol{\Gamma} = \gamma^{\frac{1}{2}}\mathbf{I}$, and $\mathbf{x}_p = \mathbf{0}$, the modified complex least squares equation reduces to the form used for the remainder of this text

$$\boxed{\left[\mathbf{A}^H\mathbf{A} + \gamma\mathbf{I}\right]\mathbf{x}_s = \mathbf{A}^H\mathbf{b}}. \tag{3.112}$$

This is a well-posed inverse problem with a unique solution!

3.4 Iterative Methods

The previous section demonstrated how to cast an ill-posed inverse problem into a well-posed inverse problem using complex least squares with regularization. However no attempt was made to solve the modified problem. Gauss-Jordon elimination could be used in principle, but has a time complexity of $O_t(n^3)$, which is computationally prohibitive for very large sensing matrices. Iterative solvers become the only practical

Table 3.1: Iterative Linear Solver Symbols

Symbol	Name
f	cost function
q	quadratic form, special case of f
$\mathbf{x}$, $\mathbf{v}$, $\mathbf{w}$	arbitrary vectors
$\mathbf{x}_s$	solution vector
$\mathbf{x}_i$	iterate vector
$\mathbf{x}_0$	initial guess vector
$\mathbf{e}_i$	error vector
$\mathbf{d}_i$	search direction vector
$\mathbf{D}_i$	search direction vector augmented matrix
α_i	search scale
$\mathbf{A}$	system matrix
$\mathbf{b}$	data vector
$\mathbf{r}_i$	residual vector

way to solve very large linear systems. Three iterative solvers are developed in this section; gradient descent on quadratic forms, conjugate gradient, and GMRES.

A large number of symbols are defined in this section. A summary is found in Tab. 3.1.

3.4.1 Iterative Optimization

Consider the general problem of optimizing a continuous real-valued function of complex variables $f : \mathbb{C}^n \to \mathbb{R}$. A solution vector $\mathbf{x}_s \in \mathbb{C}^n$ at an extreme value of f will

necessarily be a critical point with zero CR gradient

$$\nabla f(\mathbf{x}_s) = \mathbf{0}. \tag{3.113}$$

In general, iterative optimization generates a sequence of approximate solutions called **iterate vectors** $\mathbf{x}_i \in \mathbb{C}^n$, starting with an **initial guess vector** $\mathbf{x}_0 \in \mathbb{C}^n$. Each iteration the iterate vector $\mathbf{x}_i$ is updated by adding a **search direction vector** $\mathbf{d}_i \in \mathbb{C}^n$ multiplied by a **search scale** $\alpha_i \in \mathbb{C}$

$$\boxed{\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{d}_i \alpha_i}. \tag{3.114}$$

Algorithms have different strategies for choosing $\mathbf{d}_i$ and α_i, which affects convergence. The recurrence relation can be fully expand from the initial guess as the summation

$$\mathbf{x}_i = \mathbf{x}_0 + \sum_{j=1}^{i-1} \mathbf{d}_j \alpha_j. \tag{3.115}$$

A statement can already be made about the nature of iterative optimization without supplying more details. Suppose on the i-th iteration a search direction vector $\mathbf{d}_i$ is chosen by an unspecified algorithm. A general condition between $\mathbf{d}_i$ and the gradient at the next iterate $\mathbf{x}_{i+1}$ is found by optimizing f with respect to α_i

$$\frac{\partial}{\partial \alpha_i} f(\mathbf{x}_{i+1}) = \frac{\partial f(\mathbf{x}_{i+1})}{\partial \mathbf{x}_{i+1}} \frac{\partial \mathbf{x}_{i+1}}{\partial \alpha_i} + \frac{\partial f(\mathbf{x}_{i+1})}{\partial \mathbf{x}_{i+1}{}^*} \frac{\partial \mathbf{x}_{i+1}{}^*}{\partial \alpha_i} = \boxed{\nabla f(\mathbf{x}_{i+1})^H \mathbf{d}_i = 0} \tag{3.116}$$

where the CR chain rule Eq. 3.70 has been applied and the second term eliminated because $\partial/\partial\alpha_i[\mathbf{x}_i^* + \mathbf{d}_i^* \alpha_i{}^*] = \mathbf{0}$. Geometrically this equation says follow a search direction vector until the gradient is orthogonal when updating an iterate. At such a point f is not changing along the search direction vector, and so a new search direction should be chosen to further optimize f.

Iterate vectors are approximate solutions, so an **error vector** $\mathbf{e}_i \in \mathbb{C}^n$ can be defined as

$$\boxed{\mathbf{e}_i \equiv \mathbf{x}_i - \mathbf{x}_s} \tag{3.117}$$

which is like the residual but for the domain of f. Error is typically unknown because it is defined in terms of a solution vector $\mathbf{x}_s$. Despite this limitation the concept is theoretically useful. The definition for the error vector can be used to rewrite the iterate recurrence relation Eq. 3.114 as an error recurrence relation

$$\mathbf{e}_{i+1} = \mathbf{e}_i + \mathbf{d}_i \alpha_i \tag{3.118}$$

and the iterate summation Eq. 3.115 as an error summation

$$\mathbf{e}_i = \mathbf{e}_0 + \sum_{j=1}^{i-1} \mathbf{d}_j \alpha_j. \tag{3.119}$$

3.4.2 Quadratic Form

Application of complex least squares with L^2 regularization results in a linear system with a conjugate-symmetric positive-definite matrix. These properties can be used to design efficient gradient-based iterative minimization algorithms by considering the complex least squares cost function for the modified problem. Therefore consider the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.120}$$

where the matrix $\mathbf{A} \in \mathbb{C}^{n \times n}$ is conjugate-symmetric

$$\mathbf{A} = \mathbf{A}^H \tag{3.121}$$

and positive-definite

$$\langle \mathbf{x}, \mathbf{A}\mathbf{x} \rangle \geq 0 \quad \text{and} \quad \langle \mathbf{x}, \mathbf{A}\mathbf{x} \rangle = 0 \Leftrightarrow \mathbf{x} = \mathbf{0} \tag{3.122}$$

and thus invertible. The complex least squares cost function for invertible systems is

$$\begin{aligned} f(\mathbf{x}) &= \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 \\ &= \langle \mathbf{b} - \mathbf{A}\mathbf{x}, \mathbf{b} - \mathbf{A}\mathbf{x} \rangle \\ &= \langle \mathbf{b}, \mathbf{b} \rangle - \langle \mathbf{b}, \mathbf{A}\mathbf{x} \rangle - \langle \mathbf{A}\mathbf{x}, \mathbf{b} \rangle + \langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{x} \rangle \\ &= \|\mathbf{b}\|^2 - 2\mathfrak{Re}(\mathbf{b}^H \mathbf{A}\mathbf{x}) + \mathbf{x}^H \mathbf{A}^H \mathbf{A}\mathbf{x} \\ &= c' - 2\mathfrak{Re}(\mathbf{b}'^H \mathbf{x}) + \mathbf{x}^H \mathbf{A}'\mathbf{x} \end{aligned} \tag{3.123}$$

where

$$c' = \|\mathbf{b}\|^2, \quad \mathbf{b}' = \mathbf{A}^H \mathbf{b}, \quad \mathbf{A}' = \mathbf{A}^H \mathbf{A}. \tag{3.124}$$

Note the cost function does not need regularization because matrix $\mathbf{A}$ is invertible. Matrix $\mathbf{A}'$ is both conjugate-symmetric

$$\mathbf{A}'^H = [\mathbf{A}^H \mathbf{A}]^H = \mathbf{A}^H \mathbf{A} = \mathbf{A}' \tag{3.125}$$

and positive-definite

$$\langle \mathbf{x}, \mathbf{A}'\mathbf{x} \rangle = \langle \mathbf{x}, \mathbf{A}^H \mathbf{A}\mathbf{x} \rangle = \langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{x} \rangle = \|\mathbf{A}\mathbf{x}\|^2 \geq 0. \tag{3.126}$$

There is an opportunity to be clever when $\mathbf{A}$ shares these properties. Define the expanded form of cost function f as the quadratic form $q : \mathbb{C}^n \to \mathbb{R}$

$$q(\mathbf{x}) \equiv \mathbf{x}^T \mathbf{A}\mathbf{x} - 2\mathfrak{Re}(\mathbf{b}^H \mathbf{x}) + c \tag{3.127}$$

where the primes have been suppressed for reasons that will become apparent in a moment. This is the most general way to write a real-valued quadratic equation of complex variables. Take the CR gradient of q

$$\nabla q(\mathbf{x}) = \nabla \left[\mathbf{x}^T \mathbf{A}\mathbf{x} - 2\mathfrak{Re}(\mathbf{b}^H \mathbf{x}) + c \right] \tag{3.128}$$

convert to summation notation to evaluate

$$\begin{aligned}
\left[\frac{\partial q}{\partial x_k}\right]^* &= \left[\frac{\partial}{\partial x_k}\left[\sum_j x_j^* \sum_i A_{ji}x_i - \sum_j b_j^* x_j - \sum_j x_j^* b_j + c\right]\right]^* \\
&= \left[\sum_j \sum_i A_{ji}\frac{\partial}{\partial x_k}\left[x_j^* x_i\right] - \sum_j b_j^* \frac{\partial x_j}{\partial x_k} - \sum_j \frac{\partial x_j^*}{\partial x_k} b_j + \frac{\partial c}{\partial x_k}\right]^* \\
&= \left[\sum_j \sum_i A_{ji}\left[\frac{\partial x_j^*}{\partial x_k}x_i + x_j^*\frac{\partial x_i}{\partial x_k}\right] - \sum_j b_j^*\delta_{jk} - \sum_j 0 b_j + 0\right]^* \\
&= \left[\sum_j \sum_i A_{ji}\left[0\, x_i + x_j^*\delta_{ik}\right] - b_k^*\right]^* \\
&= \left[\sum_j \left[\sum_i A_{ji}\delta_{ik}\right] x_j^* - b_k^*\right]^* \\
&= \left[\sum_j A_{jk}x_j^* - b_k^*\right]^* \\
&= \sum_j A_{jk}^* x_j - b_k \\
&= \sum_j A_{kj} x_j - b_k \quad \text{(conjugate-symmetric)}
\end{aligned} \tag{3.129}$$

and convert back again to matrix notation to be concise

$$\nabla q(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}. \tag{3.130}$$

Optimize q by setting the gradient to zero

$$\nabla q(\mathbf{x}_s) = \mathbf{A}\mathbf{x}_s - \mathbf{b} = \mathbf{0} \tag{3.131}$$

to derive the *original system to be solved*

$$\mathbf{A}\mathbf{x}_s = \mathbf{b}. \tag{3.132}$$

This is a remarkable result. Evidently the gradient of the quadratic form is the (negative) residual of the linear system to be solved, and when q is optimized the

system is solved. Since every conjugate-symmetric positive-definite matrix has a unique Cholesky decomposition $\mathbf{A} = \mathbf{L}\mathbf{L}^H$, where $\mathbf{L} \in \mathbb{C}^{n\times n}$ is a lower-triangular conjugate-symmetric positive-definite matrix, the complex least squares problem can be replaced with the equivalent problem of minimizing the quadratic form using matrix $\mathbf{A}$, data vector $\mathbf{b}$, and constant $c \in \mathbb{R}$. It is now shown q has a unique global minimum at $\mathbf{x}_s$. Consider evaluating q at some solution vector $\mathbf{x}_s$ offset by non-zero error vector $\mathbf{e} \in \mathbb{C}^n - \{0\}$

$$\begin{aligned}
&= q(\mathbf{x}_s + \mathbf{e}) \\
&= [\mathbf{x}_s + \mathbf{e}]^H \mathbf{A}[\mathbf{x}_s + \mathbf{e}] - 2\mathfrak{Re}(\mathbf{b}^H[\mathbf{x}_s + \mathbf{e}]) + c \\
&= [\mathbf{x}_s^H \mathbf{A}\mathbf{x}_s - 2\mathfrak{Re}(\mathbf{b}^H\mathbf{x}_s) + c] + \mathbf{e}^H\mathbf{A}\mathbf{e} + [\mathbf{x}_s^H\mathbf{A}\mathbf{e} + \mathbf{e}^H\mathbf{A}\mathbf{x}_s] - 2\mathfrak{Re}(\mathbf{b}^H\mathbf{e}) \\
&= q(\mathbf{x}_s) + \mathbf{e}^H\mathbf{A}\mathbf{e} + 2\mathfrak{Re}(\mathbf{x}_s^H\mathbf{A}\mathbf{e}) - 2\mathfrak{Re}(\mathbf{b}^H\mathbf{e}) \\
&= q(\mathbf{x}_s) + \mathbf{e}^H\mathbf{A}\mathbf{e} \\
&> q(\mathbf{x}_s) \quad \text{for} \quad \mathbf{e} \neq \mathbf{0} \quad \text{(positive-definite)}
\end{aligned} \tag{3.133}$$

therefore $q(\mathbf{x}_s)$ is a global minimum.

Due to the close connection between the gradient of a quadratic form and the residual of the associated linear system, it is helpful to use the definition of residual Eq. 3.31 to rewrite the iterate recurrence relation Eq. 3.114 as a residual recurrence relation

$$\boxed{\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{A}\mathbf{d}_i\alpha_i} \tag{3.134}$$

and the iterate summation Eq. 3.115 as a residual summation

$$\mathbf{r}_i = \mathbf{r}_0 - \sum_{j=1}^{i-1} \mathbf{A}\mathbf{d}_j\alpha_j. \tag{3.135}$$

Summarizing the connections between various quantities, it is found

$$\boxed{\mathbf{r}_i = \mathbf{b} - \mathbf{A}\mathbf{x}_i = -\mathbf{A}\mathbf{e}_i = -\nabla q(\mathbf{x}_i)}. \tag{3.136}$$

With an analytic gradient for q, iterative optimization condition Eq. 3.116 can be evaluated

$$\frac{\partial}{\partial \alpha_i} q(\mathbf{x}_{i+1}) = \nabla q(\mathbf{x}_{i+1})^H \mathbf{d}_i = -\mathbf{r}_{i+1}^H \mathbf{d}_i = -[\mathbf{r}_i - \mathbf{A}\mathbf{d}_i \alpha_i]^H \mathbf{d}_i = 0 \tag{3.137}$$

and the optimal search scale α_i is found to be

$$\boxed{\alpha_i = \frac{\mathbf{d}_i^H \mathbf{r}_i}{\mathbf{d}_i^H \mathbf{A}\mathbf{d}_i}}. \tag{3.138}$$

The only unaddressed detail for iterative optimization of a quadratic form is the choice of search direction $\mathbf{d}_i$. Two algorithms will be discussed; gradient descent, and the superior conjugate gradient method.

3.4.3 Gradient Descent

Gradient descent is a general and intuitive iterative minimization algorithm. The gradient of a function points in the direction of greatest increase, so choose as the i-th search direction vector $\mathbf{d}_i$ the negative gradient at the current iterate $\mathbf{x}_i$

$$\boxed{\mathbf{d}_i = -\nabla f(\mathbf{x}_i)}. \tag{3.139}$$

In the case of a quadratic form, the gradient is available as an easy to compute residual. Adding to the list of equivalences, it is found

$$\mathbf{d}_i = -\nabla q(\mathbf{x}_i) = \mathbf{r}_i = \mathbf{b} - \mathbf{A}\mathbf{x}_i = -\mathbf{A}\mathbf{e}_i. \tag{3.140}$$

Gradient descent on a quadratic form is outlined in Algorithm 1. Given are a conjugate-symmetric positive-definite matrix $\mathbf{A} \in \mathbb{C}^{n \times n}$, data vector $\mathbf{b} \in \mathbb{C}^n$, initial guess vector $\mathbf{x}_0 \in \mathbb{C}^n$, and tolerance $\epsilon \in \mathbb{R}^+$. First compute the the initial residual $\mathbf{r}_0$. Then iterate until a maximum number of iterations $i_{max} \in \mathbb{Z}^+$ is reached, or the

residual norm is less than or equal to the tolerance. Each iteration the search scale is computed with Eq. 3.138, then the iterate is updated with Eq. 3.114, and then the residual is updated with Eq. 3.134.

Algorithm 1: Gradient Descent on Quadratic Form

Result: $\mathbf{x}_i$

$\mathbf{x}_1 = \mathbf{x_0}$

$\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_1$

for $i \in [1, i_{max}]$ **do**

  // tolerance test

  if $\|\mathbf{r}_i\|^2 \leq \epsilon^2$ **then**

    break

  end

  // compute search direction vector

  $\mathbf{d}_i = \mathbf{r}_i$

  // update

  $\alpha_i = \mathbf{d}_i^H \mathbf{r}_i / [\mathbf{d}_i^H \mathbf{A} \mathbf{d}_i]$

  $\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{d}_i \alpha_i$

  $\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{A}\mathbf{d}_i \alpha_i$

end

The computational complexity of this algorithm can be analyzed as a function of vectorspace dimension $n \in \mathbb{Z}^+$. Assume a single thread of execution. The space complexity is $O(n)$ per run, as only a small fixed number of vectors must be stored (the input is taken for granted). The time complexity can be analyzed per iteration and run. Each iteration is dominated by the matrix-vector multiplication $\mathbf{A}\mathbf{d}_i$ and so has time complexity $O(n^2)$. The time complexity per run is harder to assess.

Gradient descent is easy to understand but should not be used for minimizing

a quadratic form. Its main failing is poor convergence. Iterates tend to oscillate between subspaces given enough iterations. More advanced iterative solvers address this issue directly and to great effect by constructing solutions in orthogonal bases.

3.4.4 Conjugate Gradient

Continuing the discussion on the quadratic form q, the optimal search scale Eq. 3.138 can be written in terms of error using equation 3.140

$$\alpha_i = -\frac{\mathbf{d}_i^H \mathbf{A}\mathbf{e}_i}{\mathbf{d}_i^H \mathbf{A}\mathbf{d}_i} \tag{3.141}$$

and inserted into the error recurrence relation Eq. 3.119

$$\mathbf{e}_i = \mathbf{e}_0 - \sum_{j=1}^{i-1} \mathbf{d}_j \frac{\mathbf{d}_j^H \mathbf{A}\mathbf{e}_j}{\mathbf{d}_j^H \mathbf{A}\mathbf{d}_j}. \tag{3.142}$$

Observe that the summands are the negative vector-valued $\mathbf{A}$-projection of $\mathbf{e}_j$ on $\mathbf{d}_j$.

To elaborate, $\mathbf{A}$-projection indicates the underlying inner product is the $\mathbf{A}$ inner product defined to be

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{A}} \equiv \mathbf{u}^H \mathbf{A}\mathbf{v} \tag{3.143}$$

which is verified to satisfy the inner product axioms; the definition is positive-definite because $\mathbf{A}$ is positive-definite by assumption, the definition is conjugate-symmetric because $\mathbf{A}$ is conjugate-symmetric by assumption

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{A}} = \mathbf{u}^H \mathbf{A}\mathbf{v} = [\mathbf{v}^H \mathbf{A}\mathbf{u}]^H = \langle \mathbf{v}, \mathbf{u} \rangle_{\mathbf{A}}^*, \tag{3.144}$$

and the definition is linear in one of its arguments

$$\langle \mathbf{u}, \alpha\mathbf{v} + \beta\mathbf{w} \rangle_{\mathbf{A}} = \mathbf{u}^H \mathbf{A}[\alpha\mathbf{v} + \beta\mathbf{w}] = \alpha \mathbf{u}^H \mathbf{A}\mathbf{v} + \beta \mathbf{u}^H \mathbf{A}\mathbf{w} = \alpha \langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{A}} + \beta \langle \mathbf{u}, \mathbf{w} \rangle_{\mathbf{A}}. \tag{3.145}$$

The vector-valued **A**-projection of **v** on **u** is then

$$\mathbf{u}\frac{\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{A}}}{\langle \mathbf{u}, \mathbf{u} \rangle_{\mathbf{A}}} = \mathbf{u}\frac{\mathbf{u}^H \mathbf{A}\mathbf{v}}{\mathbf{u}^H \mathbf{A}\mathbf{u}}. \tag{3.146}$$

Equation 3.141 suggests that the search direction vectors $\mathbf{d}_i$ should be chosen to be **A**-orthogonal so each iteration can be interpreted as deconstructing an initial error vector in a basis one component per iteration. This is accomplished by taking the residual $\mathbf{r}_i$ as a seed search direction vector, since the negative gradient is a good guess otherwise, and making it **A**-orthogonal to previous search directions by subtracting off its **A**-projections

$$\mathbf{d}_i = \mathbf{r}_i - \sum_{j=1}^{i-1} \mathbf{d}_j \beta_{ji} \tag{3.147}$$

where the coefficients β_{ij} are

$$\beta_{ij} = \frac{\mathbf{d}_j^H \mathbf{A}\mathbf{r}_i}{\mathbf{d}_j^H \mathbf{A}\mathbf{d}_j} \tag{3.148}$$

leading to the **A**-orthogonality relation

$$\boxed{\mathbf{d}_i^H \mathbf{A}\mathbf{d}_j = 0 \quad \text{when} \quad i \neq j}. \tag{3.149}$$

Iteration on quadratic forms with this choice of search direction vectors results in the **conjugate gradient method**. The word "conjugate" in this context refers to **A**-orthogonality of the search direction vectors, and not to complex conjugation.

The naive conjugate gradient algorithm is presented as Algorithm 2. It is a modified gradient descent. The only difference is that instead of assigning the search direction vector to be the residual, it is assigned to be the part of the residual that

is **A**-orthogonal to the previous search direction vectors.

Algorithm 2: Naive Conjugate Gradient

Result: $\mathbf{x}_i$

$\mathbf{x}_1 = \mathbf{x}_0$

$\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_1$

for $i \in [1, i_{max}]$ **do**

  // tolerance test

  if $\|\mathbf{r}_i\|^2 \leq \text{tol}^2$ **then**

    break

  end

  // compute search direction vector

  $\mathbf{d}_i = \mathbf{r}_i - \sum_{j=1}^{i-1} \mathbf{d}_j [\mathbf{d}_j^H \mathbf{A}\mathbf{r}_i]/[\mathbf{d}_j^H \mathbf{A}\mathbf{d}_j]$

  // update

  $\alpha_i = \mathbf{d}_i^H \mathbf{r}_i / [\mathbf{d}_i^H \mathbf{A}\mathbf{d}_i]$

  $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$

  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{A}\mathbf{d}_i$

end

Algorithm 2 is conceptually complete, but substantial simplifications can be made with further analysis and clever substitutions. The set of search directions $\mathbf{d}_i$ forms an **A**-orthogonal basis. Consider the initial error vector expanded in this basis

$$\mathbf{e}_0 = \sum_{j=1}^{n} \mathbf{d}_j \delta_j \tag{3.150}$$

with **A**-projection expansion coefficients δ_j

$$\delta_j = \frac{\mathbf{d}_j^H \mathbf{A}\mathbf{e}_0}{\mathbf{d}_j^H \mathbf{A}\mathbf{d}_j}. \tag{3.151}$$

Insert the initial error expansion Eq. 3.150 into the error summation Eq. 3.119

$$\mathbf{e}_i = \sum_{j=1}^{n} \mathbf{d}_j \delta_j + \sum_{j=1}^{i-1} \mathbf{d}_j \alpha_j \tag{3.152}$$

and form the $\mathbf{A}$ inner product with direction vector $\mathbf{d}_i$

$$\mathbf{d}_i^H \mathbf{A}\mathbf{e}_i = \sum_{j=1}^{n} \mathbf{d}_i^H \mathbf{A}\mathbf{d}_j \delta_j + \sum_{j=1}^{i-1} \mathbf{d}_i^H \mathbf{A}\mathbf{d}_j \alpha_j = \mathbf{d}_i^H \mathbf{A}\mathbf{d}_i \delta_i. \tag{3.153}$$

Solving for δ_i and comparing to Eq. 3.151 and Eq. 3.138 implies the following relation holds for the first $i-1$ coefficients of the error summation

$$\delta_j = \frac{\mathbf{d}_j^H \mathbf{A}\mathbf{e}_0}{\mathbf{d}_j^H \mathbf{A}\mathbf{d}_j} = \frac{\mathbf{d}_j^H \mathbf{A}\mathbf{e}_j}{\mathbf{d}_j^H \mathbf{A}\mathbf{d}_j} = -\alpha_j \tag{3.154}$$

and thus the error summation simplifies to

$$\mathbf{e}_i = \sum_{j=i}^{n} \mathbf{d}_j \delta_j. \tag{3.155}$$

This means the error vector converges after n iterations. Furthermore the error vectors are $\mathbf{A}$-orthogonal to all previous search direction vectors, meaning residuals are orthogonal to all previous search direction vectors

$$-\mathbf{d}_i^H \mathbf{A}\mathbf{e}_j = \mathbf{d}_i^H \mathbf{r}_j = 0 \quad \text{for} \quad i < j. \tag{3.156}$$

Additional orthogonality relationships and useful identities can be derived from Eq. 3.147 by taking an inner product with residual $\mathbf{r}_k$

$$\mathbf{d}_i^H \mathbf{r}_k = \mathbf{r}_i^H \mathbf{r}_k - \sum_{j=1}^{i-1} \mathbf{d}_j^H \mathbf{r}_k \beta_{ij}^* \tag{3.157}$$

and considering the various cases. Using Eq. 3.156 it is seen

$$\mathbf{r}_i^H \mathbf{r}_k = 0 \quad \text{for} \quad i < k \quad \Rightarrow \quad \mathbf{r}_i^H \mathbf{r}_k = 0 \quad \text{for} \quad i \neq k \tag{3.158}$$

and

$$\mathbf{d}_i^H \mathbf{r}_i = \mathbf{r}_i^H \mathbf{r}_i \quad \text{for} \quad i = k. \tag{3.159}$$

When $k = i-1$, the previous results further imply

$$\mathbf{d}_i^H \mathbf{r}_{i-1} = -\mathbf{d}_{i-1}^H \mathbf{r}_{i-1} \beta_{i,i-1}^*. \tag{3.160}$$

These seemingly random equations are now used to make drastic simplifications.

Consider the inner product between two residuals, and expand one residual in terms of the residual summation Eq. 3.135

$$\mathbf{r}_i^H \mathbf{r}_j = \left[\mathbf{r}_0 - \sum_{k=1}^{i-1} \mathbf{A}\mathbf{d}_k \alpha_k \right]^H \mathbf{r}_j = \mathbf{r}_0^H \mathbf{r}_j - \sum_{k=1}^{i-1} \mathbf{d}_k^H \mathbf{A}\mathbf{r}_j \alpha_k^*. \tag{3.161}$$

This sum equals 0 when $i \neq j$ because of the orthogonality relations. Let $j > 3$. Consider $i = 2$ and assume $\alpha_i \neq 0$ (otherwise the iteration has converged), then $\mathbf{d}_1^H \mathbf{A}\mathbf{r}_j = 0$. Next consider $i = 3$ and take the previous result, then $\mathbf{d}_2^H \mathbf{A}\mathbf{r}_j = 0$. This pattern repeats through $i = j - 1$. When $i = j$, the inner product is no longer 0, so $\mathbf{d}_{i-1}^H \mathbf{A}\mathbf{r}_i \neq 0$. Thus

$$\mathbf{d}_j^H \mathbf{A}\mathbf{r}_i = 0 \quad \text{for} \quad j < i - 1. \tag{3.162}$$

This is a remarkable result because $\beta_{ij} = 0$ except when $j = i - 1$. In other words, residual vectors are automatically **A**-orthogonal to all previous search directions except the previous one. This means only storage for a single search direction vector is ever needed in the algorithm, and making the search directions **A**-orthogonal is inexpensive. Thus redefine $\beta_{i-1} \equiv \beta_{i,i-1}$.

Clever substitutions help reduce the computational cost of computing quantities each iteration. The α_i coefficients can be rewritten as

$$\alpha_i = \frac{\mathbf{d}_i^H \mathbf{r}_i}{\mathbf{d}_i^H A \mathbf{d}_i} = \frac{\mathbf{r}_i^H \mathbf{r}_i}{\mathbf{d}_i^H A \mathbf{d}_i} \tag{3.163}$$

and the β_{i-1} is rewritten as

$$\beta_{i-1} = -\frac{\mathbf{d}_i^H \mathbf{r}_{i-1}}{\mathbf{d}_{i-1}^H \mathbf{r}_{i-1}} = -\frac{\mathbf{d}_i^H \left[\mathbf{r}_{i-1} - \alpha_{i-1} \mathbf{A}\mathbf{d}_{i-1} \right]}{\mathbf{d}_{i-1}^H \mathbf{r}_{i-1}} = -\frac{\mathbf{d}_i^H \mathbf{r}_i}{\mathbf{d}_{i-1}^H \mathbf{r}_{i-1}} = -\frac{\mathbf{r}_i^H \mathbf{r}_i}{\mathbf{r}_{i-1}^H \mathbf{r}_{i-1}} \tag{3.164}$$

which reuses the residual norm squared computed as a byproduct of the tolerance test.

Putting the above results together, an efficient version of the conjugate gradient algorithm is presented in Algorithm 3. Comparing to Algorithm 2, the simplifications in the computation of the search direction vectors is obvious. Care has been taken

use temporary variables wherever results can be reused.

Algorithm 3: Conjugate Gradient

Result: $\mathbf{x}_i$

$i_{max} = \min(i_{max}, n)$

$\mathbf{x}_1 = \mathbf{x}_0$

$\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_1$

for $i \in [1, i_{max}]$ **do**

  // tolerance test

  rr $= \mathbf{r}_i^H \mathbf{r}_i$

  if rr $\leq$ tol^2 **then**

    break

  end

  // compute search direction vector

  if $i == 1$ **then**

    $\mathbf{d} = \mathbf{r}_1$

  else

    $\beta =$ rr / rrold

    $\mathbf{d} = \mathbf{r}_i + \beta\mathbf{d}$

  end

  // update

  $\mathbf{t} = \mathbf{A}\mathbf{d}$

  $\alpha_i =$ rr / $\mathbf{d}^H\mathbf{t}$

  $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i\mathbf{d}$

  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i\mathbf{t}$

  rrold = rr

end

A final piece of sophistication is revisiting the preconditioner. While a method to

include left preconditioning was discussed in the complex least squares problem, it is better to include preconditioners directly into the algorithm to take advantage of analytical simplifications. The goal of preconditioning is not to change the solution of the problem, but to make the linear system have a better condition number. This makes the cost function more spherical, which aids in convergence. Let $\mathbf{E} \in \mathbb{C}^{n \times n}$ be an invertible matrix, and consider the equivalent problem of solving

$$\mathbf{E}^{-1}\mathbf{A}\mathbf{E}^{-H}\mathbf{x}' = \mathbf{E}^{-1}\mathbf{b} \tag{3.165}$$

where

$$\mathbf{x}' = \mathbf{E}^{H}\mathbf{x} \tag{3.166}$$

It is easy to show the matrix $\mathbf{E}^{-1}\mathbf{A}\mathbf{E}^{-H}$ is conjugate-symmetric positive-definite, so the underlying assumptions about the problem have not been altered. Incorporating these changes into Algorithm 3 results in the following sequence of equations

$$\begin{cases} \mathbf{r}'_1 = \mathbf{E}^{-1}\mathbf{b} - \mathbf{E}^{-1}\mathbf{A}\mathbf{E}^{-H}\mathbf{x}'_0 \\ \mathbf{d}'_1 = \mathbf{r}'_1 \\ \beta_{i-1} = -\frac{\mathbf{r}'^{H}_i \mathbf{r}'_i}{\mathbf{r}'^{H}_{i-1}\mathbf{r}'_{i-1}} \\ \mathbf{d}'_i = \mathbf{r}'_i - \beta_{i-1}\mathbf{d}'_{i-1} \\ \alpha_i = \frac{\mathbf{r}'^{H}_i \mathbf{r}'_i}{\mathbf{d}'^{H}_i \mathbf{E}^{-1}\mathbf{A}\mathbf{E}^{-H}\mathbf{d}'_i} \\ \mathbf{x}'_{i+1} = \mathbf{x}'_i + \alpha_i \mathbf{d}'_i \\ \mathbf{r}'_{i+1} = \mathbf{r}'_i - \alpha_i \mathbf{E}^{-1}\mathbf{A}\mathbf{E}^{-H}\mathbf{d}'_i \end{cases} \tag{3.167}$$

which seems to add a great deal of complexity. However the following substitutions

$$\begin{cases} \mathbf{r}'_i = \mathbf{E}^{-1}\mathbf{r}_i \\ \mathbf{d}'_i = \mathbf{E}^{H}\mathbf{d}_i \\ \mathbf{x}'_i = \mathbf{E}^{H}\mathbf{x}_i \\ \mathbf{E}^{-H}\mathbf{E}^{-1} = \mathbf{P}^{-1} \end{cases} \tag{3.168}$$

show only modest changes must be made to the algorithm

$$\begin{cases} \mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 \\ \mathbf{d}_1 = \mathbf{P}^{-1}\mathbf{r}_1 \\ \beta_{i-1} = -\frac{\mathbf{r}_i^H \mathbf{P}^{-1}\mathbf{r}_i}{\mathbf{r}_{i-1}^H \mathbf{P}^{-1}\mathbf{r}_{i-1}} \\ \mathbf{d}_i = \mathbf{P}^{-1}\mathbf{r}_i - \beta_{i-1}\mathbf{d}_{i-1} \\ \alpha_i = \frac{\mathbf{r}_i^H \mathbf{P}^{-1}\mathbf{r}_i}{\mathbf{d}_i^H \mathbf{A}\mathbf{d}_i} \\ \mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i \\ \mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{A}\mathbf{d}_i \end{cases} \tag{3.169}$$

The resulting preconditioned conjugate gradient method is presented in Algorithm 4.

Note, the preconditioner is sometimes specified in its factored form $\mathbf{P}^{-1} = \mathbf{E}^{-H}\mathbf{E}^{-1}$.

Algorithm 4: Preconditioned Conjugate Gradient

Result: $\mathbf{x}_i$

$i_{max} = \max(i_{min}, n)$

$\mathbf{x}_1 = \mathbf{x}_0$

$\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_1$

for $i \in [1, i_{max}]$ **do**

  // tolerance test

  if $\|\mathbf{r}_i\|^2 \leq \text{tol}^2$ **then**

    break

  end

  // compute preconditioned search direction vector

  $\mathbf{z}_i = \mathbf{P}^{-1}\mathbf{r}_i$

  rz $= \mathbf{r}_i^H \mathbf{z}_i$

  if $i == 1$ **then**

    $\mathbf{d} = \mathbf{z}_1$

  else

    $\beta =$ rz / rzold

    $\mathbf{d} = \mathbf{z}_i + \beta\mathbf{d}$

  end

  // update

  $\mathbf{t} = \mathbf{A}\mathbf{d}$

  $\alpha_i =$ rz / $\mathbf{d}^H\mathbf{t}$

  $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i\mathbf{d}$

  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i\mathbf{t}$

  rzold = rz

end

The space complexity of conjugate gradient is $O(n)$ rather than $O(n^2)$ because only the previous search direction vector is needed when making the new search direction **A**-orthogonal to all the previous search directions. The time complexity each iteration is dominated by two matrix-vector products, and so is $O(n^2)$ per iteration. The algorithm is guaranteed to converge after n iterations, so the time complexity per run is $O(n^3)$. This doesn't seem like an improvement over Gauss-Jordan elimination, however conjugate gradient gives a sequence of approximate answers. Convergence is shown to be quite fast, and so conjugate gradient is usually terminated long before n iterations with good results.

3.4.5 GMRES

An alternative to gradient-based iterative methods is **General Minimal Residual (GMRES)**, which iteratively minimizes the complex least squares cost function directly. GMRES only assumes that matrix **A** is invertible, and is thus a more general solver than the gradient-based methods presented in the previous sections which assumed a conjugate-symmetric positive-definite matrix. However, reformulating an ill-posed problem as a well-posed complex least squares problem with regularization produces a conjugate-symmetric positive-definite matrix, so this isn't much of an advantage for image reconstruction.

Recall the complex least squares cost function is

$$\mathbf{x}_s = \underset{\mathbf{x}\in\mathbb{C}^n}{\arg\min} \left\|\mathbf{b} - \mathbf{A}\mathbf{x}\right\|^2 . \tag{3.170}$$

Iterate summation Eq. 3.115 can be written in matrix notation as

$$\mathbf{x}_i = \mathbf{x}_0 + \mathbf{D}_i\boldsymbol{\alpha}_i \tag{3.171}$$

where $\mathbf{D}_i \in \mathbb{C}^{n\times i}$ is a search direction augmented matrix and $\boldsymbol{\alpha}_i \in \mathbb{C}^i$ is a vector of search scales. Combine these equations to yield

$$\|\mathbf{r}_i\|^2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}_i\|^2 = \|\mathbf{b} - \mathbf{A}[\mathbf{x}_0 + \mathbf{D}_i\boldsymbol{\alpha}_i]\|^2 = \|\mathbf{r}_1 - \mathbf{A}\mathbf{D}_i\boldsymbol{\alpha}_i\|^2 . \tag{3.172}$$

The search directions must be specified to proceed. As with conjugate gradient, a judicious choice of search direction vectors greatly simplifies the problem.

A deeper understanding of iterative methods can be achieved by thinking in terms of **Krylov subspaces**. The i-th Krylov subspace of matrix $\mathbf{A}$ and vector $\mathbf{u}$ is the vectorspace spanned by the sequence of vectors generated by repeatedly applying $\mathbf{A}$ to $\mathbf{u}$ up to i times

$$K_i(\mathbf{A}, \mathbf{u}) \equiv \text{span}\left(\bigcup_{j=0}^{i} \left\{\mathbf{A}^j\mathbf{u}\right\}\right) = \text{span}\left(\left\{\mathbf{u}, \mathbf{A}\mathbf{u}, \mathbf{A}^2\mathbf{u}, \ldots, \mathbf{A}^i\mathbf{u}\right\}\right). \tag{3.173}$$

Elements of a Krylov subspace are linearly independent up to some positive integer which can be smaller than or equal the dimension of the underlying vectorspace. For both gradient descent and conjugate gradient the search direction vector $\mathbf{d}_i$ is an element of the i-th Krylov subspace of the first search direction vector $\mathbf{d}_1 = \mathbf{r}_1$

$$\mathbf{d}_i \in K_i(\mathbf{A}, \mathbf{r}_1). \tag{3.174}$$

This is called the Arnoldi iteration. Furthermore, conjugate gradient built an $\mathbf{A}$-orthogonal basis out of the search direction vectors which guaranteed the algorithm would converge after at most n iterations.

GMRES takes a similar tack to conjugate gradient and builds an orthonormal set of Krylov subspace vectors from Eq. 3.174 such that

$$\left\langle \hat{\mathbf{d}}_i, \hat{\mathbf{d}}_j \right\rangle = \delta_{ij}. \tag{3.175}$$

The seed search direction is taken to be the initial residual $\mathbf{r}_1$, although the residual can't be interpreted as a gradient in the case of GMRES. This is acceptable because the non-zero seed is arbitrary, and is a good choice for the subset of conjugate-symmetric positive-definite systems that GMRES can solve. In the case this Krylov subspace fails to span the underlying vectorspace, GMRES will converge in fewer than n iterations. With this choice of search direction vectors, the search direction augmented matrix satisfies the recurrence relation

$$\mathbf{A}\mathbf{D}_i = \mathbf{D}_{i+1}\mathbf{H}_{i+1,i} \tag{3.176}$$

where $\mathbf{H}_{i+1,i} \in \mathbb{C}^{[i+1]\times i}$ is a matrix of expansion coefficients. Because of the Arnoldi iteration, matrix $\mathbf{H}_{i+1,i}$ is zero below the first sub-diagonal (like an upper Hessenburg matrix), and is computed as a byproduct of taking projections when making the search directions orthonormal. Furthermore, the matrix $\mathbf{H}_{i+1,i}$ is related to the next matrix $\mathbf{H}_{i+2,i+1}$ by the recurrence relation

$$\mathbf{H}_{i+2,i+1} = \begin{bmatrix} \mathbf{H}_{i+1,i} & \mathbf{h}_{i+1} \\ \mathbf{0}^H & H_{i+2,i+1} \end{bmatrix} \tag{3.177}$$

where vector $\mathbf{h}_{i+2} \in \mathbb{C}^{i+2}$ is a new column. The base case is matrix $\mathbf{H}_{2,1} \in \mathbb{C}^{2\times 1}$.

Returning to the complex least squares cost function Eq. 3.172, use Eq. 3.176 and the identity $\hat{\mathbf{d}}_1 = \mathbf{r}_1 / \|\mathbf{r}_1\|$ to write

$$\begin{aligned} \|\mathbf{r}_i\|^2 &= \|\mathbf{r}_1 - \mathbf{A}\mathbf{D}_i\boldsymbol{\alpha}_i\|^2 \\ &= \left\| \|\mathbf{r}_1\| \hat{\mathbf{d}}_1 - \mathbf{D}_{i+1}\mathbf{H}_{i+1,i}\boldsymbol{\alpha}_i] \right\|^2 \\ &= \|\mathbf{D}_{i+1}[\|\mathbf{r}_1\| \hat{\mathbf{e}}_1 - \mathbf{H}_{i+1,i}\boldsymbol{\alpha}_i]\|^2 \\ &= \|\|\mathbf{r}_1\| \hat{\mathbf{e}}_1 - \mathbf{H}_{i+1,i}\boldsymbol{\alpha}_i\|^2 \end{aligned} \tag{3.178}$$

where $\hat{\mathbf{e}} \in \mathbb{C}^{i+1}$ is the first canonical basis unit vector. The final step follows because $\mathbf{D}_{i+1}$ is an augmented matrix of orthonormal vectors

$$\|\mathbf{D}_i\mathbf{u}\|^2 = \langle \mathbf{D}_i\mathbf{u}, \mathbf{D}_i\mathbf{u} \rangle = \mathbf{u}^H\mathbf{D}_i^H\mathbf{D}_i\mathbf{u} = \mathbf{u}^H\mathbf{I}\mathbf{u} = \mathbf{u}^H\mathbf{u} = \langle \mathbf{u}, \mathbf{u} \rangle = \|\mathbf{u}\|^2 . \tag{3.179}$$

The problem of minimizing the residual norm with respect to $\mathbf{x}$ is replaced by the equivalent problem of minimizing the residual norm with respect to $\boldsymbol{\alpha}_i$

$$\boldsymbol{\alpha}_i = \underset{\boldsymbol{\alpha}_i' \in \mathbb{C}^i}{\arg\min} \left\| \|\mathbf{r}_1\| \, \hat{\mathbf{e}}_1 - \mathbf{H}_{i+1,i}\boldsymbol{\alpha}_i' \right\|^2 . \tag{3.180}$$

This equation can be efficiently solved with some manipulation.

Every matrix has a QR decomposition that factors the matrix into a unitary matrix multiplying a right triangular matrix. The existence of such a decomposition is easy to establish by considering orthonormalization applied to the columns of a matrix. For the task at hand consider the QR decomposition of $\mathbf{H}_{i+1,i}$

$$\mathbf{H}_{i+1,i} = \mathbf{Q}_{i+1}\mathbf{R}_{i+1,i} \tag{3.181}$$

for unitary matrix $\mathbf{Q}_{i+1} \in \mathbb{C}^{[i+1]\times[i+1]}$ and right triangular matrix $\mathbf{R}_{i+1,i} \in \mathbb{C}^{[i+1]\times i}$. Recall the defining property of a unitary matrix is $\mathbf{Q}^{-1} = \mathbf{Q}^H$, which implies the columns are orthonormal. Thus the right triangular matrix in the previous equation can be solved for

$$\mathbf{Q}_{i+1}^H\mathbf{H}_{i+1,i} = \mathbf{R}_{i+1,i} = \begin{bmatrix} \mathbf{R}_i \\ \mathbf{0}^H \end{bmatrix} . \tag{3.182}$$

Matrix $\mathbf{H}_{i+1,i}$ is transformed into matrix $\mathbf{R}_{i+1,i}$ when multiplied by matrix $\mathbf{Q}_{i+1}^H$. Right triangular matrices are trivial to solve by backsubstitution, so this is a highly

desirable transformation. Rewrite Eq. 3.178 using the proposed QR decomposition

$$\begin{aligned}
\|\mathbf{r}_i\|^2 &= \left\| \|\mathbf{r}_0\| \, \hat{\mathbf{e}}_1 - \mathbf{H}_{i+1,i}\boldsymbol{\alpha}_i \right\|^2 \\
&= \left\| \mathbf{Q}_{i+1}^H [\|\mathbf{r}_0\| \, \hat{\mathbf{e}}_1 - \mathbf{H}_{i+1,1}\boldsymbol{\alpha}_i] \right\|^2 \\
&= \left\| \|\mathbf{r}_0\| \, \mathbf{Q}_{i+1}^H \hat{\mathbf{e}}_1 - \mathbf{R}_{i+1,i}\boldsymbol{\alpha}_i \right\|^2 \\
&= \left\| \begin{bmatrix} \mathbf{g}_i \\ g_{i+1} \end{bmatrix} - \begin{bmatrix} \mathbf{R}_i\boldsymbol{\alpha}_i \\ 0 \end{bmatrix} \right\|^2 .
\end{aligned} \tag{3.183}$$

The residual norm is minimized when

$$\boldsymbol{\alpha}_i = \mathbf{R}_i^{-1}\mathbf{g}_i \tag{3.184}$$

which implies $g_{i+1} = \|\mathbf{r}_i\|$.

The QR decomposition is only useful if it is inexpensive to compute. This is shown to be the case. From recurrence relation Eq. 3.177 only the right-most column of $\mathbf{H}_{i+1,i}$ needs to be computed each iteration, which is done as a byproduct of making the search direction vector orthonormal to previous search direction vectors. Recursion is then used to compute $\mathbf{R}_i$. Consider the base case of Eq. 3.182 when $i = 1$

$$\begin{bmatrix} c & -s \\ s^* & c^* \end{bmatrix}^H \begin{bmatrix} H_{1,1} \\ H_{2,1} \end{bmatrix} = \begin{bmatrix} R_{1,1} \\ 0 \end{bmatrix} \tag{3.185}$$

where the $\mathbf{Q}_2$ takes the form of a (Givens) "rotation"

$$\mathbf{Q}_2 = \begin{bmatrix} c & -s \\ s^* & c^* \end{bmatrix} \quad c = \frac{H_{1,1}}{R_{1,1}} \quad s = \frac{H_{2,1}^*}{R_{1,1}} \quad R_{1,1} = \sqrt{|H_{1,1}|^2 + |H_{2,1}|^2}. \tag{3.186}$$

To derive a recurrence relation for the QR decomposition, start with the recurrence relation Eq. 3.177 and incorporate the previous QR decomposition to get the partial

result

$$\begin{bmatrix} \mathbf{Q}_{i+1}^H & \mathbf{0} \\ \mathbf{0}^H & 1 \end{bmatrix} \begin{bmatrix} \mathbf{H}_{i+1,i} & \mathbf{h}_{i+1} \\ \mathbf{0}^H & H_{i+2,i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{i+1,i} & \mathbf{Q}_{i+1}^H \mathbf{h}_{i+1} \\ \mathbf{0}^H & H_{i+2,i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i & \mathbf{r}_i \\ \mathbf{0}^H & \rho_i \\ \mathbf{0}^H & \sigma_i \end{bmatrix}. \tag{3.187}$$

The right hand side is almost right triangular. All that needs to be done is another rotation on the last two rows. Generalizing from the base case, construct the matrix $\mathbf{G}_{i+2} \in \mathbb{C}^{[i+2]\times[i+2]}$ such that

$$\mathbf{G}_{i+2} = \begin{bmatrix} \mathbf{I}_i & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^H & c_i & -s_i \\ \mathbf{0}^H & s_i^* & c_i^* \end{bmatrix} \quad \text{where} \quad c_i = \frac{\rho_i}{\tau_i} \quad s_i = \frac{\sigma_i^*}{\tau_i} \quad \tau_i = \sqrt{|\rho_i|^2 + |\sigma_i|^2} \tag{3.188}$$

and left multiply Eq. 3.187 to derive the QR right triangular matrix recurrence relation

$$\begin{bmatrix} \mathbf{I}_i & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^H & c_i^* & s_i \\ \mathbf{0}^H & -s_i^* & c_i \end{bmatrix} \begin{bmatrix} \mathbf{R}_i & \mathbf{r}_i \\ \mathbf{0}^H & \rho_i \\ \mathbf{0}^H & \sigma_i \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i & \mathbf{r}_i \\ \mathbf{0}^H & \tau_i \\ \mathbf{0}^H & 0 \end{bmatrix} = \mathbf{R}_{i+2,i+1}. \tag{3.189}$$

It is seen that only the right-most column of must be computed each iteration. Furthermore this implies the QR unitary matrix recurrence relation

$$\mathbf{Q}_{i+2}^H = \mathbf{G}_{i+2}^H \begin{bmatrix} \mathbf{Q}_{i+1}^H & \mathbf{0} \\ \mathbf{0}^H & 1 \end{bmatrix}. \tag{3.190}$$

Taken together, only the right-most column of $\mathbf{H}_{i+1,i}$ and $\mathbf{R}_{i+1,i}$ need to be computed each iteration, the column of $\mathbf{H}_{i+1,i}$ is computed during orthonormalization, and the column of $\mathbf{H}_{i+1,i}$ is transformed into the column of $\mathbf{R}_{i+1,i}$ by applying $\mathbf{Q}_{i+1}$ which is equivalent to a sequence of rotations between pairs of rows.

The GMRES algorithm is now presented in Algorithm 5. Due to the solution being constructed in a basis, at most n iterations are needed for convergence. Seed the

Arnoldi process with the residual $\mathbf{r}_1$ and compute $\mathbf{g}_1$ from the equivalent cost function. For each iteration several steps must happen. First compute a new search direction based on the Arnoldi iteration (apply $\mathbf{A}$ to the previous search direction vector) and make it orthonormal to the previous search direction vectors using a stabilized algorithm, saving the expansion coefficients as the new column of $\mathbf{H}$. Next transform the $\mathbf{H}$ column by applying the sequence of rotations up to the last rotation. Then compute the current rotation and complete the transformation of the $\mathbf{H}$ column into an $\mathbf{R}$ column. Examining the equivalent cost function, vector $\mathbf{g}$ must be transformed by the rotations along with $\mathbf{H}$, but unlike the entirely new column of $\mathbf{H}$, only the most recent rotation needs to be applied to vector $\mathbf{g}$ to update it. Finally, check the tolerance for early convergence. After the loop the solution vector is constructed by solving Eq. 3.184 for $\boldsymbol{\alpha}_i$ by backsubstitution, and then evaluating the matrix form of the iterate summation Eq. 3.187. Preconditioning can be incorporated if needed.

The computational complexity of GMRES should be considered. Due to the need to store all previous search direction vectors to apply orthonormalization, the spatial complexity is $O(n^2)$. This is a major short-coming of GMRES. Many modifications to the basic algorithm try to circumvent this by periodically restarting the algorithm, however in this case convergence isn't guaranteed. The time complexity per iteration is dominated by a matrix-vector product, and is so $O(n^2)$. The time complexity per run if allowed to iterate a full n times is $O(n^3)$. Like conjugate gradient, this seems to be the same as Gauss-Jordon elimination, however GMRES provides a sequence of approximate solution vectors that converge quickly, and can often be terminated

long before a full n iterations. Convergence is more difficult to assess.

Algorithm 5: General Minimal Residual

Result: $\mathbf{x}_i$
$i_{max} = \min(i_{max}, n)$
$\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{d}_1 = \mathbf{r}_1 / \|\mathbf{r}_1\|$
$\mathbf{g}_1 = \|\mathbf{r}_1\| \hat{\mathbf{e}}_1$
for $i \in [1, i_{max}]$ **do**
  // stabilized arnoldi process
  $\mathbf{d}_{i+1} = \mathbf{A}\hat{\mathbf{d}}_i$
  for $j \in [1.i]$ **do**
    $H_{j,i} = \langle \hat{\mathbf{d}}_j, \mathbf{d}_{i+1} \rangle$
    $\mathbf{d}_{i+1} = \mathbf{d}_{i+1} - \hat{\mathbf{d}}_j H_{j,i}$
  end
  $H_{i+1,i} = \|\mathbf{d}_{i+1}\|$
  $\hat{\mathbf{d}}_{i+1} = \mathbf{d}_{i+1} / \|\mathbf{d}_{i+1}\|$
  // transform column of H to (almost) column of R
  for $j \in [1, i-1]$ **do**
    $\begin{bmatrix} H_{j,i} \\ H_{j+1,i} \end{bmatrix} = \begin{bmatrix} c_j^* & s_j \\ -s_j^* & c_j \end{bmatrix} \begin{bmatrix} H_{j,i} \\ H_{j+1,i} \end{bmatrix}$
  end
  // compute rotation
  $\tau = \sqrt{|H_{i,i}|^2 + |H_{i+1,i}|^2}$
  $c_i = H_{i,i} / \tau$
  $s_i = H_{i+1,i}^* / \tau$
  // complete transform to R
  $\begin{bmatrix} H_{i,i} \\ H_{i+1,i} \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}$
  // transform g
  $\begin{bmatrix} g_i \\ g_{i+1} \end{bmatrix} = \begin{bmatrix} c_j^* & s_j \\ -s_j^* & c_j \end{bmatrix} \begin{bmatrix} g_i \\ g_{i+1} \end{bmatrix}$
  // tolerance test
  if $g_{i+1} \leq$ tol **then**
    break
  end
end
$\begin{bmatrix} \mathbf{R}_i \\ \mathbf{0}^H \end{bmatrix} = \mathbf{H}_{i+1,i}$
$\boldsymbol{\alpha}_i = \mathbf{R}_i^{-1} \mathbf{g}_i$
$\mathbf{x}_i = \mathbf{x}_0 + \mathbf{D}_i \boldsymbol{\alpha}_i$
return $\mathbf{x}_i$

3.5 Summary

This chapter presented different ways to invert the linear forward model to reconstruct an image. Algorithms and their complexities were derived.

SVD pseudoinverse was briefly discussed. The SVD singular value spectrum can be used to analyze the quality of a forward model. A flat spectrum indicates low correlation between measurements which is desirable so every measurement gives new information, while a steeply falling spectrum indicates the opposite. Truncating the spectrum can deal with poorly conditioned systems and nullspaces. However an SVD is expensive to compute for very large systems.

Matched filter reconstruction was presented as a simple and fast approximate inverse that takes advantage of the statistical properties of a random matrix. It provides adequate results for little computational cost.

The inverse problem is almost always ill-posed. Complex least squares with L^2 regularization was developed to form a well-posed linear inverse problem with a unique solution, which results in a conjugate-symmetric positive-definite matrix. While the well-posed problem can be solved directly with Gauss-Jordon elimination, this is computationally expensive for very large linear systems and potentially unstable.

Iterative methods become the only viable method for inverting very large linear systems. These algorithms generate a sequence of approximate solutions, and can be stopped short of an exact solution. The basic iterative algorithm framework was discussed, and applied to three algorithms. Gradient descent is conceptually simple, but has terrible convergence. Conjugate gradient dramatically improved upon gradient descent by carefully choosing a basis to expand a solution in with remarkable properties. These two algorithms are based on gradients and require the linear system

to be conjugate-symmetric and positive-definite, which the complex least squares matrix is. GMRES was presented as an alternative to gradient-based algorithms, and can be applied as long as the linear system is invertible.

Other topics tangentially related to image reconstruction are discussed in other chapters. Notably, APB matrix partitioning is used to slash the time complexity of matrix multiplication by the imager forward model on GPU-enabled hardware (Chapter 2). Depth cameras are also used to slash the size of the inverse problem by identifying a region of interest (Chapter 5). These techniques are invaluable for reaching the goal of a real-time imaging system.

Finally, additional forward models and reconstruction strategies exist. With some prepossessing of antenna data, it is possible to use RMA. For a completely different approach to imaging based on asymptotic series see FAMI.

Chapter 4

Registration

Registration is the process of estimating the spatial relationship between things. Accurate registration is a prerequisite to several core metaimager tasks including antenna registration, RF calibration with a calibration object, depth camera sensor fusion, and image stitching. There is significant theoretical overlap between the registration tasks. This chapter describes the mathematics of rigid transformations, describes a method to estimate the optimal rigid transformation between sets of points, and describes a constellation concept to experimentally register rigid bodies. These ideas are then applied to solve the crucial task of antenna registration. Other applications are discussed in later chapters.

4.1 Rigid Transformations

A **rigid body** is a physical idealization of a solid object that doesn't change shape over time. More formally, a rigid body is a set V of points in $\mathbb{R}^3$ that depend on time

$$V \subset \left\{ \mathbf{r}(t) | \mathbf{r} : \mathbb{R} \to \mathbb{R}^3 \right\} \tag{4.1}$$

where the (Euclidean) distances between points $\mathbf{r}, \mathbf{r}' \in V$ are constant in time

$$\frac{d}{dt} \left\| \mathbf{r}(t) - \mathbf{r}'(t) \right\|^2 = 0. \tag{4.2}$$

It follows that the instantaneous configuration of a rigid body is described by an **isometry**, which is a transformation that preserves distance. Examples of isometries

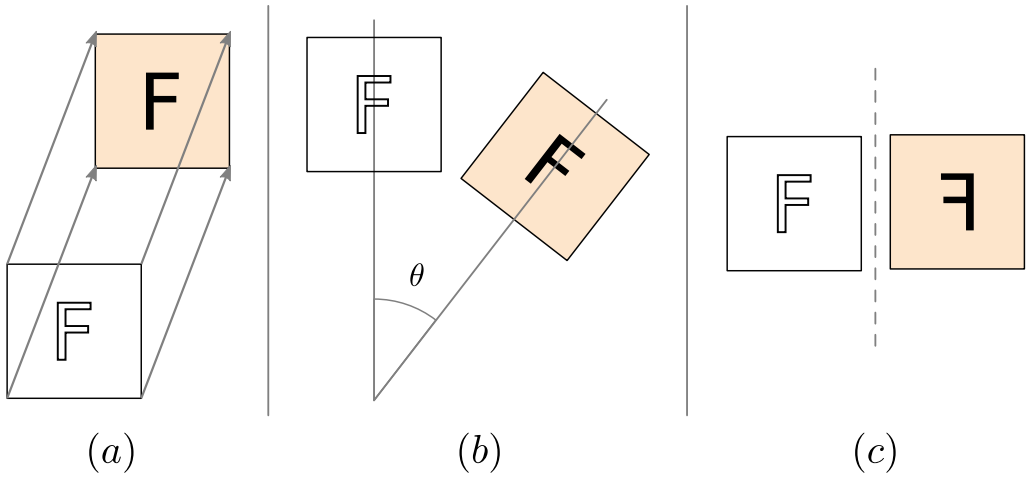

Figure 4.1: Isometries — (a) translation, (b) rotation around an axis, and (c) reflection across a plane. Combinations of these are also isometries.

are translations, rotations, reflections, and combinations of these (Fig. 4.1). In general an isometry on $\mathbb{R}^3$ can be represented by an affine map $A: \mathbb{R}^3 \to \mathbb{R}^3$ of the form

$$A(\mathbf{r}) = \mathbf{Q}\mathbf{r} + \mathbf{t} \tag{4.3}$$

for an **orthogonal matrix** $\mathbf{Q} \in O(3) = \left\{\mathbf{M} \in \mathbb{R}^{3\times 3} | \mathbf{M}^T = \mathbf{M}^{-1}\right\}$ and constant **translation vector** $\mathbf{t} \in \mathbb{R}^3$. Exactly in this case the distance between transformed points is equal to the distance between untransformed points

$$\|A(\mathbf{r}) - A(\mathbf{r}')\|^2 = \|[\mathbf{Q}\mathbf{r} + \mathbf{t}] - [\mathbf{Q}\mathbf{r}' + \mathbf{t}]\|^2 = \|\mathbf{Q}[\mathbf{r} - \mathbf{r}']\|^2 = \|\mathbf{r} - \mathbf{r}'\|^2 \tag{4.4}$$

where the last step follows from the orthogonal matrix identity

$$\|\mathbf{Q}\mathbf{u}\|^2 = [\mathbf{Q}\mathbf{u}]^T\mathbf{Q}\mathbf{u} = \mathbf{u}^T\mathbf{Q}^T\mathbf{Q}\mathbf{u} = \mathbf{u}^T\mathbf{Q}^{-1}\mathbf{Q}\mathbf{u} = \mathbf{u}^T\mathbf{u} = \|\mathbf{u}\|^2 . \tag{4.5}$$

The set of all isometries on $\mathbb{R}^3$ is denoted $E(3) = \{A(\mathbf{r}) = \mathbf{Q}\mathbf{r} + \mathbf{t} | \mathbf{Q} \in O(3), \mathbf{t} \in \mathbb{R}^3\}$.

Elements of $E(3)$ are not generally linear maps

$$A(\mathbf{u}a + \mathbf{u}'a') = \mathbf{Q}[\mathbf{u}'a + \mathbf{u}'a'] + \mathbf{t} \neq [\mathbf{Q}\mathbf{u} + \mathbf{t}]a + [\mathbf{Q}\mathbf{u}' + \mathbf{t}]a' = A(\mathbf{u})a + A(\mathbf{u}')a' \quad (4.6)$$

but can always be represented as linear maps on 4D projective coordinates (App. B)

$$\begin{bmatrix} \mathbf{Q} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \left[\begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix} a + \begin{bmatrix} \mathbf{u}' \\ 1 \end{bmatrix} a' \right] = \begin{bmatrix} \mathbf{Q}\mathbf{u} + \mathbf{t} \\ 1 \end{bmatrix} a + \begin{bmatrix} \mathbf{Q}\mathbf{u}' + \mathbf{t} \\ 1 \end{bmatrix} a'. \quad (4.7)$$

Projective coordinates are used in code to internally represent geometry partially because of this property. However their unique properties are not required for registration, so this chapter will adopt regular coordinates.

Physically the orthogonal matrix $\mathbf{Q} \in O(3)$ must not be a reflection because a rigid body can't be transformed into its mirror image smoothly in time (a left hand can't become a right hand). This is related to the characteristic property that orthogonal matrices have real determinants with unit magnitude

$$1 = \det(\mathbf{1}) = \det(\mathbf{Q}\mathbf{Q}^{-1}) = \det(\mathbf{Q}\mathbf{Q}^T) = \det(\mathbf{Q})\det(\mathbf{Q}^T) = \det(\mathbf{Q})^2 \quad (4.8)$$

that may equal $+1$ or -1 by example of an identity and reflection matrix respectively

$$\det(\mathbf{Q}) \in \{1, -1\}. \quad (4.9)$$

This property demonstrates no continuous path can exist between elements of $O(3)$ with determinants of opposite sign, and thus $O(3)$ is disconnected. Therefore exclude reflections by restricting $O(3)$ to the set of **rotations** with determinant $+1$

$$\boxed{SO(3) = \left\{\mathbf{M} \in \mathbb{R}^{3\times3} | \det(\mathbf{M}) = 1\right\}}. \quad (4.10)$$

The corresponding restricted set of isometries are the **rigid transformations**

$$\boxed{SE(3) = \left\{A(\mathbf{r}) = \mathbf{R}\mathbf{r} + \mathbf{t} | \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3\right\}}. \quad (4.11)$$

Elements of $SE(3)$ can be interpreted as an **active transformation** that moves the points of a rigid body with respect to a stationary orthonormal basis (Fig. 4.2.a), or equivalently as a **passive transformation** change of basis between like-handed orthonormal bases describing stationary points (Fig. 4.2.b). Subscripts are introduced to aid in the change of basis interpretation

$$\boxed{\mathbf{r}_b = A_{ba}(\mathbf{r}_a) \equiv \mathbf{R}_{ba}\mathbf{r}_a + \mathbf{t}_{ba}}. \tag{4.12}$$

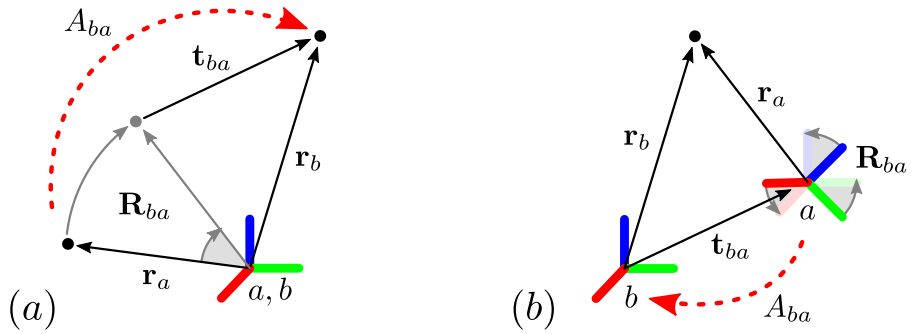

Figure 4.2: Rigid Transformations — Transformation by $A_{ba} \in SE(3)$ can be interpreted as (a) an active transformation acting on points or (b) a passive transformation acting on bases.

$SE(3)$ has the structure of a mathematical group under functional composition. Practically this guarantees inverses always exist. This is proven by verifying the 4

group axioms . Elements of $SE(3)$ are closed under functional composition

$$A_{cb}(A_{ba}(\mathbf{r}_a)) = \mathbf{R}_{cb}[\mathbf{R}_{ba}\mathbf{r}_a+\mathbf{t}_{ba}]+\mathbf{t}_{cb} = \mathbf{R}_{cb}\mathbf{R}_{ba}\mathbf{r}_a+[\mathbf{R}_{cb}\mathbf{t}_{ba}+\mathbf{t}_{cb}] = \mathbf{R}_{ca}\mathbf{r}_a+\mathbf{t}_{ca} = A_{ca}(\mathbf{r}_a) \tag{4.13}$$

where $\mathbf{R}_{ca} = \mathbf{R}_{cb}\mathbf{R}_{ba} \in SO(3)$ because $\det(\mathbf{R}_{cb}\mathbf{R}_{ba}) = \det(\mathbf{R}_{cb})\det(\mathbf{R}_{ba}) = 1$ and $\mathbf{t}_{ca} = \mathbf{R}_{cb}\mathbf{t}_{ba} + \mathbf{t}_{cb}$. This is concisely written with functional composition notation as

$$\boxed{A_{cb} \circ A_{ba} = A_{ca}}. \tag{4.14}$$

Functional composition is always associative

$$\boxed{A_{dc} \circ [A_{cb} \circ A_{ba}] = [A_{dc} \circ A_{cb}] \circ A_{ba}}. \tag{4.15}$$

There exists an identity element $A_{id} \in SE(3)$ with $\mathbf{R}_{id} = \mathbf{1} \in SO(3)$ and $\mathbf{t}_{id} = \mathbf{0} \in \mathbb{R}^3$

$$\boxed{A_{id}(\mathbf{r}) = \mathbf{1}\mathbf{r} + \mathbf{0} = \mathbf{r}} \tag{4.16}$$

which is both a left and right identity element

$$A_{id}(A_{ba}(\mathbf{r}_a)) = \mathbf{1}[\mathbf{R}_{ba}\mathbf{r}_a + \mathbf{t}_{ba}] + \mathbf{0} = \mathbf{R}_{ba}\mathbf{r}_a + \mathbf{t}_{ba} = A_{ba}(\mathbf{r}_a) \tag{4.17}$$

$$A_{ba}(A_{id}(\mathbf{r}_a)) = \mathbf{R}_{ba}[\mathbf{1}\mathbf{r}_a + \mathbf{0}] + \mathbf{t}_{ba} = \mathbf{R}_{ba}\mathbf{r}_a + \mathbf{t}_{ba} = A_{ba}(\mathbf{r}_a) \tag{4.18}$$

so

$$\boxed{A_{id} \circ A_{ba} = A_{ba} \circ A_{id} = A_{ba}}. \tag{4.19}$$

Also for all $A_{ba} \in SE(3)$ there exists an inverse element $A_{ba}^{-1} \in SE(3)$

$$\boxed{A_{ba}^{-1}(\mathbf{r}_b) = \mathbf{R}_{ba}^{-1}[\mathbf{r}_b - \mathbf{t}_{ba}] = \mathbf{r}_a} \tag{4.20}$$

that is verified to be both a left and right inverse of A_{ba}

$$A_{ba}^{-1}(A_{ba}(\mathbf{r}_a)) = \mathbf{R}_{ba}^{-1}[[\mathbf{R}_{ba}\mathbf{r}_a + \mathbf{t}_{ba}] - \mathbf{t}_{ba}] = \mathbf{1}\mathbf{r}_a + \mathbf{0} = A_{id}(\mathbf{r}_a) \tag{4.21}$$

$$A_{ba}(A_{ba}^{-1}(\mathbf{r}_b)) = \mathbf{R}_{ba}[\mathbf{R}_{ba}^{-1}[\mathbf{r}_b - \mathbf{t}_{ba}]] + \mathbf{t}_{ba} = \mathbf{1}\mathbf{r}_b + \mathbf{0} = A_{id}(\mathbf{r}_b) \tag{4.22}$$

so

$$\boxed{A_{ba}^{-1} \circ A_{ba} = A_{ba} \circ A_{ba}^{-1} = A_{id}}. \tag{4.23}$$

An inverse can be written as a forward transformation with the subscripts swapped

$$\boxed{A_{ab} = A_{ba}^{-1}} \tag{4.24}$$

by taking $\mathbf{R}_{ab} = \mathbf{R}_{ba}^{-1}$ and $\mathbf{t}_{ab} = -\mathbf{R}_{ba}^{-1}\mathbf{t}_{ba}$, which lends flexibility to the basis subscript notation and is used to simplify expressions when composing several transformations. Together, Eq. 4.14, Eq. 4.15, Eq. 4.19, and Eq. 4.23 imply $SE(3)$ is a group under functional composition. It is easy to verify $O(3)$, $SO(3)$, and $E(3)$ are also groups under functional composition.

In summary, the position and orientation of rigid bodies are described by rigid transformations. Rigid transformations can be interpreted either as actively transforming geometry or passively transforming bases. Notation was introduced to aid in the change of basis interpretation. Also, rigid transformations form a group under functional composition. These facts establish the basic mathematics of rigid transformations. Applications involving rigid transformations can now be considered.

4.2 Orthogonal Procrustes Analysis

An important registration problem is computing an optimal rigid transformation $A_{ba} \in SE(3)$ that maps a finite set of rigid points[1] $\mathbf{a}_i \in V_a \subset \mathbb{R}^3$ to a paired set of target rigid points $\mathbf{b}_i \in V_b \subset \mathbb{R}^3$ for indices $i \in [1, N] \subset \mathbb{Z}$, as illustrated in Fig. 4.3.a. **Orthogonal Procrustes Analysis** [Sch66] defines an optimal solution

[1] In this section points $\mathbf{r}_a$ and $\mathbf{r}_b$ will be denoted as $\mathbf{a}$ and $\mathbf{b}$ to economize on subscripts.

as one that minimizes a geometrically motivated cost function. It is named after a legendary Greek host known to stretch and cut his guests to fit his ill-fitting beds.

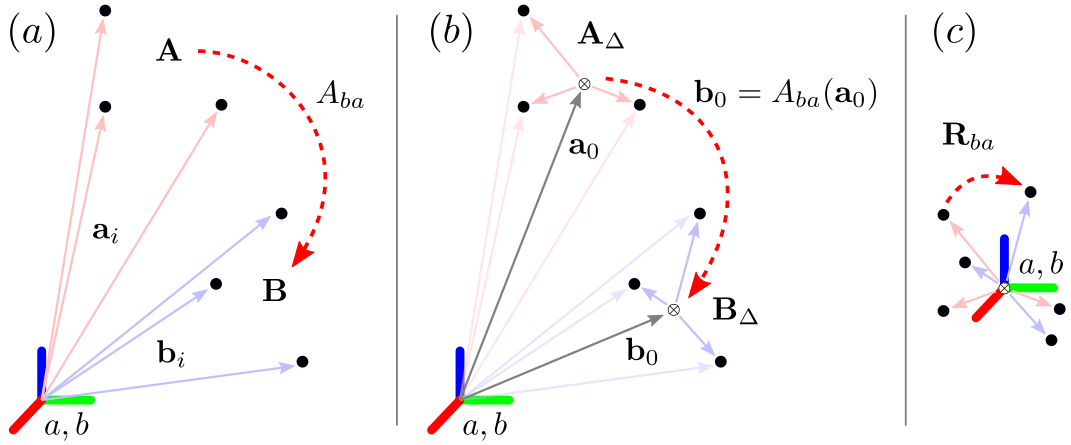

Figure 4.3: Orthogonal Procrustes Analysis – (a) The rigid transformation $A_{ba} \in SE(3)$ that minimizes the total square distance between points $A_{ba}(\mathbf{a}_i) \in \mathbb{R}^3$ and $\mathbf{b}_i \in \mathbb{R}^3$ is sought. (b) Rotation and translation are decoupled by introducing suitable rigid reference points $\mathbf{a}_0$ and $\mathbf{b}_0$ that satisfy $\mathbf{b}_0 = A_{ba}(\mathbf{a}_0)$. (c) Assuming optimal reference points, the optimal rotation $\mathbf{R}_{ba}$ is analytically determined.

The first step in formulating an optimization problem is to propose a cost function. For the task at hand it is convenient to collect all points into augmented matrices to compactly write all transformations in a single matrix equation

$$\boxed{A_{ba}(\mathbf{A}) \equiv \mathbf{R}_{ba}\mathbf{A} + \mathbf{T}_{ba} = \begin{bmatrix} A_{ba}(\mathbf{a}_1) & A_{ba}(\mathbf{a}_2) & \dots & A_{ba}(\mathbf{a}_N) \end{bmatrix}} \tag{4.25}$$

where $\mathbf{A}, \mathbf{B}, \mathbf{T}_{ba} \in \mathbb{R}^{3\times N}$ are

$$\boxed{\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_N \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_N \end{bmatrix}, \quad \mathbf{T}_{ba} = \begin{bmatrix} \mathbf{t}_{ba} & \mathbf{t}_{ba} & \dots & \mathbf{t}_{ba} \end{bmatrix}}. \tag{4.26}$$

Take the difference between all paired target points and transformed points

$$\mathbf{B} - A_{ba}(\mathbf{A}) = \begin{bmatrix} \mathbf{b}_1 - A_{ba}(\mathbf{a}_1) & \mathbf{b}_2 - A_{ba}(\mathbf{a}_2) & \dots & \mathbf{b}_N - A_{ba}(\mathbf{a}_N) \end{bmatrix}. \tag{4.27}$$

While an exact solution mapping $\mathbf{A}$ to $\mathbf{B}$ would be desirable such that

$$\mathbf{B} - A_{ba}(\mathbf{A}) = \mathbf{0} \tag{4.28}$$

there is no guarantee such a solution exists. Instead an approximate solution is defined as one that "minimizes the difference" between the vectors in a way consistent with an exact solution. Abstract vectorspaces do not come equipped with a way to compare how similar two vectors are. However for geometry, the norm of the difference between points i.e. the Euclidean distance is available and is an obvious albeit arbitrary means of comparison. For reasons of differentiability the square distance is preferred. It follows a reasonable way to compare N pairs of points is by the total square distance between the pairs, which is equivalent to the square distance on $\mathbb{R}^{3N}$. To compute this quantity for Eq. 4.27 define the real Frobenius inner product $\langle \cdot, \cdot \rangle_F : \mathbb{R}^{M\times N} \times \mathbb{R}^{M\times N} \to \mathbb{R}$ as (assuming an orthonormal basis)

$$\boxed{\langle \mathbf{X}, \mathbf{Y} \rangle_F \equiv \sum_i \sum_j X_{ij} Y_{ij}} \tag{4.29}$$

so the induced real Frobenius norm $\|\cdot\|_F : \mathbb{R}^{M\times N} \to \mathbb{R}$ is defined by

$$\boxed{\|\mathbf{X}\|_F^2 \equiv \langle \mathbf{X}, \mathbf{X} \rangle_F}. \tag{4.30}$$

Orthogonal Procrustes Analysis seeks the rigid transformation that minimizes the cost function of total square distance between the target and transformed points

$$\boxed{A_{ba} = \underset{A\in SE(3)}{\arg\min} \|\mathbf{B} - A(\mathbf{A})\|_F^2}. \tag{4.31}$$

Keep in mind this is just a different way to write

$$\boxed{A_{ba} = \underset{A \in SE(3)}{\arg\min} \sum_{i=1}^{n} \|\mathbf{b}_i - A(\mathbf{a}_i)\|^2} \tag{4.32}$$

and emphatically minimizes the total *square* distance, and not the total distance.

The cost function can be analytically minimized. A key insight is that rigid points transform with a rotation and translation per Eq. 4.12, while differences between rigid points transform with the rotation only

$$A(\mathbf{u}) - A(\mathbf{u}') = [\mathbf{R}\mathbf{u} + \mathbf{t}] - [\mathbf{R}\mathbf{u}' + \mathbf{t}] = \mathbf{R}[\mathbf{u} - \mathbf{u}']. \tag{4.33}$$

This observation is used to decouple the problem of finding the optimal rotation from the optimal translation vector by introducing two undetermined rigid reference points $\mathbf{a}_0 \in \mathbb{R}^3$ and $\mathbf{b}_0 \in \mathbb{R}^3$ (Fig. 4.3.b). Start by decomposing the augmented matrix $\mathbf{A}$ into the sum of augmented matrices $\mathbf{A}_\Delta, \mathbf{A}_0 \in \mathbb{R}^{3\times N}$

$$\boxed{\mathbf{A} = \mathbf{A}_\Delta + \mathbf{A}_0} \tag{4.34}$$

where the columns of $\mathbf{A}_\Delta$ are the vector differences of the points $\mathbf{a}_i$ minus $\mathbf{a}_0$

$$\boxed{\mathbf{A}_\Delta = \begin{bmatrix} \mathbf{a}_1 - \mathbf{a}_0 & \mathbf{a}_2 - \mathbf{a}_0 & \dots & \mathbf{a}_n - \mathbf{a}_0 \end{bmatrix}} \tag{4.35}$$

and the columns of $\mathbf{A}_0$ are N copies of reference point $\mathbf{a}_0$

$$\boxed{\mathbf{A}_0 = \begin{bmatrix} \mathbf{a}_0 & \mathbf{a}_0 & \dots & \mathbf{a}_0 \end{bmatrix}}. \tag{4.36}$$

The reference point $\mathbf{a}_0$ is undetermined besides transforming like a rigid point. The augmented matrix $\mathbf{B}$ is similarly decomposed with an undetermined rigid reference point $\mathbf{b}_0$

$$\boxed{\mathbf{B} = \mathbf{B}_\Delta + \mathbf{B}_0}. \tag{4.37}$$

The reference points act as rotation centers by constraining that $\mathbf{a}_0$ maps to $\mathbf{b}_0$

$$\mathbf{B}_0 = \mathbf{R}\mathbf{A}_0 + \mathbf{T}. \tag{4.38}$$

Translation vector $\mathbf{t}$ can then be written in terms of points $\mathbf{a}_0, \mathbf{b}_0$ and rotation $\mathbf{R}$

$$\boxed{\mathbf{T} = \mathbf{B}_0 - \mathbf{R}\mathbf{A}_0} \tag{4.39}$$

$$\mathbf{t}_{ba} = \mathbf{b}_0 - \mathbf{R}_{ba}\mathbf{a}_0 \tag{4.40}$$

and eliminated from the cost function by substituting Eq. 4.34, Eq. 4.37, and Eq. 4.39 into Eq. 4.31

$$\begin{aligned} A_{ba} &= \underset{A \in SE(3)}{\arg\min} \|\mathbf{B} - A(\mathbf{A})\|_F^2 \\ &= \underset{A \in SE(3)}{\arg\min} \|[\mathbf{B}_\Delta + \mathbf{B}_0] - [\mathbf{R}[\mathbf{A}_\Delta + \mathbf{A}_0] + \mathbf{T}]\|_F^2 \\ &= \underset{A \in SE(3)}{\arg\min} \|[\mathbf{B}_\Delta - \mathbf{R}\mathbf{A}_\Delta] + [\mathbf{B}_0 - \mathbf{R}\mathbf{A}_0] - \mathbf{T}]\|_F^2 \\ &= \underset{A \in SE(3)}{\arg\min} \|\mathbf{B}_\Delta - \mathbf{R}\mathbf{A}_\Delta\|_F^2 . \end{aligned} \tag{4.41}$$

This last equation has the same form as a simpler problem (Fig. 4.3.c) of finding the optimal rotation $\mathbf{R}_{ba}$ that maps the set of vectors $\mathbf{A}_\Delta$ to the paired set of vectors $\mathbf{B}_\Delta$

$$\mathbf{R}_{ba} = \underset{\mathbf{R} \in SO(3)}{\arg\min} \|\mathbf{B}_\Delta - \mathbf{R}\mathbf{A}_\Delta\|_F^2 . \tag{4.42}$$

Setting aside the issue of the optimal choice of reference points, this cost function can be analytically minimized for any $\mathbf{A}_\Delta$ and $\mathbf{B}_\Delta$ and the optimal rotation found. Later if optimal reference points are chosen, the rotation determined by Eq. 4.31 and Eq. 4.42 will coincide.

The rotation problem is solved by utilizing the Singular Value Decomposition (SVD) . The following identity will be helpful

$$\langle \mathbf{X}\mathbf{Y}, \mathbf{Z} \rangle_F = \langle \mathbf{Y}, \mathbf{X}^T\mathbf{Z} \rangle_F = \langle \mathbf{X}, \mathbf{Z}\mathbf{Y}^T \rangle_F \tag{4.43}$$

and is verified using index notation

$$\sum_i \sum_j \sum_k X_{ik} Y_{kj} Z_{ij} = \sum_k \sum_j Y_{kj} \sum_i X_{ik} Z_{ij} = \sum_i \sum_k X_{ik} \sum_j Z_{ij} Y_{kj}. \tag{4.44}$$

First temporarily relax the set of acceptable solutions of Eq. 4.42 from $SO(3)$ to $O(3)$, and expand the norm in terms of inner products

$$\begin{aligned}
\mathbf{Q}_{ba} &= \underset{\mathbf{Q}\in O(3)}{\arg\min} \left\|\mathbf{B}_\Delta - \mathbf{Q}\mathbf{A}_\Delta\right\|_F^2 \\
&= \underset{\mathbf{Q}\in O(3)}{\arg\min} \left\langle \mathbf{B}_\Delta - \mathbf{Q}\mathbf{A}_\Delta, \mathbf{B}_\Delta - \mathbf{Q}\mathbf{A}_\Delta \right\rangle_F \\
&= \underset{\mathbf{Q}\in O(3)}{\arg\min} [\left\langle \mathbf{B}_\Delta, \mathbf{B}_\Delta \right\rangle_F - \left\langle \mathbf{B}_\Delta, \mathbf{Q}\mathbf{A}_\Delta \right\rangle_F - \left\langle \mathbf{Q}\mathbf{A}_\Delta, \mathbf{B}_\Delta \right\rangle_F + \left\langle \mathbf{Q}\mathbf{A}_\Delta, \mathbf{Q}\mathbf{A}_\Delta \right\rangle_F] \\
&= \underset{\mathbf{Q}\in O(3)}{\arg\min} [\left\langle \mathbf{B}_\Delta, \mathbf{B}_\Delta \right\rangle_F - 2\left\langle \mathbf{B}_\Delta, \mathbf{Q}\mathbf{A}_\Delta \right\rangle_F + \left\langle \mathbf{A}_\Delta, \mathbf{A}_\Delta \right\rangle_F] \\
&= \underset{\mathbf{Q}\in O(3)}{\arg\max} \left\langle \mathbf{B}_\Delta, \mathbf{Q}\mathbf{A}_\Delta \right\rangle_F .
\end{aligned} \tag{4.45}$$

The last step recasts the minimization problem into a maximization problem by recognizing that only the negative term depends on $\mathbf{Q}$. This expression is manipulated using Eq. 4.43 and an SVD factorization

$$\begin{aligned}
\mathbf{Q}_{ba} &= \underset{\mathbf{Q}\in O(3)}{\arg\max} \left\langle \mathbf{B}_\Delta \mathbf{A}_\Delta^T, \mathbf{Q} \right\rangle_F \\
&= \underset{\mathbf{Q}\in O(3)}{\arg\max} \left\langle \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \mathbf{Q} \right\rangle_F \quad \text{(SVD factorization)} \\
&= \underset{\mathbf{Q}\in O(3)}{\arg\max} \left\langle \mathbf{\Sigma}, \mathbf{U}^T\mathbf{Q}\mathbf{V} \right\rangle_F
\end{aligned} \tag{4.46}$$

where matrix $\mathbf{B}_\Delta \mathbf{A}_\Delta^T$ is factored by SVD

$$\boxed{\mathbf{B}_\Delta \mathbf{A}_\Delta^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T}. \tag{4.47}$$

The SVD of a real matrix is composed of orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ and a diagonal matrix $\mathbf{\Sigma}$ of real non-negative singular values. $\mathbf{Q}$ is an orthogonal matrix, so the

product $\mathbf{U}^T\mathbf{Q}\mathbf{V}$ is also an orthogonal matrix. $\mathbf{\Sigma}$ is a diagonal matrix, so only the diagonal components of $\mathbf{U}^T\mathbf{Q}\mathbf{V}$ contribute to the inner product in Eq. 4.46. The components of an orthogonal matrix are constrained to the interval $[-1, 1] \subset \mathbb{R}$, and the identity matrix $\mathbf{1}$ is an orthogonal matrix with all ones on its diagonal. By inspection it must be that $\mathbf{U}^T\mathbf{Q}\mathbf{V} = \mathbf{1}$ so the optimal orthogonal matrix $\mathbf{Q}_{ba}$ is

$$\mathbf{Q}_{ba} = \mathbf{U}\mathbf{V}^T \in O(3). \tag{4.48}$$

Orthogonal matrices with $\det(\mathbf{Q}) = -1$ must be excluded when considering rigid transformations. The SVD is customarily organized such that the singular values of $\mathbf{\Sigma}$ monotonically decrease. Ensure a rotation matrix by reflecting across the dimension with the smallest penalty, the last singular value, if necessary

$$\boxed{\mathbf{R}_{ba} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{U}\mathbf{V}^T) \end{bmatrix} \mathbf{V}^T \in SO(3)}. \tag{4.49}$$

This gives an analytical solution to the optimal rotation $\mathbf{R}_{ba}$ that maps $\mathbf{A}_\Delta$ to $\mathbf{B}_\Delta$. However to determine the optimal rigid transformation A_{ba} that maps $\mathbf{A}$ to $\mathbf{B}$, it is still necessary to identity the reference vectors $\mathbf{a}_0$ and $\mathbf{b}_0$.

The 6 degrees of freedom of $\mathbf{R}$ and $\mathbf{t}$ in the cost function have effectively been rewritten in terms of $\mathbf{a}_0$ and $\mathbf{b}_0$. A condition on the reference points is derived by taking the gradients of the cost function with respect to the reference points and setting the results to the zero vector. Note that the components of $\mathbf{R}$ and $\mathbf{t}$ depend on the reference points but will have zero derivatives when optimized. Start by taking the gradient with respect to $\mathbf{b}_0$ of the rewritten cost function Eq. 4.42 and set the

result equal to the zero vector

$$\nabla_{\mathbf{b}_0} \sum_{i=1}^{N} \left\| [\mathbf{b}_i - \mathbf{b}_0] - \mathbf{R}[\mathbf{a}_i - \mathbf{a}_0] \right\|^2 = \mathbf{0}. \tag{4.50}$$

Evaluate the expression using index notation

$$\begin{aligned}
&\frac{\partial}{\partial b_{0l}} \sum_{i=1}^{N} \sum_{j=1}^{3} \left[[b_{ij} - b_{0j}] - \sum_{k=1}^{3} R_{jk}[a_{ik} - a_{0k}] \right]^2 \\
&= \sum_{i=1}^{N} \sum_{j=1}^{3} 2 \left[[b_{ij} - b_{0j}] - \sum_{k=1}^{3} R_{jk}[a_{ik} - a_{0k}] \right] \frac{\partial}{\partial b_{0l}} \left[[b_{ij} - b_{0j}] - \sum_{k=1}^{3} R_{jk}[a_{ik} - a_{0k}] \right] \\
&= \sum_{i=1}^{N} \sum_{j=1}^{3} 2 \left[[b_{ij} - b_{0j}] - \sum_{k=1}^{3} R_{jk}[a_{ik} - a_{0k}] \right] [-\delta_{lj}] \\
&= -2 \left[\left[\sum_{i=1}^{N} b_{il} - N b_{0l} \right] - \sum_{k=1}^{3} R_{lk} \left[\sum_{i=1}^{N} a_{ik} - N a_{0k} \right] \right] \\
&= -2N \left[[\langle b_{il} \rangle - b_{0l}] - \sum_{k=1}^{3} R_{lk}[\langle a_{ik} \rangle - a_{0k}] \right]
\end{aligned} \tag{4.51}$$

and convert the result to matrix notation

$$-2N[[\langle \mathbf{b}_i \rangle - \mathbf{b}_0] - \mathbf{R}[\langle \mathbf{a}_i \rangle - \mathbf{a}_0]] = \mathbf{0}. \tag{4.52}$$

Likewise, take the gradient with respect to $\mathbf{a}_0$ of the rewritten cost function Eq. 4.42 and set the result equal to the zero vector

$$\nabla_{\mathbf{a}_0} \sum_{i=1}^{N} \left\| [\mathbf{b}_i - \mathbf{b}_0] - \mathbf{R}[\mathbf{a}_i - \mathbf{a}_0] \right\|^2 = \mathbf{0}. \tag{4.53}$$

Once again evaluate the gradient in index notation

$$\begin{aligned}
&\frac{\partial}{\partial a_{0l}} \sum_{i=1}^{N} \sum_{j=1}^{3} \left[[b_{ij} - b_{0j}] - \sum_{k=1}^{3} R_{jk}[a_{ik} - a_{0k}] \right]^2 \\
&= \sum_{i=1}^{N} \sum_{j=1}^{3} 2 \left[[b_{ij} - b_{0j}] - \sum_{k=1}^{3} R_{jk}[a_{ik} - a_{0k}] \right] \frac{\partial}{\partial a_{0l}} \left[[b_{ij} - b_{0j}] - \sum_{k=1}^{3} R_{jk}[a_{ik} - a_{0k}] \right] \\
&= \sum_{i=1}^{N} \sum_{j=1}^{3} 2 \left[[b_{ij} - b_{0j}] - \sum_{k=1}^{3} R_{jk}[a_{ik} - a_{0k}] \right] [R_{jl}] \\
&= 2 \left[\sum_{j=1}^{3} R_{jl} \left[\sum_{i=1}^{N} b_{ij} - n b_{0j} \right] - \left[\sum_{i=1}^{N} a_{il} - N a_{0l} \right] \right] \\
&= 2n \left[\sum_{j=1}^{3} R_{jl} \left[\langle b_{ij} \rangle - b_{0j} \right] - \left[\langle a_{il} \rangle - a_{0l} \right] \right]
\end{aligned} \tag{4.54}$$

and convert the result to matrix notation

$$2N[\mathbf{R}^T[\langle \mathbf{b}_i \rangle - \mathbf{b}_0] - [\langle \mathbf{a}_i \rangle - \mathbf{a}_0]] = \mathbf{0}. \tag{4.55}$$

Both Eq. 4.52 and Eq. 4.55 result in the same condition

$$\langle \mathbf{b}_i \rangle - \mathbf{R} \langle \mathbf{a}_i \rangle = \mathbf{b}_0 - \mathbf{R}\mathbf{a}_0. \tag{4.56}$$

This equation is always satisfied if the average points are taken as the reference points

$$\boxed{\mathbf{a}_0 = \langle \mathbf{a}_i \rangle} \tag{4.57}$$

$$\boxed{\mathbf{b}_0 = \langle \mathbf{b}_i \rangle}. \tag{4.58}$$

In some cases other choices of reference points may satisfy Eq. 4.56. For instance, when an exact solution exists, any of the paired points $\mathbf{a}_i$ and $\mathbf{b}_i$ works. However the average points will always be optimal. With optimal reference points in hand, the optimal rotation $\mathbf{R}_{ba}$ and translation vector $\mathbf{t}_{ba}$ are determined, and the optimal rigid transformation A_{ba} is found.

In summary, the optimal rigid transformation A_{ba} mapping points $\mathbf{A}$ to $\mathbf{B}$ is computed by taking the average points $\langle \mathbf{a}_i \rangle$ and $\langle \mathbf{b}_i \rangle$ as reference points $\mathbf{a}_0$ and $\mathbf{b}_0$ (Eq. 4.57 and 4.58), then computing the SVD of the matrix $\mathbf{B}_\Delta \mathbf{A}_\Delta^T$ (Eq. 4.47), then computing the optimal rotation matrix $\mathbf{R}_{ba}$ (Eq. 4.49), and then finally computing the optimal translation vector $\mathbf{t}_{ba}$ (Eq. 4.39). This result is recorded in Algorithm 6. The boxed equations are encapsulated into a function to hide the internal complexity of this useful result

$$\boxed{A_{ba} = \text{Procrustes}(\mathbf{A}, \mathbf{B})}. \tag{4.59}$$

Algorithm 6: Orthogonal Procrustes Analysis

input : $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{3\times N}$

output: $A_{ba} \in SE(3)$

$\mathbf{a}_0 = \langle \mathbf{a}_i \rangle$

$\mathbf{b}_0 = \langle \mathbf{b}_i \rangle$

$\mathbf{A}_\Delta = \mathbf{A} - \mathbf{A}_0$

$\mathbf{B}_\Delta = \mathbf{B} - \mathbf{B}_0$

$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{svd}(\mathbf{B}_\Delta \mathbf{A}_\Delta^T)$

$\mathbf{R}_{ba} = \mathbf{U}\ \text{diag}(1, 1, \det(\mathbf{U}\mathbf{V}^T))\ \mathbf{V}^T$

$\mathbf{t}_{ba} = \mathbf{b}_0 - \mathbf{R}_{ba}\mathbf{a}_0$

$A_{ba}(\mathbf{r}) = \mathbf{R}_{ba}\mathbf{r} + \mathbf{t}_{ba}$

The order of the columns in the augmented matrices $\mathbf{A}$ and $\mathbf{B}$ is significant. If the points are unordered, a brute force method to determine the optimal order is to consider all permutations of the columns of $\mathbf{A}$

$$\mathbf{A}_\sigma = \begin{bmatrix} \mathbf{a}_{\sigma(1)} & \mathbf{a}_{\sigma(2)} & \dots & \mathbf{a}_{\sigma(N)} \end{bmatrix} \quad \text{where} \quad \sigma \in S_N \tag{4.60}$$

and select the permutation that minimizes the cost function

$$\boxed{A_{ba} = \min_{\sigma \in S_N} \left[\arg\min_{A \in SE(3)} \|\mathbf{B} - A(\mathbf{A}_\sigma)\|_F^2 \right]}. \tag{4.61}$$

$$A_{ba} = \min_{\sigma \in S_N} \text{Procrustes}(\mathbf{A}_\sigma, \mathbf{B}) \tag{4.62}$$

The time complexity of searching permutations is $O(N!)$, so this method is only suitable for about $6 \leq N$. Bundle adjustment is the appropriate technique when dealing with larger numbers of unordered points, but requires more mathematical apparatus. The unordered version of the Procrustes function is denoted

$$\boxed{A_{ba} = \text{Procrustes2}(\mathbf{A}, \mathbf{B})}. \tag{4.63}$$

For Orthogonal Procrustes Analysis to guarantee a unique solution, some conditions on the points must be satisfied. The **affine span** of a set of vectors V is the span of the set $V' = \{\mathbf{v}' = \mathbf{v} - \mathbf{v}_0 | \mathbf{v} \in V - \{\mathbf{v}_0\}\}$ for some reference vector $\mathbf{v}_0 \in V$. A unique solution is guaranteed if V_a and V_b affinely span $\mathbb{R}^3$. These conditions ensure that the matrix $\mathbf{B}_\Delta \mathbf{A}_\Delta^T$ is full rank so a unique rotation is defined. Additionally when considering unordered points there must not be any rotational symmetry, or the wrong permutation could be selected.

OPA can be extended to include a scaling factor [Roh05]. In this work all position data is assumed to be expressed in meters, so the scaling factor is always 1.

4.3 Constellation Registration

Consider the problem of experimentally registering a rigid body with a positioning system that measures the positions of point-like targets. One idea is to affix a set

of targets in a known pattern to the rigid body, and then estimate the rigid transformation between the known pattern data and measured positioning system data (Fig. 4.4). This idea is the cornerstone of constellation registration.

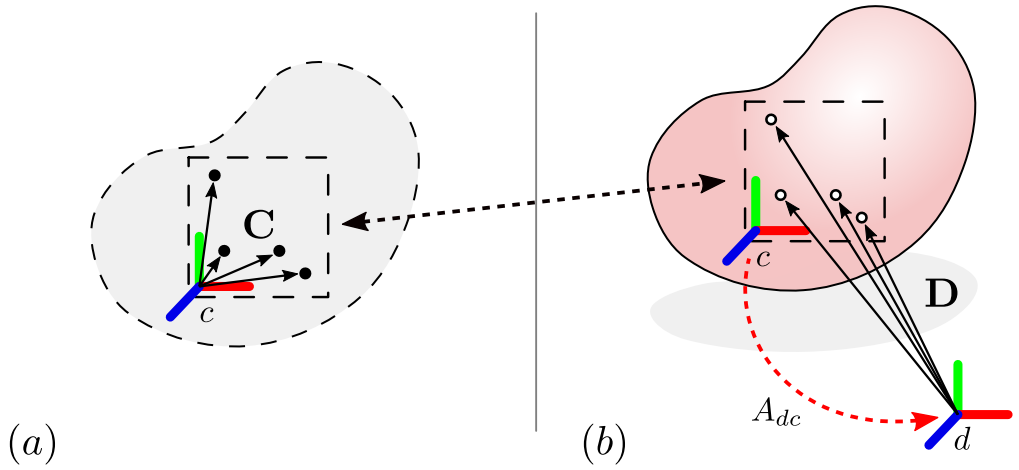

Figure 4.4: Constellation Registration

Assume the positioning system sensor is sensitive to small point-like targets called **fiducials**. The sensor measures and reports fiducial position data relative to a sensor-defined basis d. This basis is assumed to be right-handed, orthonormal, and have coordinates in units of meters. The position data is represented as $N_d \geq 0 \in \mathbb{Z}$ coordinate vectors organized into an augmented matrix $\mathbf{D} \in \mathbb{R}^{3\times N_d}$ in no particular order. Raw position data may not satisfy these assumptions, but it is always possible to achieve them with a small amount of extra processing. Measurement data will necessarily contain noise and have limited range and resolution.

A **constellation** is a set of rigid points arranged in a recognizable pattern for use in registration. This concept bridges three different realizations: a nominal,

physical, and measured constellation. The nominal constellation (Fig. 4.4.a) is the known position data relative to a constellation-defined basis c. Like basis d, this basis is assumed to be right-handed, orthonormal, and have coordinates in units of meters. The position data is represented by $N_c \geq 3 \in \mathbb{Z}$ coordinate vectors organized into an augmented matrix $\mathbf{C} \in \mathbb{R}^{3\times N_c}$. Points in $\mathbf{C}$ must affinely span $\mathbb{R}^3$ and be rotationally asymmetric to guarantee unique registration. The physical constellation (Fig. 4.4.b) is the set of fiducials that physically realizes the nominal constellation. These fiducials are affixed to the rigid body being registered. Their positions are similar to the nominal constellation up to a rigid transformation to within manufacturing tolerances. The measured constellation $\mathbf{D}$ (Fig. 4.4.b) is the sensor data of the physical constellation, which must necessarily have $N_d = N_c$ points.

Suppose a nominal constellation $\mathbf{C}$ is physically realized with fiducials and affixed to a rigid body. A positioning system sensor measures the physical constellation and reports the measured constellation position data $\mathbf{D}$. The optimal rigid transformation $A_{dc} \in SE(3)$ that maps constellation $\mathbf{C}$ to sensor data $\mathbf{D}$ is then

$$\boxed{A_{dc} = \text{Procrustes2}(\mathbf{C}, \mathbf{D})}. \tag{4.64}$$

Equivalently A_{dc} is the change of basis from c to d. Therefore A_{dc} fully represents the registration information of the rigid body with respect to the sensor. Take a moment to consider this result. Three different realizations of the same constellation are identified to experimentally determine the optimal rigid transformation from the rigid body basis to the sensor basis. The mathematical correspondence between the bases is established without either basis physically existing.

The nominal constellation can either be designed or measured. Designing the constellation is useful for mass production and high-precision registration, while mea-

suring the constellation is useful for rapid experimentation in the lab. In either case, constellation points should be as far apart as possible to geometrically reduce the effects of measurement noise. It can be helpful to make the constellation basis c physically accessible by choosing points in $\mathbf{C}$ to coincide with the origin and basis vectors. This makes it easier to understand and debug registration results.

Often it is convenient to define a global basis g that differs from the positioning system sensor basis d. This is particularly useful when registering multiple rigid-bodies. Suppose $m \geq 2$ rigid-bodies indexed by i are registered with constellations $\mathbf{C}_i$ against sensor data $\mathbf{D}_i$ resulting in rigid registration $A_{dc,i}$ respectively

$$A_{dc,i} = \text{Procrustes2}(\mathbf{C}_i, \mathbf{D}_i). \tag{4.65}$$

Select the j-th constellation basis to serve as the global basis g. The change of basis from sensor basis d to global basis g is then simply the inverse of the j-th registration transformation, that is $A_{gd} \equiv A_{dc,j}^{-1}$. Eliminate references to sensor basis d by composing A_{gd} with the other registration transformations

$$\boxed{A_{gc,i} = A_{gd} \circ A_{dc,i}}. \tag{4.66}$$

In this way the constellation basis of any registered rigid body can serve as the global basis. Moreover all physical and mathematical traces of the positioning system can be removed from the imaging system once registration is complete, which simplifies implementation.

4.4 Antenna Registration

The primary imager registration task is antenna registration. The imaging forward model (Ch. 2) depends on the $\mathbf{E}$ fields of the transmit and receive antennas in a

given imaging volume

$$H_{ij} = \mathbf{E}_{rx,i}(\mathbf{r}_j, \omega_i) \cdot \mathbf{E}_{tx,i}(\mathbf{r}_j, \omega_i) \Delta V_j. \tag{4.67}$$

Obviously the fields will depend on the position and orientation of the antennas relative to the imaging volume. Therefore the antennas must be registered with the imaging volume. Furthermore, the antenna fields are computed from near field scan (NFS) measurements. The spatial relationship between an antenna and its intangible NFS has not been established. This becomes an issue when relocating an antenna from the scanning environment, as only the antenna can be observed and registered at a later time. Therefore a NFS must be registered with its antenna. By registering NFSs with their antennas, and registering antennas with an imaging volume, the antenna fields within the imaging volume can be modeled from the NFSs.

This section develops antenna registration in two major parts. First, NFS registration is discussed where a NFS is registered to its antenna using an RF constellation. Second, array registration is described where all antennas in an array are simultaneously registered using photogrammetry constellations. A summary of the full antenna registration model then follows.

4.4.1 NFS Registration

A near field scan (NFS) measures the fields radiated by an antenna on an invisible 2D surface (Fig. 4.5), which permits modeling the fields radiated anywhere. A NFS must be registered with its antenna if the antenna is to be moved from the scanning environment. This section describes how to use sub-wavelength radiating elements on the antenna as **RF fiducials**. These are located by using the NFS to model the fields in a volume containing the antenna and searching for points of high field

intensity in the expected places. The 3D field data is processed to precisely estimate the location of the RF fiducials. Constellation registration is then applied to register the NFS with its antenna.

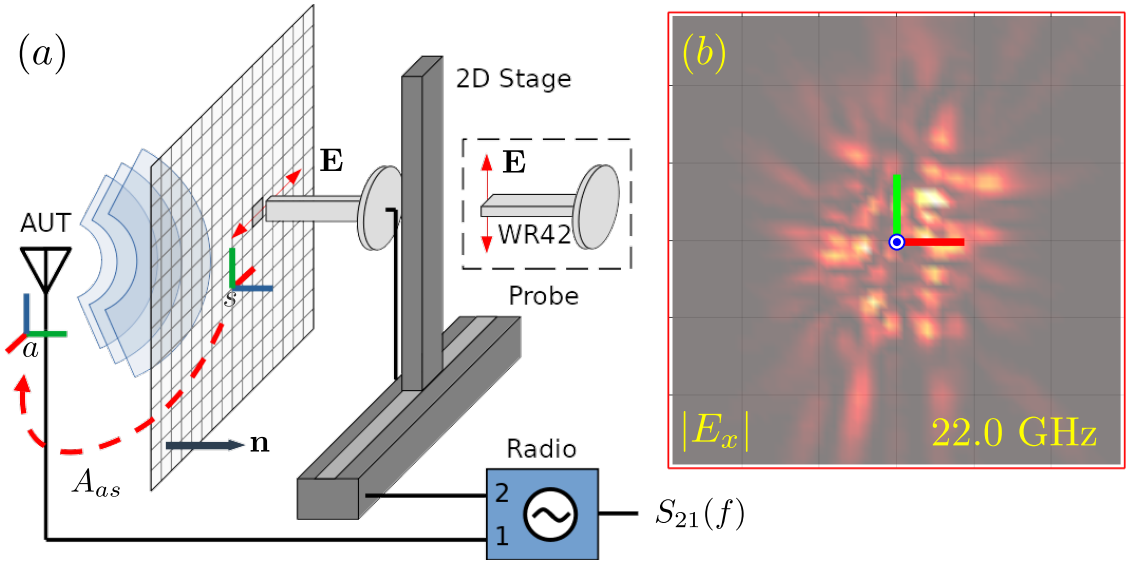

Figure 4.5: Antenna Near Field Scan

The Ch. 2 antenna model is not suitable for computing the fields in the half-space bounded by the scan plane containing the antenna. The antenna model invokes equivalence principles and image theory that trades information in the antenna half-space for mathematical simplicity on the boundary plane; the fields are mirrored across the plane, but only magnetic surface current densities radiating in freespace exist on the plane. Modeling the fields in the vicinity of the antenna requires a different approach.

The **Angular Spectrum Method (ASM)** relates coplanar slices of a field composed of plane waves restricted to k-vectors with the same magnitude that lie on one side of the slice plane (Fig.4.6). This sounds complicated, but this is the exact situ-

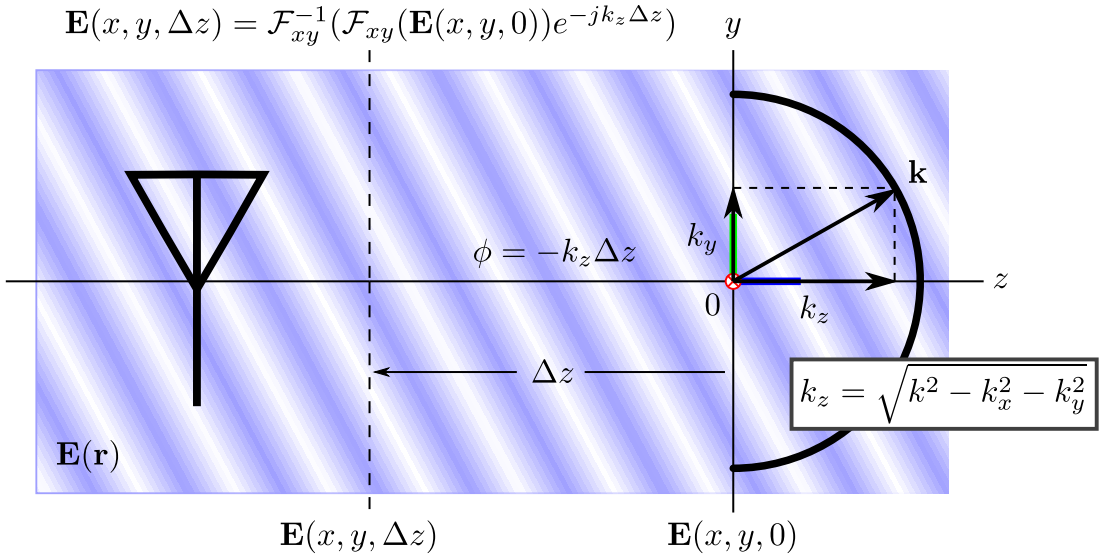

Figure 4.6: Angular Spectrum Method

ation for an antenna being driven at a single frequency f and radiating in freespace through a NFS plane and out to infinity (assuming no reflections!). Practically speaking, given a NFS, the fields can be backpropagated with the ASM to the volume surrounding an antenna. Consider the electric field in rectangular coordinates

$$\mathbf{E}(x,y,z) = \hat{\mathbf{x}}E_x(x,y,z) + \hat{\mathbf{y}}E_y(x,y,z) + \hat{\mathbf{z}}E_z(x,y,z). \tag{4.68}$$

Given the electric field $\mathbf{E}(x,y,0)$ on the $z=0$ plane, the field at $\mathbf{E}(x,y,\Delta z)$ is sought. Start by taking the 2D Fourier transform of $\mathbf{E}$ in the xy-plane at $z=0$

$$\mathbf{E}(k_x,k_y,0) = \mathcal{F}_{xy}(\mathbf{E}(x,y,0)). \tag{4.69}$$

For any wavevector in the plane, k_z is determined by the magnitude of k

$$k_z = (k^2 - k_x^2 - k_y^2)^{\frac{1}{2}}. \tag{4.70}$$

The requirement that the field only have k-vectors on one side of the plane ensures k_z is uniquely determined. A key insight is that the Fourier components of coplanar

field slices separated by Δz are the same up to a phase change of $\phi = -k_z \Delta z$

$$\mathbf{E}(k_x, k_y, \Delta z) = \mathbf{E}(k_x, k_y, 0) e^{-jk_z \Delta z}. \tag{4.71}$$

This result is most easily understood in terms of a single plane wave, as illustrated in Fig. 4.6. The plane wave magnitude is the same at any two slices, but the phases are different in a way that depends on the direction of the wave and distance between the slices. The inverse Fourier transform is applied to yield the field at the slice $z = \Delta z$.

$$\mathbf{E}(x, y, \Delta z) = \mathcal{F}_{xy}^{-1}(\mathcal{F}_{xy}(\mathbf{E}(x, y, 0)) e^{-jk_z \Delta z}) \tag{4.72}$$

Implementing the ASM on computer takes some care, as Fourier transforms must be appropriately mapped to discrete Fourier transforms for numeric computation. Assume the NFS is finely enough sampled to satisfy the sampling theorem, and large enough to capture the high-magnitude parts of the field. First, field samples should be modeled as being at the center of a 2D rectangular "pixel". Second, the NFS margins should be padded with zeros to mitigate aliasing. Third, k-space support in the xy-plane should be limited to radius k

$$k_x^2 + k_y^2 \leq k^2 \tag{4.73}$$

with components that exceed this limit zeroed out. To relate indices to continuous quantities, the following relationship is helpful

$$\Delta k_x = \frac{2\pi}{N_x \Delta x} \tag{4.74}$$

where N_x is the number of sample points and Δx the step size along the x-axis. A similar equation applies along the y-axis. Typically $\Delta x = \Delta y$ for NFSs.

NFS registration is now described. An antenna equipped with a RF constellation is measured by NFS at several frequencies. The NFS is used to compute the fields

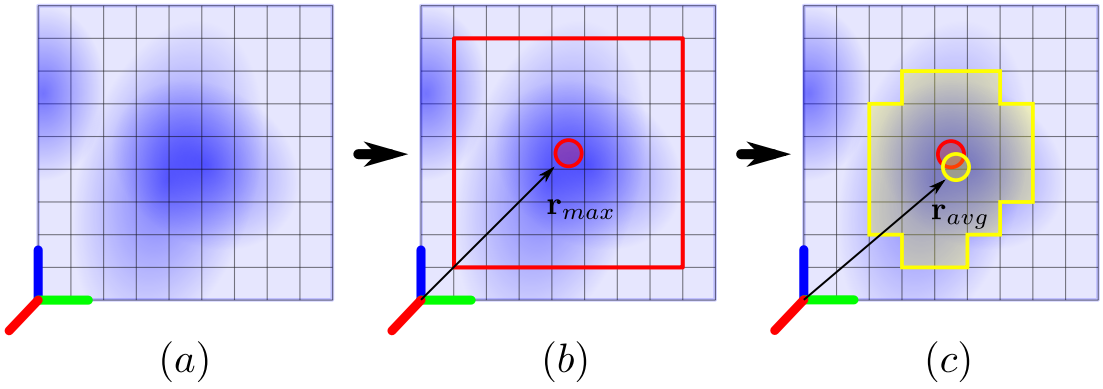

Figure 4.7: RF Fiducial Position Estimation

in the volume containing the antenna by ASM. ASM only considers one frequency at a time, so for multiple frequencies only the maximum field magnitude across all frequencies is recorded for each voxel. The result is upsampled. The RF fiducial positions are then ready to be estimated (Fig. 4.7.a). A nominal antenna constellation $\mathbf{C}_a$ is specified and a guess is made for the registration. Small search volumes are constructed where fiducials are expected to be found (Fig. 4.7.b). For each search volume the maximum field value and position of that value are found. Then 3D flood fills are computed starting at each maximum position and given a lower threshold that is some fraction of the maximum (i.e. 0.5) (Fig. 4.7.c). The fills result in masks of connected regions of high intensity. To isolate the point-like RF fiducials, a weighted average over each fill is taken

$$\mathbf{r}_{avg} = \frac{1}{M} \int_{V_{fill}} \|\mathbf{E}\|^2 \, \mathbf{r} dV \quad \text{where} \quad M = \int_{V_{fill}} \|\mathbf{E}\|^2 \, dV. \tag{4.75}$$

This yields measured constellation $\mathbf{D}_s$. Finally constellation registration is applied

to register the NFS basis s with the antenna basis a to get rigid transformation A_{sa}

$$A_{sa} = \text{Procrustes2}(\mathbf{C}_a, \mathbf{D}_s). \tag{4.76}$$

NFS registration is shown in Fig. 4.8.

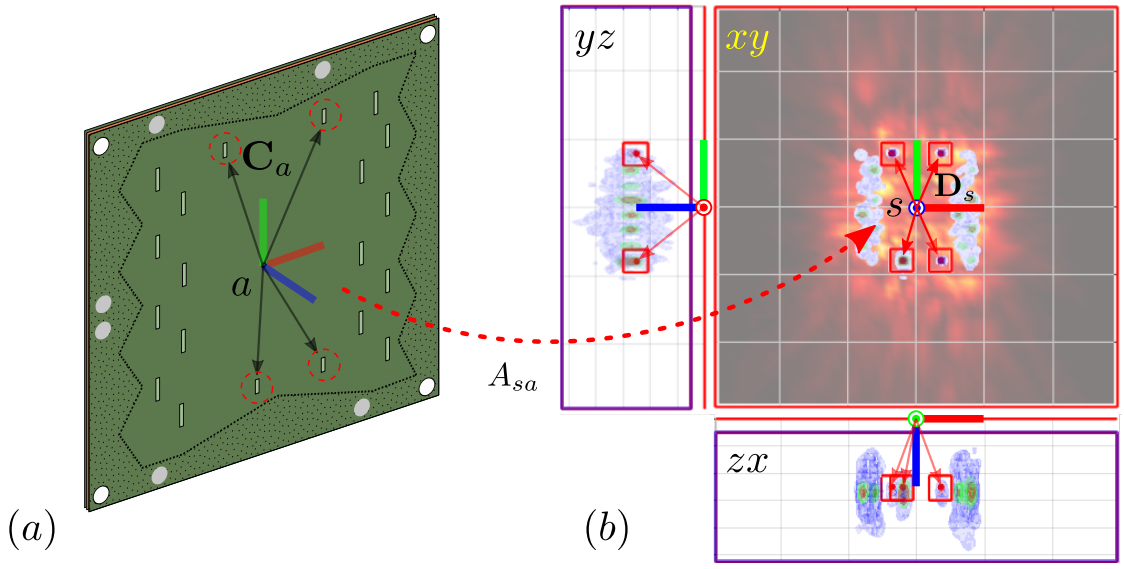

Figure 4.8: NFS Registration

4.4.2 Array Registration

An array of antennas are assembled to form an aperture. Per the forward model, the antennas must be registered with an imaging volume to model their fields within the imaging volume. This section describes an array registration strategy that works by processing photogrammetry data and applying constellation registration.

Registration error analysis indicates antenna centers should be located to within 1.0 mm accuracy over the extent of a 2.0 m × 2.0 m aperture to minimize image degradation (4.9) [OIL+16]. Few registration options are available that satisfy this

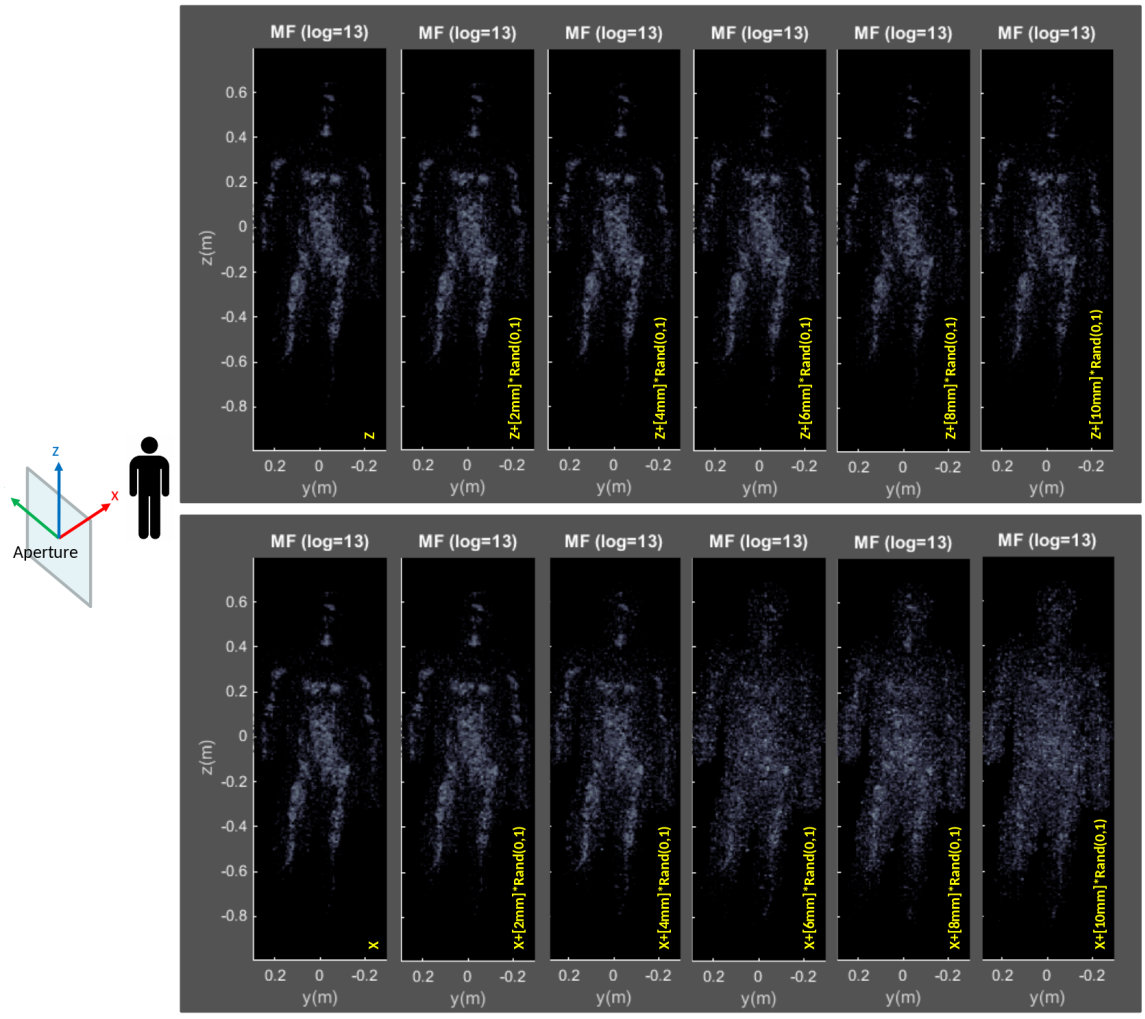

Figure 4.9: Array Registration Error — Simulated registration error. Cross-range registration error weakly affects image quality (top) while range registration error strongly affects image quality (bottom) due to the phase of antenna fields changing most rapidly in the direction of propagation.

requirement. One option is to design the array registration and manufacture an array with precision machining. However this lends itself to high per-unit cost and is not a robust solution. Another option is to build an array and then measure the array registration with photogrammetry. Photogrammetry is the art of using photography to estimate geometry. The microscopic wavelength of visible light and high resolution of digital camera sensors makes highly accurate and precise 3D position measurements on everyday scales possible and economical. Conveniently, commercial photogrammetry systems are available that meet the requirements of the imager.

Figure 4.10: Creaform MaxSHOT 3D

A Creaform MaxSHOT 3D (Fig. 4.10) photogrammetry system was purchased

with a specified volumetric accuracy of 0.025 mm/m [2]. The MaxSHOT 3D measures the 3D positions of small circular reflective fiducials using a set of photos taken with a camera. These fiducials are used in several ways as illustrated in Fig. 4.11. There are two types of fiducials; uncoded fiducials and coded fiducials. Uncoded fiducials are stickers representing a single fiducial. Coded fiducials are cards with a unique constellation and numeric identifier. Coded fiducials are also found on three registration props that determine the photogrammetry coordinate system origin and calibrate measurement scale.

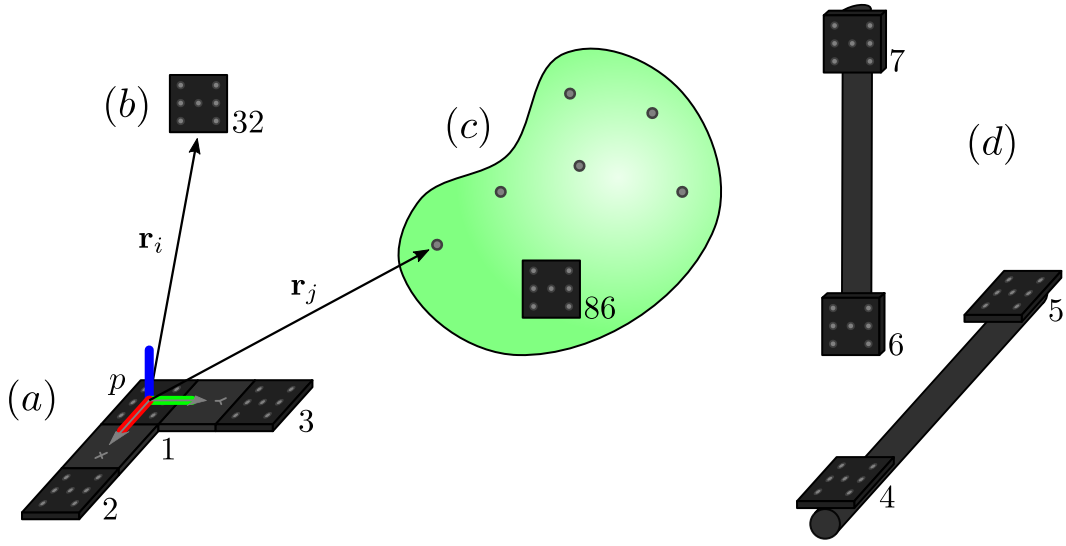

Figure 4.11: Creaform MaxSHOT 3D Fiducials

Operation of the MaxSHOT 3D is a multistep process. First, uncoded and coded fiducials are affixed to a scene. Coded fiducials should be scattered throughout the scene to help establish correspondences between photos in post-processing. Next, the registration props are placed in the scene to set the origin and calibrate scale.

[2]https://www.creaform3d.com/en/maxshot-3d-g1-scanner

Then a set of about 100 photos of the scene is captured from a variety of positions and orientations. It is important that the scene is not changed during this process. For each photo, the camera illuminates the scene with a flash that the fiducials are designed to strongly retroreflect and that the camera filter is designed to selectively pass, resulting in a high-contrast photo of bright fiducials on a dark background. This makes detecting fiducials and estimating their 2D positions easy. Finally, a computer finds correspondences between photos and applies bundle adjustment [TMHF99] to the 2D position data to simultaneously solve for all fiducial 3D positions and camera poses relative to the origin. The results are recorded in a text file. The entire process takes about 15 minutes in the lab assuming uncoded fiducials have already been placed.

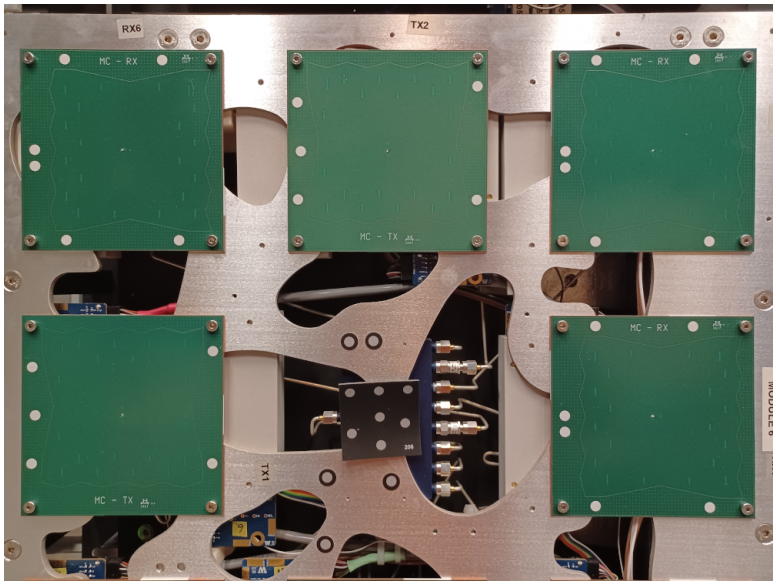

Figure 4.12: Array Photogrammetry Fiducials

Array registration using the photogrammetry system is now considered. A constellation registration strategy was adopted using constellations of uncoded photogram-

metry fiducials (Fig. 4.12). The need to register multiple antennas at once introduces the need to segment photogrammetry data into multiple constellations and the need to identify constellations from a library of multiple known constellations.

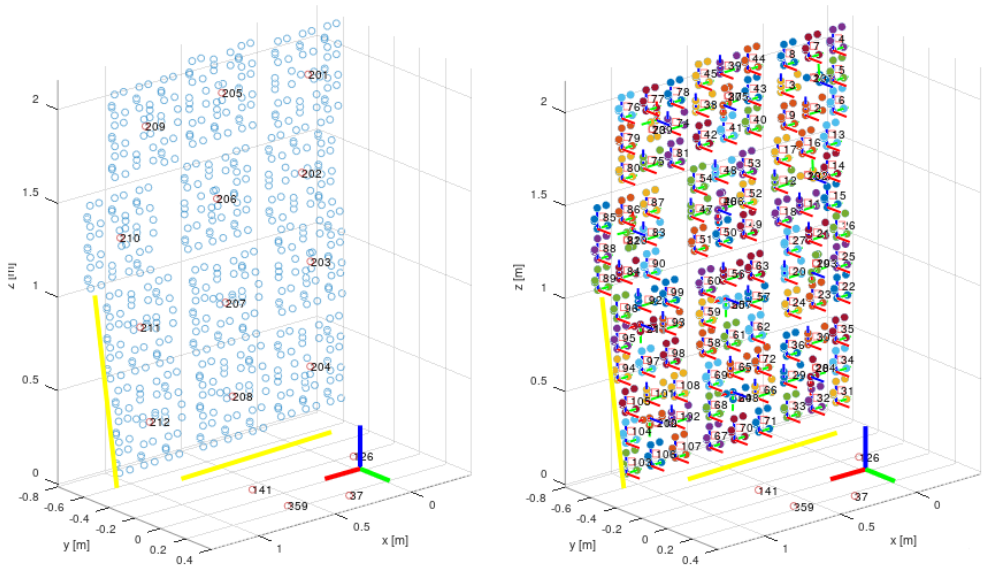

Figure 4.13: Array Photogrammetry Data

Segmenting photogrammetry data into constellations is the first task (Fig. 4.13). The constellations are seen to be in close proximity to each other. Let $D_p \subset \mathbb{R}^3$ be a set of fiducials positions measured by photogrammetry. A basic but reliable segmentation strategy is to group fiducials into a constellation if they are within a ball. The i-th coded fiducial can be used to define ball center point $\mathbf{r}_i$, and the ball radius R_i can be specified. All uncoded fiducials within the ball are elements of set $D_{p,i}$ representing a measured constellation

$$D_{p,i} = \left\{ \mathbf{r} \in D_p \middle| \, \|\mathbf{r} - \mathbf{r}_i\|^2 \leq R_i^2 \right\}. \tag{4.77}$$

Measured constellation points are packed into augmented matrix $\mathbf{D}_{p,i}$ for analysis.

Identifying each measured constellation from a library is the second task. A brute force strategy is employed. Consider a set of constellations L (the library). For each measured constellation, constellation registration is applied for each library constellation, and the library constellation that results in the smallest error is defined to be the match

$$A_{pl,i} = \min_{\mathbf{C}_l \in L} \text{Procrustes2}(\mathbf{C}_l, \mathbf{D}_{p,i}). \tag{4.78}$$

Constellations must have the same number of fiducials to match, and matches with high registration error print warnings. The library basis l can represent many things. In the case of antennas, relabeling it a is appropriate, so the rigid transform is $A_{pa,i}$.

The above ideas work well to register a small array, but using coded fiducials to define grouping centers does not scale. To alleviate this issue, library entries contain a list of child grouping center points and radii. In this way a single coded fiducial can be used to register a parent constellation, and then locate the grouping regions of several child constellations relative to it. The experimental imaging system uses this feature to simplify registering antenna modules. Each module is given a constellation which is found with a coded fiducial, and the child antenna constellations are found using library information.

Originally it was intended that every antenna have a unique constellation. This would have allowed an antenna to be identified by its constellation, and the constellation could be given a name associated with the NFS file for that antenna. A miscommunication resulted in all transmit antennas sharing the same constellation, and all receive antennas sharing the same constellation. To associate each antenna with its unique NFS file, the option for a parent to override its child constellation names was introduced.

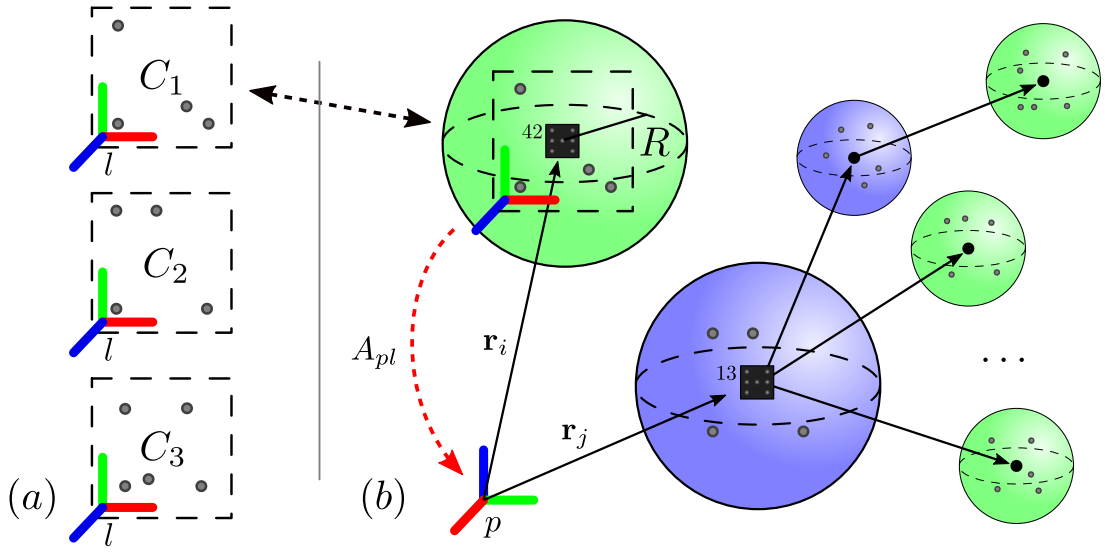

Figure 4.14: Array Registration

With these considerations, the complete array registration algorithm is now described in detail. Assume library data, photogrammetry data, and a list of coded fiducial numbers and grouping radii are given. The algorithm works using a stack of groups to be processed. The stack is initialized by pushing the specified coded fiducial positions and grouping radii. While the stack is not empty, the algorithm loops. The loop starts by pulling the next group off the stack. Next, fiducials are grouped into a constellation based on the grouping center point and grouping radius. Then, the constellation is analyzed using constellation registration against all library entries with the same number of constellation points. The library entry that results in the lowest error is defined to be the match. At this point, if a name override for the group exists, it replaces the matched name. If no match has been made, then a warning is printed and no registration information is recorded. Otherwise the registration information for the group is recorded. If registration error is high, a warning

is printed. A recursion guard asserts the group isn't it's own child. Finally, any child groups are transformed using the newly computed registration and pushed onto the stack. The algorithm outputs group information (matching library index, rigid transformation A_{pl}, registration error, and name) in the order it was processed. The result is visualized in Fig. 4.13.

4.4.3 Summary

Antenna registration for an antenna equipped with both a RF constellation and photogrammetry constellation is summarized in Fig. 4.15. First, the antenna is mea-

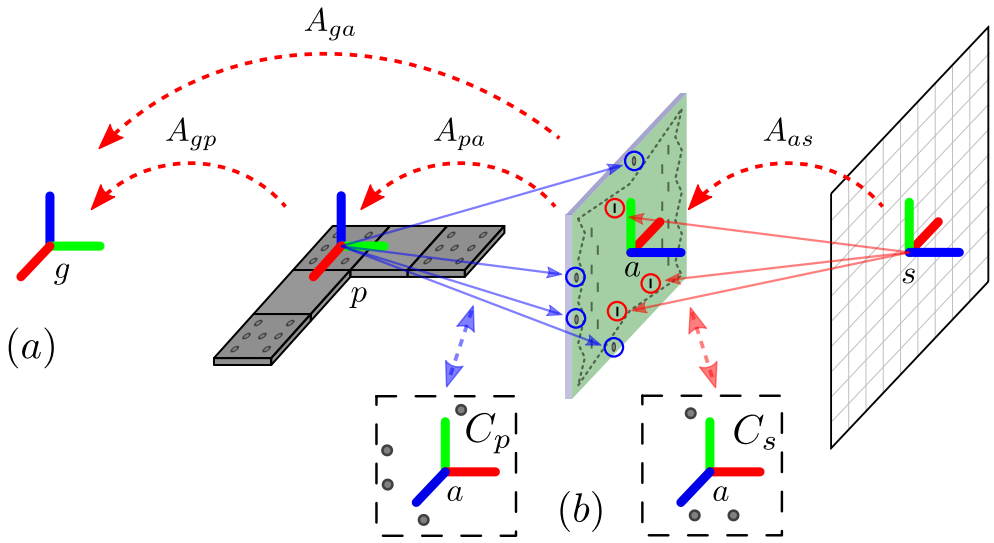

Figure 4.15: Imager Registration

sured in NFS basis s, which is registered to antenna basis a with the RF constellation built into the antenna, resulting in rigid transformation A_{as}. Next, the antenna is installed as the i-th antenna in an array that forms an aperture, and the aperture is measured with photogrammetry relative to basis p. The photogrammetry data is

segmented into constellations. The antenna basis a is registered to photogrammetry basis p with the photogrammetry constellation built into the antenna, resulting in rigid transformation $A_{pa,i}$. Antenna registration relative to photogrammetry basis p is fully described by

$$A_{ps,i} = A_{pa,i} \circ A_{as,i}. \tag{4.79}$$

It is convenient to eliminate references to photogrammetry basis p by specifying some global basis g that has greater significance to the imager using rigid transformation A_{gp}

$$A_{ga,i} = A_{gp} \circ A_{pa,i}. \tag{4.80}$$

Any choice of g works. In the lab an antenna roughly in the center of the aperture was chosen to serve as global basis g; in this case $A_{gp} = A_{pa,j}^{-1}$ for the j-th antenna basis.

Putting this together, the complete registration equation for the i-th antenna is

$$\boxed{A_{gs,i} = A_{gp} \circ A_{pa,i} \circ A_{as,i}}. \tag{4.81}$$

which maps scan basis s to antenna basis a to photogrammetry basis p to global basis g.

Chapter 5

Region of Interest

The previous chapters describe a functional RF imaging system that can measure people walking through a security screening volume in real-time and reconstruct detailed images of that volume in minutes using a workstation. The image reconstruction times as they stand are too long for security screening applications, but short enough to encourage refining the underlying techniques in pursuit of a solution. This chapter describes how to reduce image reconstruction times by introducing a **region of interest (ROI)** that restricts image volumes to only foreground objects, making the image reconstruction problem smaller and thus faster to solve. Depth cameras are integrated into the RF imaging system to realize this idea.

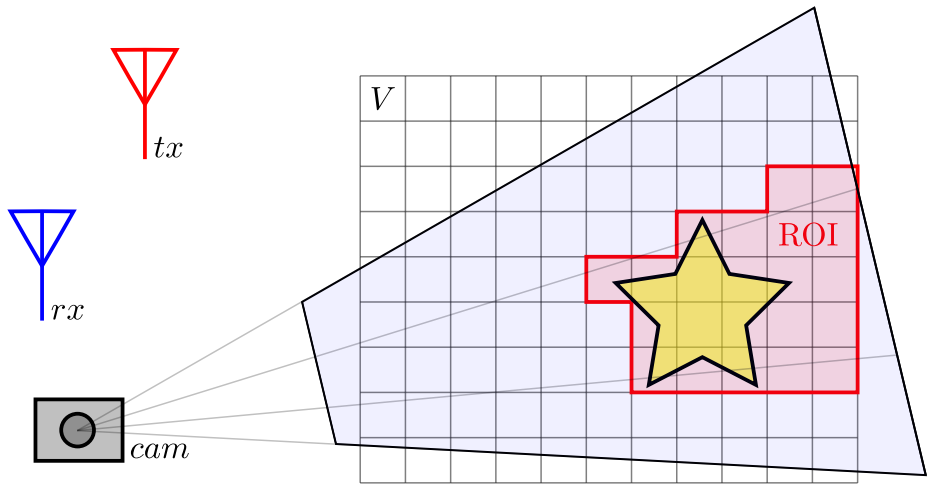

Figure 5.1: ROI Concept

Image reconstruction time should depend on the number of voxels N in the image, where more voxels correspond to longer reconstruction times. This is true for the linear imaging model described in Ch. 2 and Ch. 3. An obvious idea to reduce image reconstruction time is to reduce the number of voxels in the reconstructed image. Voxel volume is set by the imager configuration, so the total image volume must be reduced to reduce N. It becomes natural to ask what exactly determines the total volume of an image?

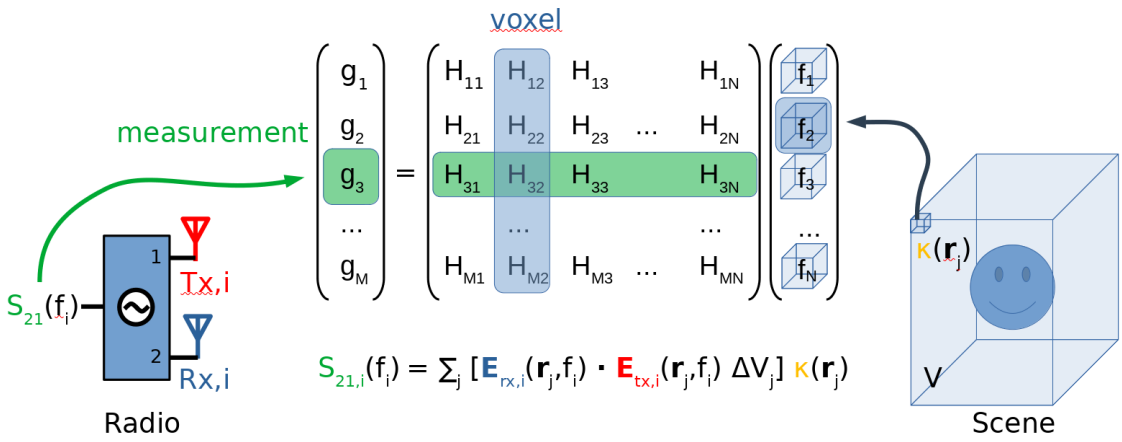

Figure 5.2: g = H f

A key observation about the forward model is only voxels in the RF foreground contribute to the measurement. Voxels not in the RF foreground (i.e. part of the RF background or air) will have components equal to zero in the scene vector **f**. These voxels and their corresponding columns in measurement matrix **H** can be eliminated without affecting measurement vector **g**. Thus an image volume must include at minimum all voxels occupied by objects in the RF foreground.

There is tension between wanting small total image volume and wanting the ability to form images anywhere within a large volume. The imaging forward model

is generally much more computationally expensive to build than it is to apply, and it is computationally infeasible to build the measurement matrix on the fly. Therefore it is necessary to build **H** before imaging. The location of foreground objects is not known before imaging, so the forward model is built over a larger volume than a single image is expected to occupy. It is desirable to compute the model over a large volume to maximize the value of the imaging hardware. Unfortunately a large imaging volume results in a large forward model and slow reconstruction time.

An economical solution is to use depth cameras to identify voxels in the RF foreground to reduce the size of a pre-computed forward model. Images are reconstructed over the ROI only. This is an efficient form of L0 regularization that constrains the number of non-zero voxels with a prior. As long as the number of measurements is comparable or more than the number of voxels in the ROI, image quality should be good.

This chapter develops the ROI concept in four sections. The first section describes the depth camera hardware and depth image. The second section develops a method to register depth cameras with the RF imaging system. The third section describes how to use registered depth cameras to generate a ROI. The fourth section describes how to modify the image reconstruction algorithms to use the ROI.

Additional avenues for reducing image reconstruction times exist. Accelerating the forward model is discussed in Ch. 2. Accelerating the image reconstruction algorithms is discussed in Ch. 3. External to this work, FAMI [MYS17a] demonstrates real-time imaging for security screening using a different forward model and image reconstruction algorithm also with a depth signal prior.

Table 5.1: Depth Camera Specifications [Zha12][SLK15]

Parameter	Kinect 1	Kinect 2
Color Resolution	640×480	1920×1080
Color FoV	$62.0° \times 48.6°$	$70.6° \times 60.0°$
Depth Resolution	320×240	512×424
Depth FoV	$57.0° \times 43.0°$	$70.6° \times 60.0°$
Range	$[0.8, 4.0]$m	$[0.5, 4.5]$m

5.1 Depth Cameras

Depth cameras are optical devices that measure depth "images". Each pixel in a depth image records a depth value that indicates how far away from the camera the object seen by that pixel is. These devices usually work in the infrared (IR) by active illumination, making their operation invisible to the human eye. Depth images can be mapped to position images, where each pixel records a 3D point relative to to the camera. Position images can be visualized as 3D point clouds.

5.1.1 Kinect 1

The Kinect 1 (Fig. 5.3) is a structured illumination depth camera. The device projects a known dot pattern on a scene and detects depth by parallax shifts in the pattern relative to a reference depth. This technique inherently has depth error on the order of $O(d^2)$, where d is the distance from the device [SLK15].

Multiple Kinect 1 depth cameras operating in the same scene will interfere. The structured illumination pattern from one depth camera will confuse a second depth

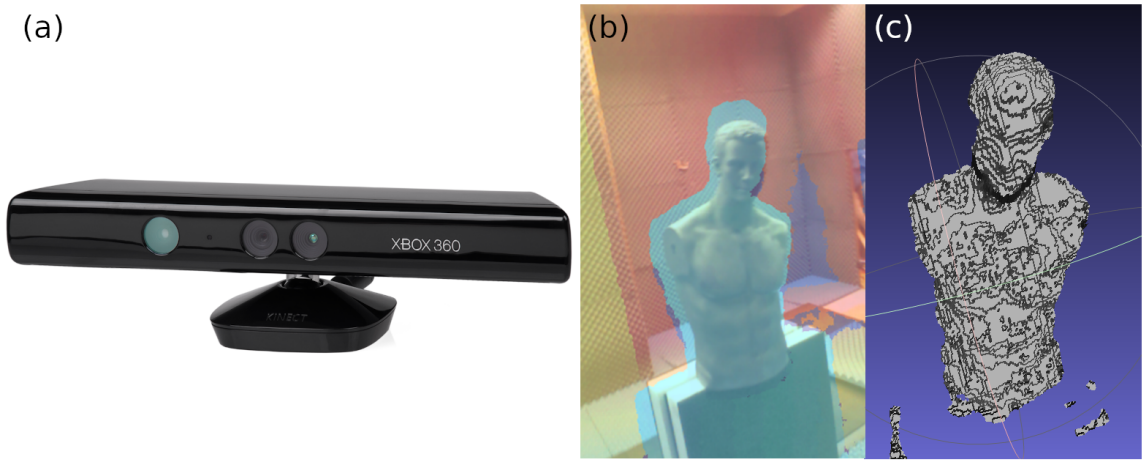

Figure 5.3: Kinect 1 — (a) device (b) depth image (c) position image

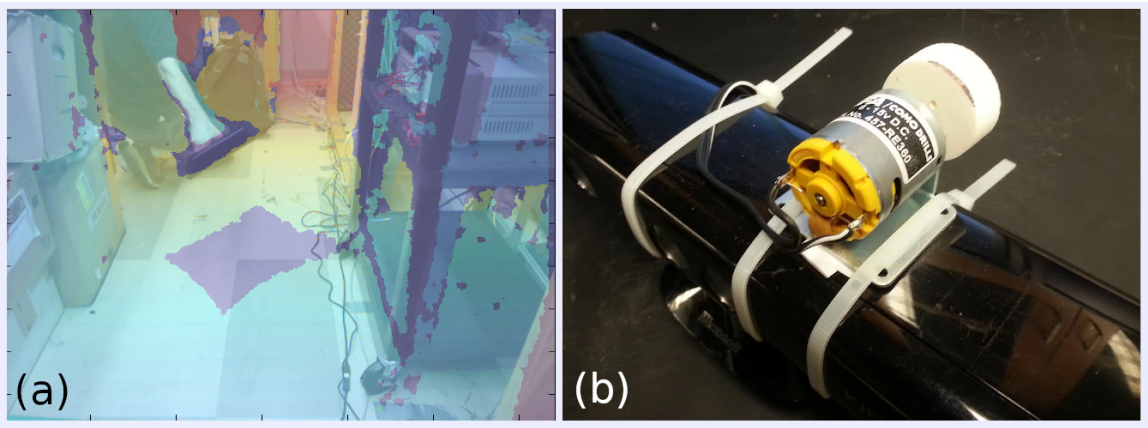

Figure 5.4: Structured Illumination Depth Camera Interference

camera trying to detect parallax in its own pattern. This causes holes to appear in the depth image where the patterns overlap (Fig. 5.4.a). The problem is severe enough to stymie utilizing multiple depth cameras without a solution. Time multiplexing the depth cameras is too slow, and optical choppers introduce unwanted synchronization and mechanical complexity. A clever and simple solution is to vibrate the depth cameras [MF12]. The projector and camera for each device are rigidly attached to each other, so a vibrating depth camera sees virtually no change in its projected dot pattern. However to an observer the vibration smears out the projected dot pattern, reducing its brightness. The effect is even greater when the observer is also vibrating. A weighted motor attached to the chassis of the camera provides the vibration (Fig. 5.4.b). This technique eliminates most interference.

5.1.2 Kinect 2

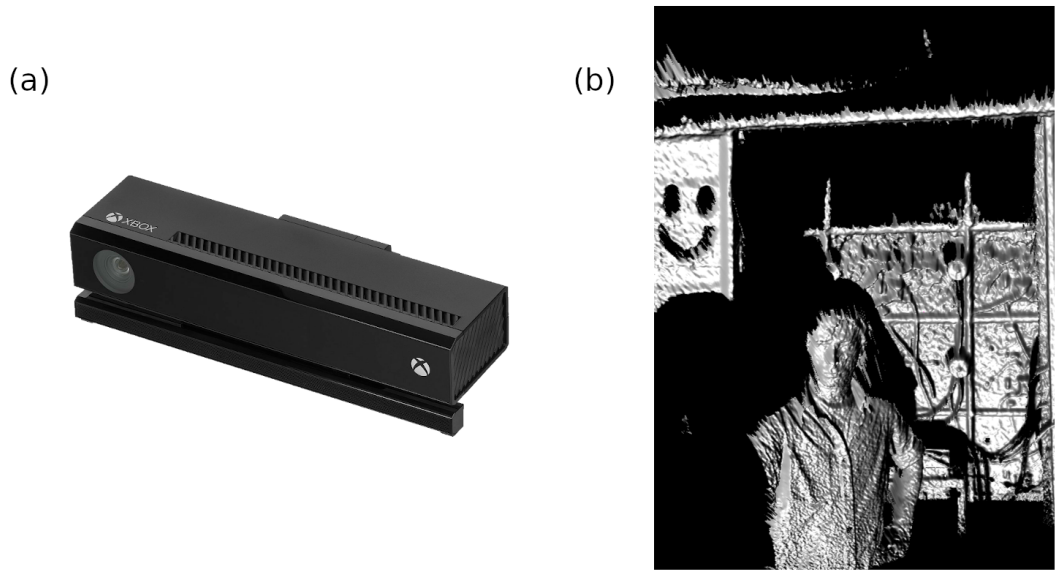

Figure 5.5: Kinect 2 — (a) device (b) position image

The Kinect 2 (Fig. 5.5) is a time of flight depth camera. Similar to radar and sonar, the device illuminates a scene with pulses of light and detects depth by measuring the travel time of the reflected signal. Time of flight technology has higher precision and accuracy over structured illumination devices. A pair of Kinect 2 depth cameras were used on the final experimental system.

Multiple Kinect 2 depth cameras can be operated in the same scene without special considerations. The sensors could theoretically interfere with each other, but experimentally this is not observed. Apparently during the extremely short amount of time a sensor is sensitive to its reflected pulse it is unlikely another device is illuminating the scene with its own pulse.

5.2 Depth Camera Registration

The depth cameras must be registered with the RF imaging system to inform the imaging model. This is achieved by applying constellation registration from Ch. 4.

A physical constellation with fiducials distinguishable in depth and RF reflectivity is required for registration. An experimental multisensor constellation is shown in Fig. 5.6. The body is made of extruded polystyrene foam for its low RF reflectivity and mechanical rigidity. Narrow foam stalks with flat ends that are sufficiently point-like and isolated in depth serve as depth camera fiducials. The stalk ends are covered in metal foil to serve as RF fiducials coincident with the depth camera fiducials (this is not required, but convenient). Shiny surfaces interfere with the function of the depth cameras, so the foil is covered in masking tape to give it a matte finish. Coincident photogrammetry fiducials are also added so the nominal constellation $\mathbf{C}_m$ can be accurately measured rather than designed. There are at least 3 asymmetric

fiducials to guarantee a unique constellation registration solution, and these fiducials are placed as far apart as practical to maximize registration accuracy.

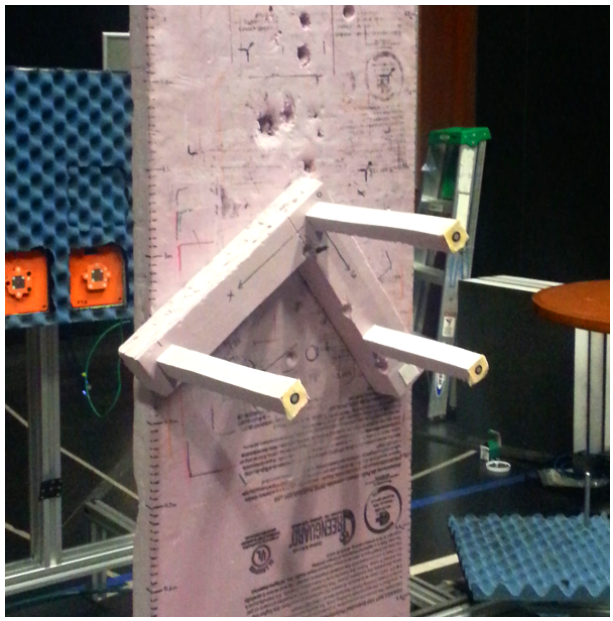

Figure 5.6: Experimental Multisensor Registration Constellation

Depth camera registration proceeds by placing the multisensor constellation in the imaging volume where its fiducials are simultaneously visible to all imaging devices being registered (Fig. 5.7). The constellation is imaged with each device. For each image the fiducial positions are estimated relative to the respective imaging device. The exact details depend on the imaging device and will be addressed shortly, but in all cases the result is a measured constellation. A depth camera measures constellation $\mathbf{D}_d$ w.r.t basis d, and is registered with the physical constellation using the nominal constellation $\mathbf{C}_m$

$$A_{dm} = \text{Procrustes2}(\mathbf{C}_m, \mathbf{D}_d). \tag{5.1}$$

Likewise, an RF imaging system measures constellation $\mathbf{D}_h$ w.r.t forward model basis

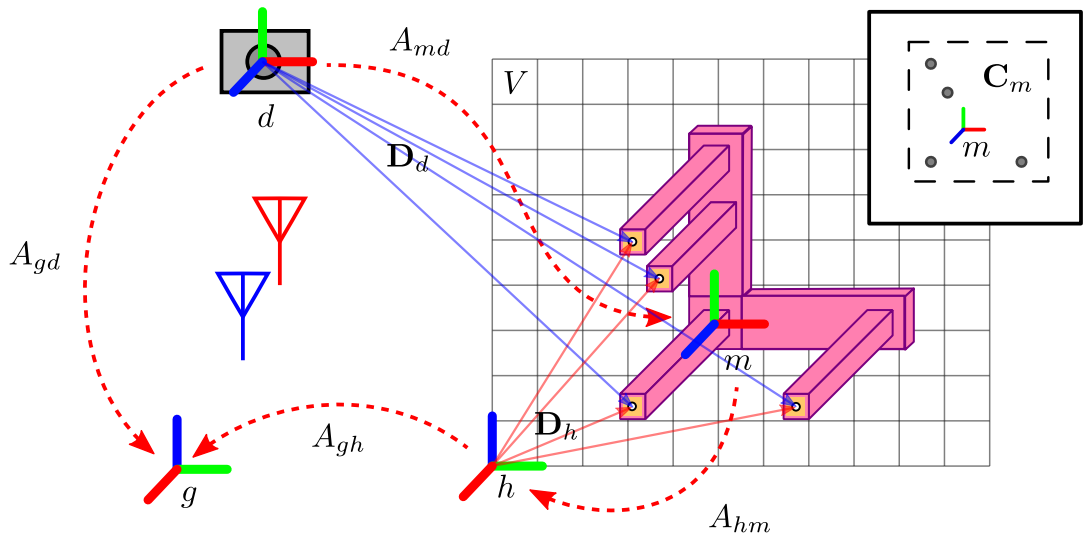

Figure 5.7: Depth Camera Registration

h, and is also registered with the physical constellation using the nominal constellation $\mathbf{C}_m$

$$A_{hm} = \text{Procrustes2}(\mathbf{C}_m, \mathbf{D}_h). \tag{5.2}$$

With rigid transformations A_{dm} and A_{hm} in hand, the camera basis d is registered to forward model basis h by the composition

$$A_{hd} = A_{hm} \circ A_{md}. \tag{5.3}$$

It is convenient to define a global basis g shared by all imaging devices. In this case forward model basis h will be registered with global basis g by a specified rigid transformation A_{gh}. Furthermore, there may be several depth cameras. Therefore the i-th depth camera basis d is registered to global basis g by the composition

$$\boxed{A_{gd,i} = A_{gh} \circ A_{hm} \circ A_{md,i}}. \tag{5.4}$$

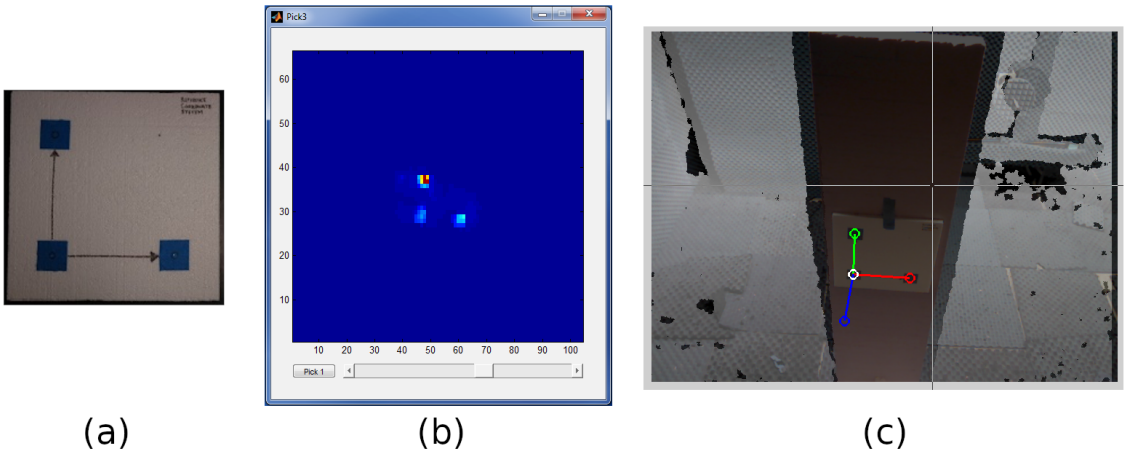

Figure 5.8: Depth Camera Registration — Experimental registration process of an early registration target (a) t

This method depends on estimating fiducial positions in 3D RF images and 2D depth images. The RF images are processed with the same strategy described in Sec. 4.4.1 and illustrated in Fig. 4.7. The 2D depth images are processed in a similar way. First, fiducials are manually located in a depth image. For each fiducial the local minimum depth value and its location are found in a small search region surrounding the manually selected location, and the 3D position values of pixels near the local minimum depth pixel are averaged to estimate the fiducial position. Care must be taken to not average invalid depth values.

An attempt to quantify the registration error that could be expected from this process was conducted. A target of 4 multisensor fiducials distributed in a vertical line was constructed. This target was imaged with the imaging system and a registered depth camera 16 times in a 4×4 2D grid, forming an array of 64 sample points. The fiducials were isolated in both the RF and depth and their 3D positions estimated can compared. The RF system is assumed to represent ground truth because there is

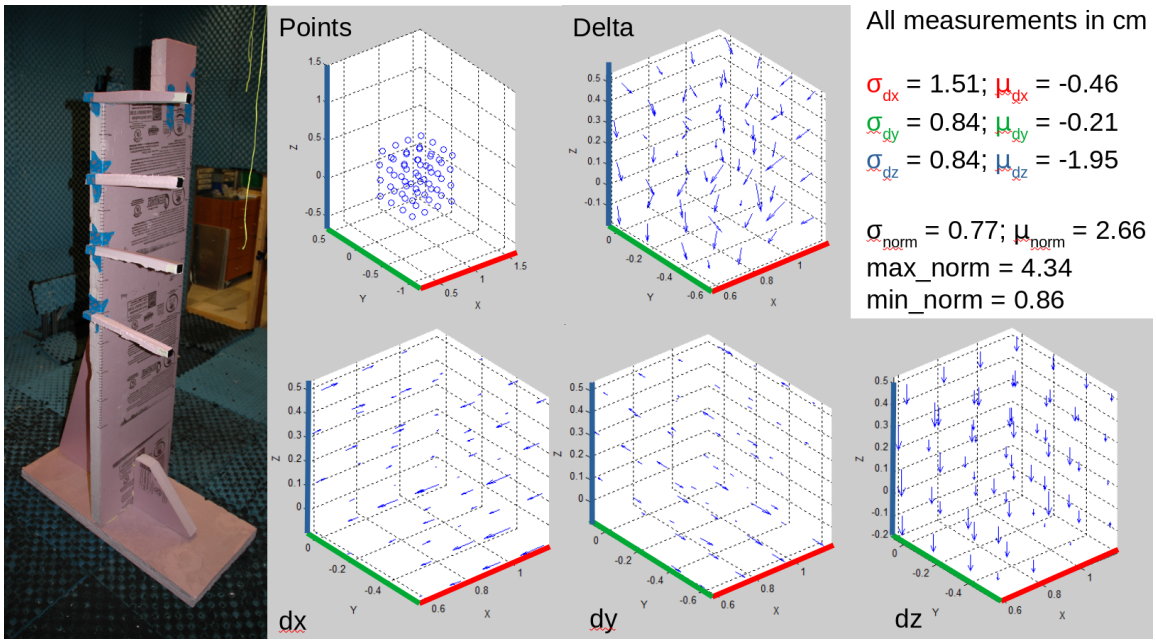

Figure 5.9: Depth Camera Registration Error

no source of spatial distortion in the forward model, unlike a structured illumination depth camera. Statistics were run on the axis-aligned components of the differences and the norm of the differences (Fig. 5.9). The maximum position error over a $1m^3$ centered about 1m in front of the imager is about 4.5cm. This is perfectly acceptable since even a coarse ROI is effective at reducing imaging times. Note, this data was taken early in development with a Kinect 1; a Kinect 2 would likely have smaller positioning error.

5.3 Measuring the ROI

The primary task of the depth cameras is to generate a tightly fitting volumetric ROI around objects in the RF foreground of an imaging volume. The ROI identifies which voxels to consider in the forward model. The process of turning registered

depth images into ROIs takes several steps: depth foreground detection, rasterization, combination, and resampling. The result is a logical mask flagging voxels in the ROI. ROIs are used by modified versions of the forward model to select a subset of the imaging domain.

5.3.1 Depth Foreground Detection

Depth image foreground detection is based on a background subtraction and thresholding strategy (Fig. 5.10).

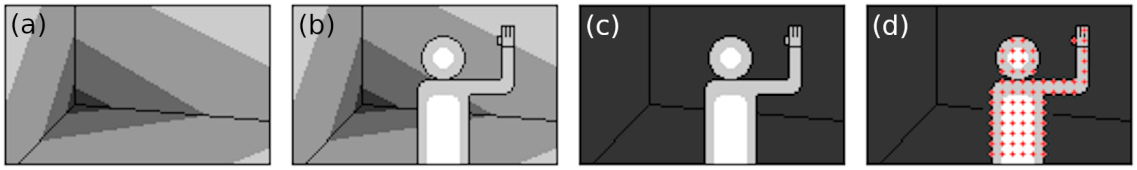

Figure 5.10: Depth Image Foreground Detection — (a) background (b) scene (c) foreground (d) subsampled foreground

Prior to depth foreground detection, a depth background is measured for each depth camera. This is done at the same time the RF background is measured to make the depth and RF backgrounds correspond to the same scene. The depth background is averaged over several frames to minimize the effects of noise.

MATLAB code for depth foreground detection is given in Listing 5.1. For a pixel to be in the foreground, its depth value must be valid (non-zero) and either the background value must be invalid (so no comparison can be made) or the depth value must be less than the background value by a specified threshold. The raw foreground is further processed by a median filter to dissolve small false positives and smooth

the result. Filtering can result in invalid pixels being marked as foreground, so pixels must have valid depth values to be identified as in the foreground.

Listing 5.1: Foreground Detection

```
function [fg_mask] = DepthForeground(z, z_bg, dz_threshold, neighborhood)
    % differentiate background from foreground
    fg_mask = z & (~z_bg | (z_bg - z) >= dz_threshold);
    % apply median filter (re-implement as filter and threshold)
    fg_mask = z & MedFilt2(fg_mask, neighborhood);
end
```

5.3.2 Rasterization

A process using a registered foreground depth image to identify foreground voxels in an imaging volume is now described (Fig. 5.11). It is assumed the imaging volume is a rectangular grid of voxels for efficiency, and its registration is known w.r.t. the depth camera. The result is a logical mask that flags foreground voxels.

First, the depth image is mapped to a position image. Each pixel in the position image specifies a point $\mathbf{r}_d \in \mathbb{R}^3$ w.r.t. depth camera basis d representing the position of the closest surface detected along the pixel's line of sight. Importantly, basis d coincides with the aperture in the pinhole camera model. This means a position measurement also identifies the line of sight on which it is situated. Along a pixel's line of sight, everything between the camera and measured position is transparent, while everything behind is occluded.

In a security screening application, objects of interest will be concealed underneath optically-opaque but RF-transparent clothing, so it is important to image voxels

behind surfaces detected by the depth camera. It is clearly important to model lines of sight to generate a ROI.

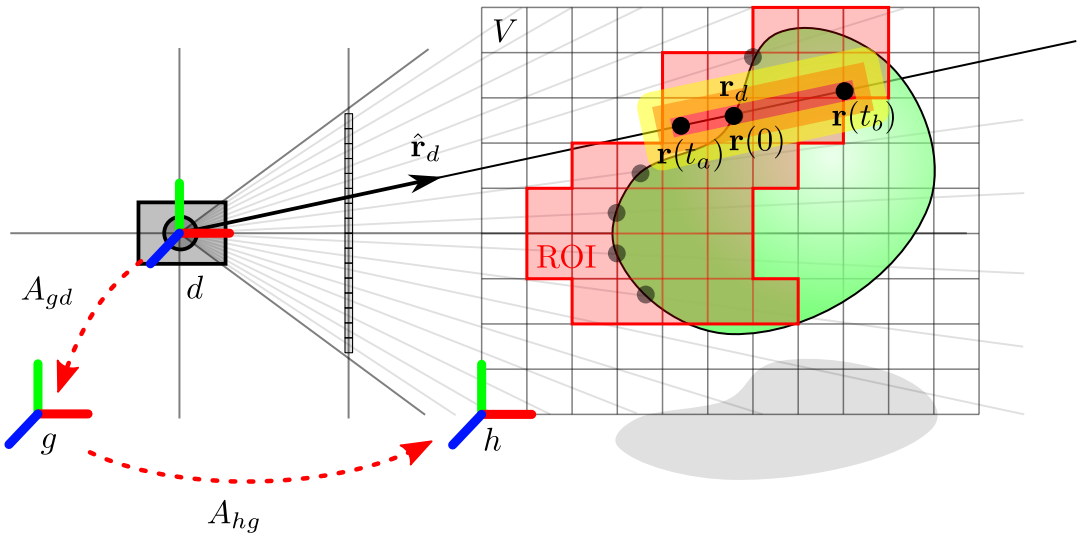

Figure 5.11: ROI Rasterization

A line can be represented by the parametric vector equation

$$\mathbf{r}(t) = \mathbf{r}_0 + \mathbf{d}t \tag{5.5}$$

where $\mathbf{r}_0 \in \mathbb{R}^3$ is a position vector on the line, $\mathbf{d} \in \mathbb{R}^3 - \{\mathbf{0}\}$ is a direction vector pointing along the line, and $t \in \mathbb{R}$ is the parameter. Consider a pixel's line of sight by setting $\mathbf{r}_0$ the measured position $\mathbf{r}_d$ and setting $\mathbf{d}$ to the unit vector $\hat{\mathbf{r}}_d = \mathbf{r}_d / \|\mathbf{r}_d\|$

$$\mathbf{r}(t) = \mathbf{r}_d + \hat{\mathbf{r}}_d t. \tag{5.6}$$

Making $\mathbf{d}$ a unit vector allows the parameter t to be specified in length units (i.e. meters). Evaluating the line at $t = 0$ returns the measured position $\mathbf{r}_d$, while $t < 0$ corresponds to the line before the surface and $t > 0$ to the line behind the surface.

A ROI along a line of sight relative to the measured surface position is specified by the parameter interval $t \in [t_a, t_b]$. This interval corresponds to the line segment with endpoints $\mathbf{r}(t_a)$ and $\mathbf{r}(t_b)$. Generally t_a is chosen to be slightly before the surface, and t_b is chosen to penetrate the surface far enough to capture hidden objects and skin, but not much deeper since RF signals attenuate rapidly in human skin. Typical values are $t_a = -0.02$m and $t_b = 0.08$m.

The ROI line segment end points are transformed to the forward model basis h with A_{hd}. The transformed position components are then mapped to grid indices with the function (as applied to the x component)

$$i_x = \left\lfloor \frac{x - x_0}{\Delta x} \right\rfloor \tag{5.7}$$

where x_0 is the minimum grid value, Δx is the pitch of the grid, and i_x is the resulting index. The result is index vectors $\mathbf{i}_a$ and $\mathbf{i}_b$.

Finally, the ROI line segment is drawn into the 3D imaging volume with a 3D line segment rasterization algorithm. A line algorithm is used instead of volumetric algorithms for simplicity and speed. There are many line segment rasterization algorithms. A conceptually simple algorithm is to iterate over a parametric line equation constructed from the two end points. The dimension with the largest difference between the line segment end points is iterated over to ensure the line is unbroken and one voxel thick. To simplify coding, the end points are potentially swapped to ensure the iteration dimension indices increase. A line is constructed by taking $\mathbf{i}_a$ as the reference position, and defining the direction vector $\mathbf{q}$ as

$$\mathbf{q} = \begin{cases} \frac{\mathbf{i}_b - \mathbf{i}_a}{\Delta i_{max}} & \text{if} \quad \Delta i_{max} \neq 0 \\ \mathbf{0} & \text{if} \quad \Delta i_{max} = 0 \end{cases} \tag{5.8}$$

where Δi_{max} is the difference between the starting and ending index along the iteration dimension. The line is evaluated and rounded at indices along the iteration dimension to generate the voxel indices of the line segment

$$\mathbf{i}_k = \text{round}(\mathbf{i}_a + \mathbf{q}k) \quad \text{for} \quad k \in [0, \Delta i_{max}]. \tag{5.9}$$

Voxel indices that are within the bounds of the imaging volume are flagged as ROI voxels. If a voxel falls outside the imaging volume, a flag is set to note this condition.

The process of rasterizing ROI line segments is repeated for all foreground pixels in a position image. This fills out a volumetric ROI using line segments.

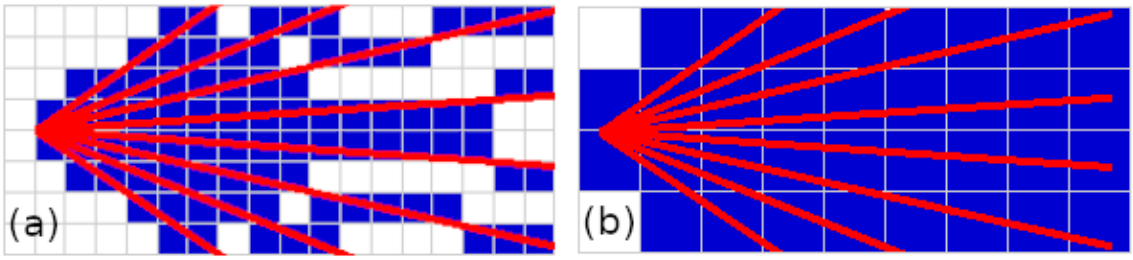

Figure 5.12: Line Rasterization Spread – (a) lines fail to fill out a volume (b) proper sampling

There are some downsides to the presented rasterization strategy. The most serious is that there is potentially a great deal of redundant calculation if lines are relatively closely spaced. This can be mitigated by rasterizing only a subset of foreground pixels. It should also be noted that if the ROI lines are very long, or if the imaging volume voxels are extremely fine, the lines will fail to fill out a volume (Fig. 5.12). However these situations are not encountered in the experimental system, as will be discussed.

5.3.3 Combining ROIs

Up until this point only a single depth camera has been considered when generating the ROI. Occlusion prevents any single camera from fully identifying the foreground of complicated scenes. Multiple depth cameras can be used to improve scene coverage. Each depth camera can provide a ROI for a given imaging volume. How best to combine these ROIs into a more complete ROI depends on the situation.

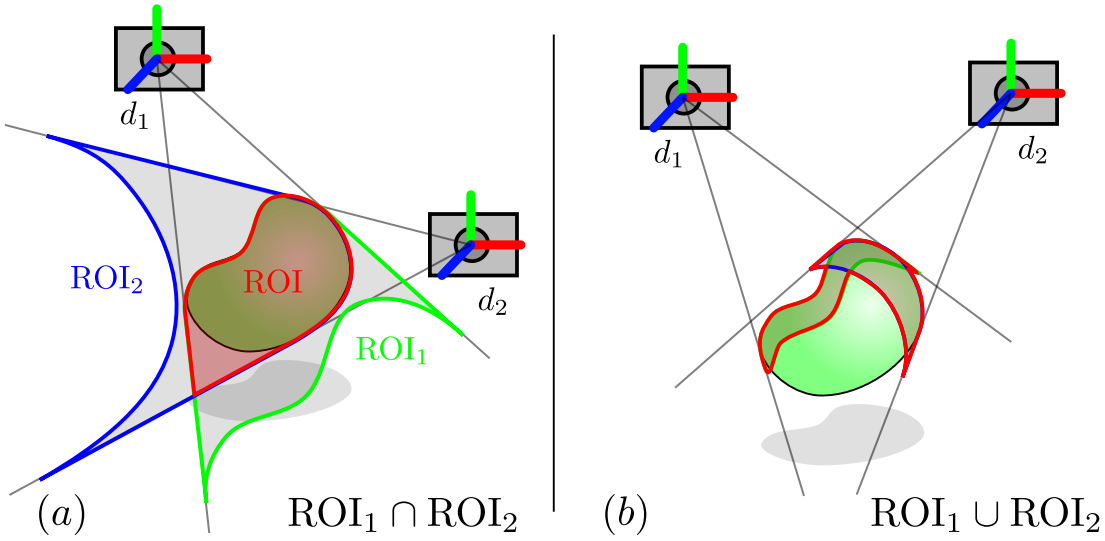

Figure 5.13: ROI Combination — (a) intersection (b) union

The ROI concept as applied to the imaging system relies on the observation that the forward model only needs to include voxels in the RF foreground. This seems to indicate all voxels occupied by foreground objects must be part of the ROI. The best way to isolate a volume occupied by an object is to surround the object with depth cameras, use long ROI line segments, and take the intersection of the ROIs by logically ANDing them (Fig. 5.13.a).

For security screening applications in K band, scenes are expected to be made

of conductive materials that strongly reflect waves from their surface and attenuate waves traveling into their interior. The interior of objects are effectively shielded and play no role in the forward model. Practically speaking the RF foreground volume is a subset of the actual foreground volume. In this case it is a shell of voxels around a foreground object on the side of the aperture that are part of the RF foreground; everything else is shielded by the body of the object. The best way to isolate a volumetric shell of the surface of an object on the side facing an aperture is to collocate cameras with the aperture, use short ROI line segments, and take the union of the ROIs by logically ORing them (Fig. 5.13.b).

5.3.4 Resampling

It has been assumed that the imaging volume and ROI share the same grid. This is not a requirement, and it can be convenient to use a courser grid for the ROI than for the RF image. This is because the ROI does not need to be highly accurate to be highly effective, and in a real-time application this can be leveraged to adjust the time it takes to compute the ROI.

Assuming the imaging volume and ROI volume coincide but differ in voxel pitch, the ROI volume can be mapped to imaging volume with a windowing transformation (App. B). This kind of transformation maps intervals onto other intervals. Conceptually the transformation translates the minimum value of one interval to the origin, scales the size of interval to the size of the desired interval, and then translates the origin to the minimum value of the second interval. When applied to voxel indices, it is important to think of indices as being in the center of voxels to derive the correct transformation. The equation for mapping ROI indices to (continuous) image indices

is

$$i_{x,img} = \left[i_{x,roi} + \frac{1}{2} \right] \frac{N_{x,img}}{N_{x,roi}} - \frac{1}{2}. \tag{5.10}$$

Counter-intuitively, it is actually the (rounded) inverse of this transformation that is needed

$$i_{x,roi} = \text{round} \left(\left[i_{x,img} + \frac{1}{2} \right] \frac{N_{x,roi}}{N_{x,img}} - \frac{1}{2} \right). \tag{5.11}$$

The inverse is used to ensure every image voxel is mapped to a ROI voxel. The image volume indices are mapped to ROI indices, and the ROI is effectively sampled by nearest neighbors by rounding.

5.4 Applying the ROI

With a ROI in hand, all that is left to do is reduce the imaging forward model. This process was described for the **H** matrix in the introduction of this chapter; voxels outside the ROI are pruned from the forward model by eliminating the corresponding components from **f** and columns from **H**.

If **H** is factored for practical reasons, applying the ROI to reduce the forward model necessarily becomes more complicated. APB factorization is an example of this. Fortunately applying a ROI to APB requires very little modification to the basic implementation. The ROI is used to select subdomains to reconstruct. The APB formulation tracks subdomains throughout it's operators, so the ROI is simply used to select the subdomain indices as they are processed.

The ROI reduces the size of the forward model to a size where real-time image reconstruction becomes feasible. This is supported by back-of-the-envelope numbers. The experimental imager typically models a screening volume of $1 \times 2 \times 2\text{m}^3$ par-

titioned into $1 \times 1 \times 1\text{cm}^3$ voxels, resulting in $N_v = 4 \times 10^6$ voxels. People have roughly 2m^2 of skin, and half will be visible to the imager corresponding to about 1×10^4 voxels. The volumetric ROI will multiply the number of voxels by roughly a factor of 10. Thus, a person occupies about $N = 1 \times 10^5$ voxels. The imager was designed with $N_{tx} = 24$ Tx, $N_{rx} = 72$ Rx, and $N_f = 100$ frequency points to give $M = 172800 \approx 2 \times 10^5$ measurements. Thus a typical $\mathbf{H}$ matrix is about $M * N_v = 8 \times 10^{11}$ elements, and the ROI reduces that to about 2×10^{10} elements or 5% of the original size. An application of a complex matrix will take $4 * M * N$ multiplications, and $4 * M * N$ additions, so about $8 * M * N$ FLOPs total. This means it takes roughly 16×10^{10} FLOPs, or 160 GFLOPs, to apply the reduced forward model. State-of-the-art GPUs have theoretical throughput on the order of 1 TFLOPs/s, putting real-time image reconstruction in reach. In addition, the reduced problem size helps fit the problem into memory.

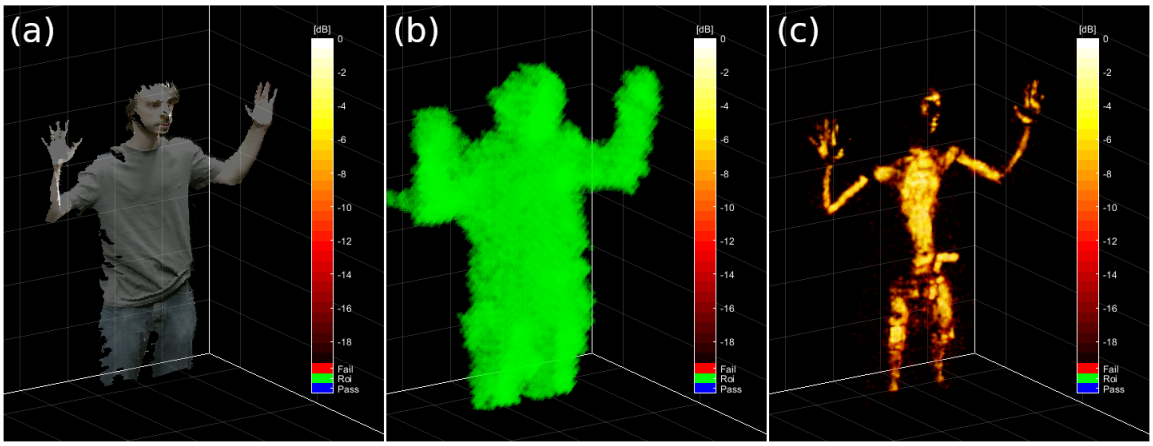

Figure 5.14: ROI Application — (a) color + surface (b) ROI (c) reconstructed image

A standard experimental example the ROI being applied is shown in Fig. 5.14.

Table 5.2: Reconstruction Times

Method	cELC Demo	Demo A	Demo B
MF	$3-5$min (GPU)	≈ 10s	≈ 6s
GMRES		$2.5-3.0$min	$2.5-3.0$min
FAMI			250ms (4 GPUs)

The ROI is clearly larger than the RF image, indicating that the image is not an artifact of the ROI. Image reconstruction times for various system revisions are given in Table 5.2. This 2 order of magnitude speedup in matched filter reconstruction times from the cELC demo to Demo A was enabled partially by the ROI.

Chapter 6

Stitching

Images produced by the imaging system must be screened for threats with automatic threat detection (ATD). There are two major factors that complicate image analysis. First, in K band many objects of interest are smooth on the scale of the wavelength and made out of materials that are highly conductive and consequently reflect like a mirror. This specular reflection becomes a problem for active illumination systems when illuminating waves are reflected away from the aperture, effectively limiting the visibility of objects to specular highlights that depend on the orientation of the object's surface relative to the probing antennas (Fig. 6.1). Second, the scene deforms over time. This is a unique property of the imaging system that greatly complicates understanding information contained in a set of images.

It is unclear what the best way to do ATD is for a walk-while-scan imager. This chapter explores stitching as a solution to the specularity and deformation problems. The observation can be made that the motion of a person through a screen-while-walk imager supplies many views of the person with different coverage. Intuitively, these images can be stitched together to produce a single image with better coverage, which may be easier to process with existing ATD. While the idea is simple, it is not clear how to combine multiple images of a deformable body into a single image in real-time. A model for stitching people in motion is systematically developed through experiment and simulation . Conceptually the technique can be applied to other 3D imagers.

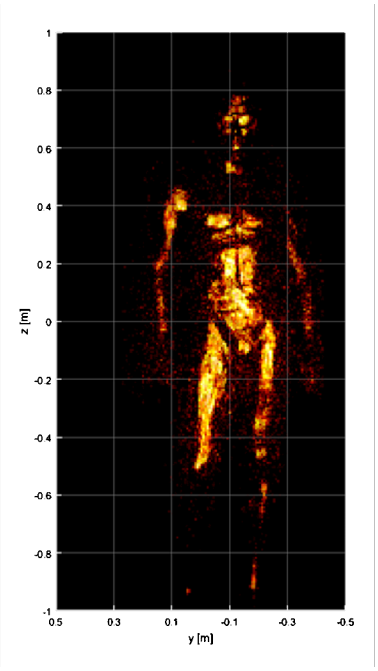

Figure 6.1: Raw Image

The primary benefit of image stitching is the potential to use existing ATD models trained for conventional stop-and-pose systems.

This chapter is divided into 3 sections. The first section covers image stitching fundamentals that are common to most stitching tasks. The second second then uses the rigid registration theory developed in Ch. 4 to implement a series of rigid stitching experiments. Notably, a rotation stitch produces images similar to commercial cylindrical SAR systems. The third section generalizes rigid stitching to build a stitching modeling based on a skeleton armature. The chapter culminates in a deformable stitching model suitable for the unique image sets produced by the imaging system.

6.1 Fundamentals

Image stitching seeks to combine multiple images of the same scene into a single composite image called a *stitch*. In general, stitching can be divided into three tasks: registration, calibration, and blending [BL07]. First, registration aligns images by modeling the geometric transformation relating images, establishing correspondences between images, and then estimating the transformation parameters between those correspondences. Next, calibration equalizes images to compensate for variations in the imaging process, making overlapping registered images locally similar. Finally, registered and calibrated images are blended to compute the resulting stitch by compositing. This section discusses how these tasks apply to millimeter wave images of people in motion, where specularity-limited coverage and deformation complicate the stitching process. The section concludes by outlining a basic stitching algorithm for rigid bodies.

6.1.1 Registration

Registration is perhaps the most involved stitching task, and the focus of this work. Humans are visually gifted by nature, and easily align overlapping images of a scene. However, distilling what we do into an algorithm is surprisingly difficult. The literature is briefly reviewed for inspiration.

A popular stitching problem is to combine multiple photographs into a panoramic stitch [MP94]. Due to user measurement patterns and camera image distortion, the geometric transformation between photos is usually parameterized by cylindrical or spherical projections [SSSS97]. Photo registration generally proceeds by correlation [Bro92], neural network inference [AED+16], or local feature analysis to detect corre-

spondences and estimate the geometric transformation parameters between photos. Local feature analysis in particular efficiently finds, describes, and matches localized image features deemed "interesting". Feature algorithms such as SIFT [Low99] and SURF [BTVG06] coupled with bundle adjustment [TMHF99] or stochastic search algorithms like RANSAC [FB87] are effective at solving the photo registration problem. However, while local feature analysis seamlessly extends to 3D images [SAS07], it is unclear if millimeter wave images characterized by specularity-limited coverage of mirror-like surfaces have sufficiently distinguishable features. Moreover, photo registration gives little insight into modeling a deformable scene.

Medical imaging is deeply concerned with the registration of deformable scenes. For instance, radiotherapy planning may use computational anatomy to accurately register anatomical structures against templates to safely predict and administer dosage [MA13]. However, sophisticated medical image analysis tends to be computationally prohibitive for real-time applications. A balance must be struck between accuracy and computational complexity.

To develop a registration model suitable for real-time millimeter wave imaging of people in motion, the first step is to investigate the subproblem of registering images of rigid body scenes that don't deform. With the language of rigid transformations described in Ch. 4 a registration model for rigid scenes can be developed (Fig. 6.2). The geometric transformation parameters, or *pose*, of a rigid scene is fully described by a rigid transformation with respect to some basis. The goal is to take a set of images of a rigid scene in different poses relative to global basis g and align them relative to stitch basis s. Suppose basis b is attached to the rigid scene and the pose A_{gb} can be measured for each image. Furthermore suppose a constant *rest pose* A_{bs}

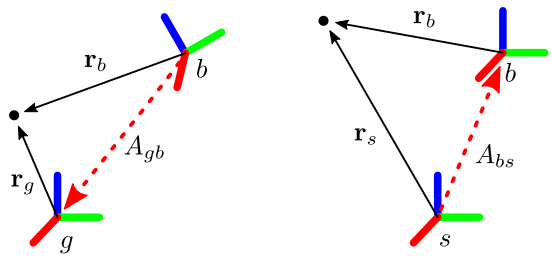

Figure 6.2: Rigid Registration Model — A rigid scene is imaged relative to global basis g (left). Images are aligned relative to stitch basis s (right). Suppose basis b is attached to the scene. Pose A_{gb} is experimentally measured each image. Rest pose A_{bs} is a constant defined to locate the stitch relative to basis s. Pose $A_{gs} = A_{gb} \circ A_{bs}$ then relates stitch points $\mathbf{r}_s$ to global points $\mathbf{r}_g$.

is defined to locate the scene relative to basis s. Then pose

$$A_{gs} = A_{gb} \circ A_{bs} \tag{6.1}$$

relates stitch points $\mathbf{r}_s$ to global points $\mathbf{r}_g$. Thus for rigid scene registration the main task is to measure pose A_{gb}.

Sec. 6.2 experimentally applies the rigid registration model using constellation registration (Ch. 4) to register rigid scenes undergoing translational and rotational motion. Sec. 6.3 extends the rigid registration model to deformable scenes by incorporating a pose skeleton, which is used to approximate the deformation of a scene using a small number of rigid transformations.

6.1.2 Calibration

Calibration makes overlapping registered images locally similar by correcting for variations in the imaging process. For photographic stitching, the primary concern is limited dynamic range, in which the dynamic range of a panorama exceeds the dynamic range of the camera, such as when stitching photos of bright sky and dark ground. Tone mapping is typically used to stitch these types of high dynamic range panoramas.

For millimeter wave images, the interpretation of signal magnitude between different images is more difficult to ascertain, which affects blending. The magnitude of the reconstructed images depends in a complicated way on scene configuration, dynamic range, signal-to-noise ratio (SNR), prior information, and regularized reconstruction algorithms. The magnitude of each image is effectively determined up to a different unknown scaling factor.

Experimental experience indicates that similar scenes have similar normalized magnitudes. Satisfactory preliminary stitching results have been obtained without extensive calibration. However, calibration must be done to properly blend dissimilar images for high quality ATD input. This remains an open avenue of research.

6.1.3 Blending

The last stitching task is to blend registered and calibrated images by compositing to produce the final stitch. For every stitch voxel, the associated stitch point $\mathbf{r}_s$ is transformed to global point $\mathbf{r}_g$, the source images are sampled using interpolation, the sampled data is combined by some blending operation, and the result is accumulated as stitch voxel data.

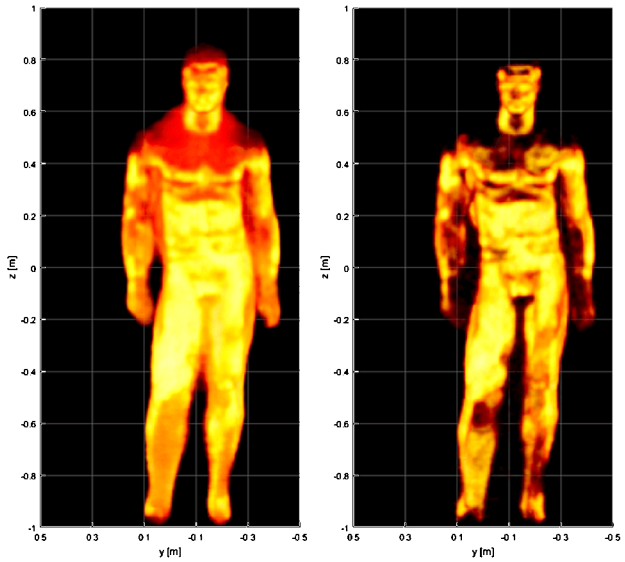

Figure 6.3: Blending — Stitched data from a rotation experiment is blended in two ways. Magnitude summation blending (left) is sensitive to noise and exposure time, while maximum magnitude blending (right) is not.

There is no one correct blending operation, and a variety of choices exist to varying effect (Figure 6.3). A plausible rule is to add the sampled image magnitudes together. This works but produces a stitch that is sensitive to exposure time and low-level noise. For data sets of unknown size, such as streaming data, a better rule is to assign the maximum magnitude of the sampled images, which mitigates the issues with addition. All stitching results use the maximum value blending strategy unless otherwise noted.

6.1.4 Algorithm

To stitch an image set, the following algorithm is used. Given input image set Σ, image reconstruction points $\mathbf{r}_g \in \mathbb{R}^3$ with respect to the global basis g, stitch points $\mathbf{r}_s \in \mathbb{R}^3$ with respect to stitch basis s, and rest pose A_{bs}, we wish to output a stitch. First initialize the stitch to zero. For each image, solve the registration problem to estimate pose A_{gb}. Then transform $\mathbf{r}_s$ with Eq. 6.1 to obtain query points $\mathbf{r}'_g$. Next, sample the image at the query points using interpolation to obtain registered image data. Finally, blend registered data with the current stitch to update the stitch. This algorithm is suited for image set Σ of unknown size.

6.2 Rigid Stitching

Due to the complexity of stitching millimeter wave images of a person in motion, first the rigid registration model (Eq. 6.1) developed in Sec. 6.1 is applied to perform rigid body stitching. Conceptually, a rigid scene and rigid imager move relative to each other along different trajectories while an image set is measured. In this section

Algorithm 7: Stitching

Input: Σ, $\mathbf{r}_g \subset \mathbb{R}^3$, $\mathbf{r}_s \subset \mathbb{R}^3$, $A_{bs} \in \mathrm{SE}(3)$

Output: stitch

stitch = 0

for image $\in \Sigma$ **do**

  A_{gb} = register(image)

  $\mathbf{r}'_g = A_{gb} \circ A_{bs}(\mathbf{r}_s)$

  data = interpolate(image, $\mathbf{r}_g$, $\mathbf{r}'_g$)

  stitch = blend(stitch, data)

end

two rigid-body experiments are discussed. The first experiment models a person being conveyed past an imaging system by using a mannequin on a 2D translation stage. The second experiment models a common SAR scanning configuration used in airport security by using a mannequin on a rotation stage.

The key task of rigid registration is to measure rigid scene pose A_{gb} with respect to the global imager basis g. The controlled experiments in this section photogrammetry with constellation registration (Fig. 6.4). A Creaform MaxSHOT 3D camera is used to measure the location of point-like reflective sticker fiducials relative to a photogrammetry prop basis p to produce a point cloud accurate to within imager tolerances [OIL+16] [Cre14]. Three or more fiducials are patterned in an asymmetric constellation on an object to unambiguously encode the scene pose. To isolate the constellation in the presence of other points, a special numbered fiducial is placed at the center of the constellation to group points by radius. The first time the constellation $\mathbf{D}_p$ is measured with respect to basis p, the points are used to construct

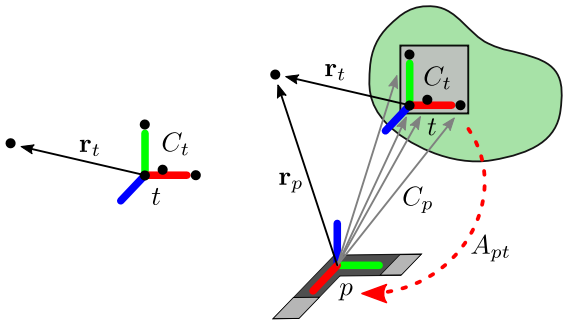

Figure 6.4: Photogrammetry Registration — Constellation C_t is designed relative to template basis t. A physical constellation is affixed to a rigid body and measured relative to photogrammetry basis p. Constellation registration is applied to estimate pose A_{pt}.

a template basis t into which they are transformed and recorded as constellation $\mathbf{C}_t$ in a template file. Subsequent measurements of constellation $\mathbf{D}_p$ are compared against the template file using OPA to determine pose A_{pt} from template basis t to photogrammetry basis p

$$A_{pt} = \text{Procrustes2}(\mathbf{C}_t, \mathbf{D}_p). \tag{6.2}$$

The pose for every antenna is determined in a similar manner as covered in Ch. 4. Choosing a particular antenna to serve as the global basis g, the pose A_{pg} is defined. Using the photogrammetry prop as an intermediate basis, the pose A_{gt} is computed as

$$A_{gt} = A_{gp} \circ A_{pt}. \tag{6.3}$$

If the constellation is affixed to a stage, basis t can be relabeled as scene basis b to measure pose A_{gb}, registering the stage with the global basis.

There is nothing special about photogrammetry, and in general any sensor system

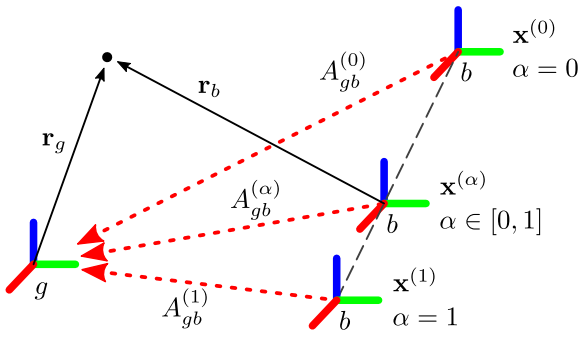

Figure 6.5: Translation Registration — A linear stage affixed with basis b is translated relative to global imager basis g. Terminal stage configurations $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(1)}$ and associated poses $A_{gb}^{(0)}$ and $A_{gb}^{(1)}$ are measured, and linearly interpolated with parameter α to compute $\mathbf{x}^{(\alpha)}$ and $A_{gb}^{(\alpha)}$.

capable of imaging points in $\mathbb{R}^3$ may be used to experimentally measure poses.

6.2.1 Translation stitching

In this experiment, a mannequin is placed on a 2D translation stage. The stage can be programmed with a sequence of stage parameter vectors $\mathbf{x} \in \mathbb{R}^2$ to follow any trajectory in a plane. To simplify the discussion attention is restricted to linear trajectories modeling a person being conveyed past an imaging system. The experiment proceeds by stepping a mannequin along a linear trajectory and collecting image data at each step.

The stage is equipped with a photogrammetry constellation so pose A_{gb} can be experimentally registered by Eq. 6.3. This is a time consuming process, so poses are measured for a small number of stage configurations and extrapolated to arbitrary poses from those measurements.

A linear trajectory can be constructed by linearly interpolating a pair of stage parameter vectors and their corresponding poses (Figure 6.5). The linear interpolation, or *lerp*, of two properties p_0 and p_1 by parameter $\alpha \in \mathbb{R}$ is defined as

$$\text{lerp}(p_0, p_1, \alpha) \equiv p_0 + (p_1 - p_0)\alpha. \tag{6.4}$$

Given stage parameter vectors $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(1)}$ with associated measured poses $A_{gb}^{(0)}$ and $A_{gb}^{(1)}$ and interpolation parameters $\alpha = 0$ and $\alpha = 1$, compute $\mathbf{x}^{(\alpha)}$ and $A_{gb}^{(\alpha)}$ for any α as

$$\mathbf{x}^{(\alpha)} = \text{lerp}(\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \alpha) \tag{6.5}$$

$$A_{gb}^{(\alpha)} = \text{lerp}(A_{gb}^{(0)}, A_{gb}^{(1)}, \alpha). \tag{6.6}$$

For a linear trajectory with $2 \leq N \in \mathbb{Z}$ points and equal-sized steps starting at $A_{gb}^{(0)}$ and ending at $A_{gb}^{(1)}$, let $\alpha = (n-1)/(N-1)$ where $n \in [1, N] \subset \mathbb{Z}$.

With image registration solved, the stitching algorithm is applied. Experimental image data and a stitch of a mannequin undergoing a linear translation trajectory past an imaging system are shown in Figure 6.6. It is clear that individual images suffer from specularity-limited coverage, while the stitch improves coverage on the side of the mannequin closest to the imager. This experiment indicates that a flat imaging system from a single perspective will be limited by specularity even when images are stitched, which suggests that different antenna configurations and trajectories should be considered.

6.2.2 Rotation stitching

A more interesting rigid body scene trajectory is pure rotational motion. Rotation mitigates specularity-limited scene coverage by viewing a scene from all sides. Imaging a rotating scene with a fixed aperture is equivalent to imaging a fixed scene with

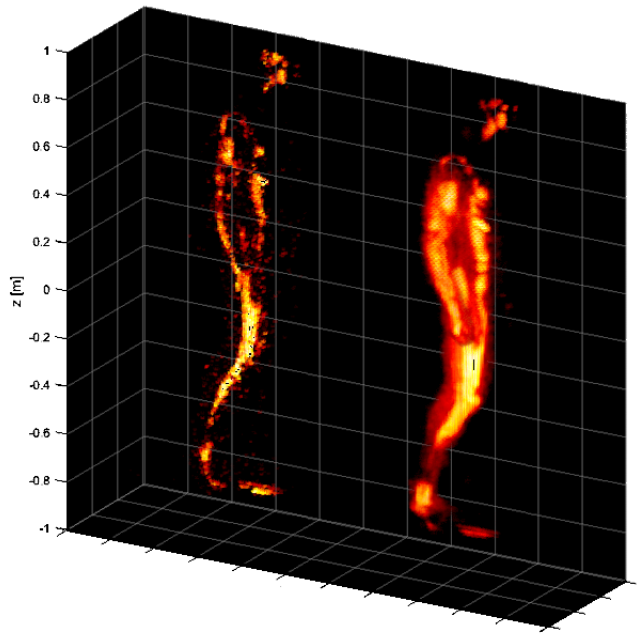

Figure 6.6: Experimental Translation Stitching — A single image (left) is compared against the stitch (right). Note the gains in coverage on the limbs.

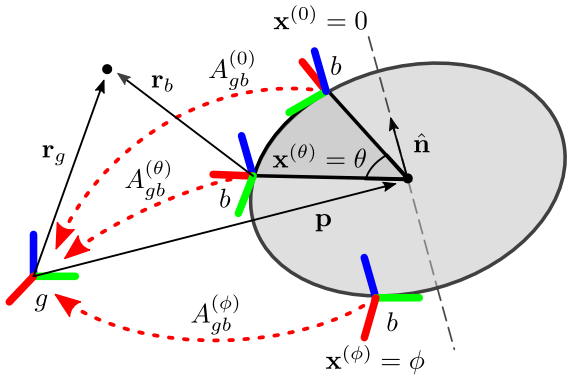

Figure 6.7: Rotation Registration — A rotation stage affixed with basis b is rotated relative to global basis g. For stage configurations $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(\phi)}$ associated poses $A_{gb}^{(0)}$ and $A_{gb}^{(\phi)}$ are measured. From these poses the stage axis $\hat{\mathbf{n}}$ and a point on that axis $\mathbf{p}$ are deduced, and pose $A_{gb}^{(\theta)}$ for any stage configuration $\mathbf{x}^{(\theta)}$ is computed.

an aperture counter-rotating around the scene. This mimics a common configuration found in commercial mechanically scanned SAR systems, permitting a fair comparison between imagers to be made. However, it should be noted that SAR coherently reconstructs a single image based on all collected measurements, while a real-time imager coherently reconstructs a sequence of images that are incoherently stitched into a single image.

In this experiment, a mannequin is placed on a rotation stage. The stage can be programmed with a sequence of stage configuration angles $\mathbf{x}^{(\theta)} = \theta \in [0, 2\pi) \subset \mathbb{R}$ to follow any rotational trajectory. The experiment proceeds by stepping a mannequin along a rotation trajectory and collecting image data at each step.

Once again photogrammetry is used to experimentally register scene pose A_{gb} for a small number of stage configurations to compute arbitrary poses (Figure 6.7). A rotation is fully described by axis direction $\hat{\mathbf{n}} \in S^2$, axis point $\mathbf{p} \in \mathbb{R}^3$, and angle θ.

$\hat{\mathbf{n}}$ and $\mathbf{p}$ are constant for any stage pose, so these can be estimated to construct new poses. Given stage configuration angles $\mathbf{x}^{(0)} = 0$ and $\mathbf{x}^{(\phi)} = \phi \neq 0$ with associated measured poses $A_{gb}^{(0)}$ and $A_{gb}^{(\phi)}$, compute pose $A_{gg}^{(\phi)}$ which encodes a rotation of ϕ about the stage axis relative to global basis g

$$A_{gg}^{(\phi)} = A_{gb}^{(\phi)} \circ A_{bg}^{(0)}. \tag{6.7}$$

To understand this equation consider scene basis b as fixed while basis g is transformed, and recognize this is equivalent to transforming points relative to fixed basis g. The rotation matrix $\mathbf{R}_{gg}^{(\phi)}$ can be extracted from $A_{gg}^{(\phi)}$ and convert it to axis-angle representation [Ebe02] to isolate axis direction $\hat{\mathbf{n}}$. However, the translation vector $\mathbf{t}_{gg}^{(\phi)}$ is not necessarily an axis point $\mathbf{p}$. This is because $\mathbf{R}_{gg}^{(\phi)}$ represents a rotation about the global origin and not the stage axis. To compute axis point $\mathbf{p}$, observe $\mathbf{p}$ must be unchanged by pose $A_{gg}^{(\phi)}$

$$\mathbf{p} = A_{gg}^{(\phi)}(\mathbf{p}) = \mathbf{R}_{gg}^{(\phi)}\mathbf{p} + \mathbf{t}_{gg}^{(\phi)}. \tag{6.8}$$

An axis is a line in 3D, so an infinite number of points satisfy this equation. To guarantee a unique solution solve for $\mathbf{p}$ in the 2D planar subspace perpendicular to axis direction $\hat{\mathbf{n}}$. The projection of $\mathbf{t}_{gg}^{(\phi)}$ on this subspace is

$$\mathbf{t}_{gg}^{(\phi)\prime} = \mathbf{t}_{gg}^{(\phi)} - (\mathbf{t}_{gg}^{(\phi)} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}. \tag{6.9}$$

Recasting (6.8) in 2D (denoted by primes), solve for $\mathbf{p}'$

$$\mathbf{p}' = (\mathbf{I}' - \mathbf{R}_{gg}^{(\phi)\prime})^{-1}\mathbf{t}_{gg}^{(\phi)\prime} \tag{6.10}$$

where $\mathbf{I}'$ is the identity matrix. Expanding $\mathbf{p}'$ relative to the 3D global basis g yields axis point $\mathbf{p}$. The scene pose $A_{gb}^{(\theta)}$ can now be constructed for arbitrary stage

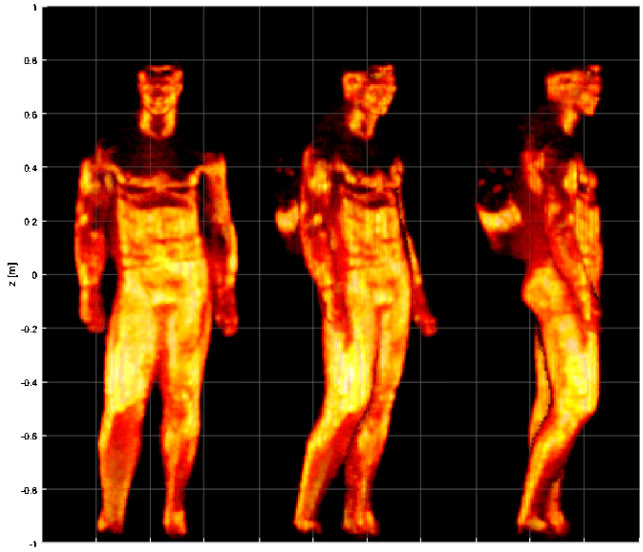

Figure 6.8: Experimental Rotation Stitch — A stitch of a mannequin wearing a backpack containing a pressure cooker is shown from three perspectives. Compare to raw image Fig. 6.1. Coverage is excellent, except for the top of the torso due to specularity. Images are comparable to SAR systems.

configuration angle $\mathbf{x}^{(\theta)} = \theta$. First construct pose $A_{gg}^{(\theta)}$ by moving the scene axis to the global origin by subtracting $\mathbf{p}$, then rotate about axis direction $\hat{\mathbf{n}}$ by angle θ, and finally restore the axis to its original position by adding $\mathbf{p}$

$$A_{gg}^{(\theta)}(\mathbf{r}_g) = \mathbf{R}(\hat{\mathbf{n}}, \theta)(\mathbf{r}_g - \mathbf{p}) + \mathbf{p}. \tag{6.11}$$

Functional composition with $A_{gb}^{(0)}$ yields the desired pose

$$A_{gb}^{(\theta)} = A_{gg}^{(\theta)} \circ A_{gb}^{(0)}. \tag{6.12}$$

For a rotation trajectory with $1 \leq N \in \mathbb{Z}$ points and equal-angle steps starting at $A_{gb}^{(0)}$, let $\mathbf{x}^{(\theta)} = \theta = 2\pi(n-1)/N$ where $n \in [1, N] \subset \mathbb{Z}$.

With registration solved, the stitching algorithm is applied. Experimental image data and a stitch of a mannequin undergoing a full rotation trajectory in front of an

imaging system are shown in Figure 6.1 and Figure 6.8. The stitch exhibits excellent scene coverage comparable to SAR systems [SMC+96] limited only by occlusion and specular surfaces significantly aligned with the axis. This results emphasizes the benefits of tightly controlling the imaging environment and imaging a scene from multiple perspectives. Unfortunately most of that control must be given up to stitch people in motion.

6.3 Skeleton stitching

The primary difficulty with stitching people in motion is that the scene is not a rigid body, but instead one that deforms, dramatically complicating the registration process. A single rigid transformation is no longer sufficient to fully characterize the scene pose. In this section the scene is segmented into a set of parts whose poses can be locally approximated by rigid transformations and combined with a deformation model.

It is impractical for security applications to affix photogrammetry constellations to people being imaged, therefore a different approach from Sec. 6.2 is needed to measure the pose of people. A serendipitous solution to this problem is to repurpose the depth cameras used to constrain the inverse imaging problem with a ROI (Ch. 5). The imager incorporates Kinect depth cameras which were originally designed as interfaces for video game systems and can fit skeleton armatures to people to approximate their pose.

The Kinect skeleton is organized as a set of *bones*, $b \in B$ whose individual poses are described by rigid transformation (Figure 6.9). The Kinect skeleton contains only about 25 bones, which is clearly not anatomically correct, reflecting the Kinect's

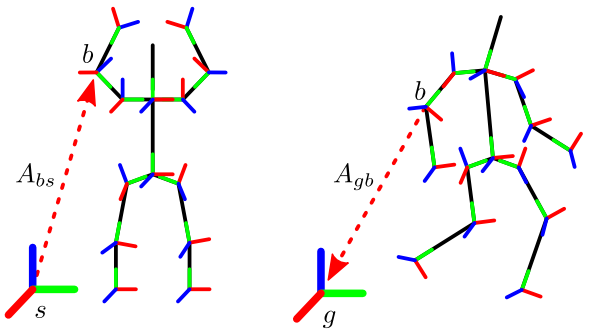

Figure 6.9: Skeleton Registration — A Kinect skeleton models a person's pose with a set of bones $b \in B$ and associated rigid transformations. The skeleton can take on any rest pose A_{bs} relative to stitch basis s (left). For each image, the skeleton pose A_{gb} is measured relative to global basis g (right).

intended usage as an entertainment device rather than a precision instrument. Thus, the Kinect skeleton is only a first order approximation of the scene pose. Regardless, it is a strong prior that can bootstrap more sophisticated registration in future research. Poses measured relative to depth camera basis c must be mapped to global basis g to perform stitching, so depth camera pose A_{gc} must first be measured by registration. This is addressed in Ch. 5.

With the depth cameras registered, depth and skeleton information are transformed into the global basis g to inform imaging and stitching. The skeleton by itself is not immediately useful for stitching.First the skeleton must be associated with a geometric "skin". Here a simple model is experimentally demonstrated that segments the scene into rigid subvolumes attached to bones. This is generalized to a model that accounts for deformation by using techniques from computer graphics, and simulated results using this advanced stitching model are presented.

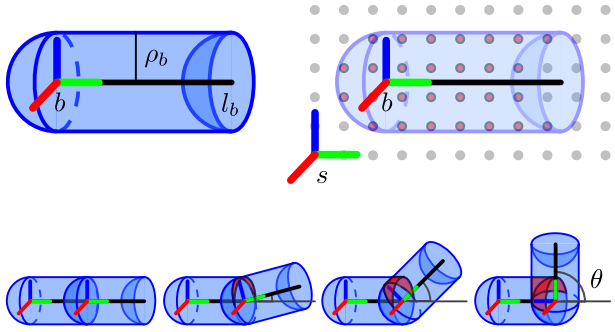

Figure 6.10: Rigid Subvolume — For each bone $b \in B$, a rigid pill characterized by radius ρ_b and length l_b is defined (top left). A grid relative to stitch basis s is partitioned by the rest pose subvolumes (top right). Transformed subvolumes overlap (red) when adjacent bones are not colinear (bottom).

6.3.1 Rigid Subvolume Stitching

A simple way to associate a skeleton with its surrounding volume is to define a rigid subvolume attached to each bone (Figure 6.10). An obvious candidate is an axis-aligned cylinder of some predetermined radius ρ_b and length l_b. Points are easily tested to be within this cylinder by checking if their projection on the axis falls within the interval of the bone and if their distance from the axis is less than the radius.

While a cylinder is a good subvolume for a single bone, cylinders do not smoothly connect at skeletal joints. A remedy is to cap the cylinder with spheres to create a pill-shaped subvolume. In this way, joints are much better represented. However, using a pill causes significant subvolume overlap around joints, double counting a large number of points. To mitigate this issue, the spherical cap belonging to the non-basis end of a bone is made to exclude points, resulting in a "dented" pill. When these subvolumes are mapped to a skeleton the overlap at joints is reduced.

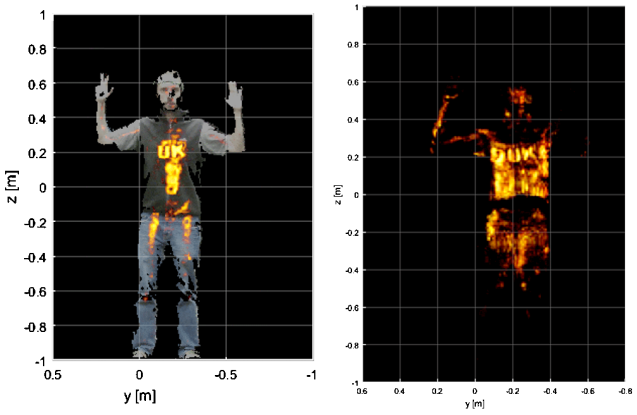

Figure 6.11: Experimental Rigid Subvolume Stitch — In no single image are the letters all visible (left); note the camera overlay. The the word "DUKE" is clearly read in the stitch (right). Note the crude stitching around joints caused by the simplistic registration model, particularly around the hips and thighs.

Before stitching a person in motion, an arbitrary skeleton must be selected to serve as the rest pose which defines the pose of the stitched image (Fig. 6.9). Conveniently, any experimental skeleton suffices, although rest poses with limbs spread apart are likely best suited for ATD. A grid of points is then defined relative to the stitch basis s. For each bone $b \in B$ with associated rest pose A_{bs}, a list of grid points $\mathbf{r}_{s,i} \in \mathbb{R}^3$ within the bone subvolume is tabulated, where the index $i \in I(b) \subset \mathbb{Z}$ enumerates points belonging to bone b. To stitch image data of a person in motion, skeleton data is measured for each frame to obtain the measured bone poses A_{gb} with respect to global basis g. For every bone $b \in B$, the associated list of grid points are transformed by

$$\mathbf{r}_{g,i} = A_{gb} \circ A_{bs}(\mathbf{r}_{s,i}); \quad i \in I(b) \tag{6.13}$$

solving the registration task. The stitching algorithm can then be applied.

Experimental demonstration of rigid subvolume stitching is presented in Figure 6.11. Here the author was imaged with a set of metal letters spelling "DUKE" across his chest and hidden under a vest. The subject faced the imager with their feet planted, and rotated their torso over their full range for 60 frames. In no single image are all 4 letters visible, however when stitched the full word is seen, lending credibility to the skeleton technique. However, it is clear that the Kinect skeleton is crude, the bone subvolumes are overly simplistic, and overlap between subvolumes is problematic. A more sophisticated deformation registration model is needed to realistically stitch people in motion.

6.3.2 SKD+SSD Stitching

Accurately modeling the deformation of a solid is computationally demanding. Approximations must be made for real-time imaging systems. A simple deformation model used to great effect in computer graphics is a combination of Shape Key Deformation (SKD) and Skeleton Space Deformation (SSD) [LCF00] (Figure 6.12). This model utilizes a skeleton conceptually compatible with the Kinect skeleton. SSD naturally extends rigid subvolume registration to account for realistic deformation. However, extreme deformations with SSD can cause unnatural results. SKD solves this issue by first modifying the underlying geometry as a function of skeleton pose before SSD is applied. While neither model is physically motivated, with enough effort, arbitrarily realistic deformations are possible.

SKD interpolates different *shape keys* with the same vertex topology to produce new shapes. Each shape key $k \in K$ defines a list of vertices $\mathbf{r}_{k,i}$ labeled with index $i \in I$. Stitch vertices $\mathbf{r}_{s,i}$ are computed by taking the weighted average of corresponding

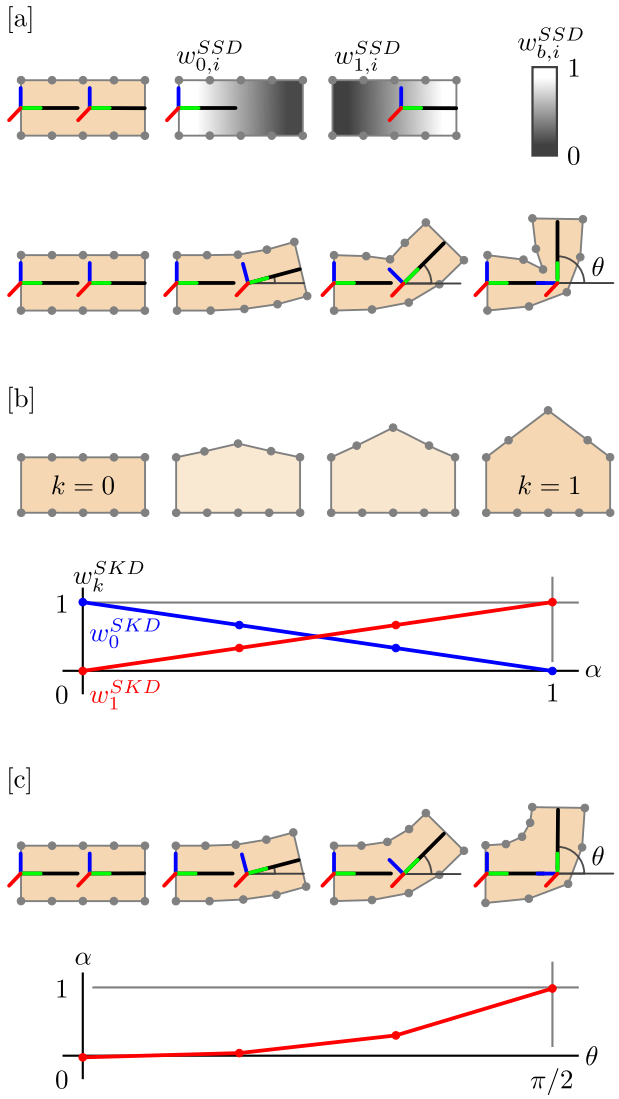

Figure 6.12: SKD+SSD — [a] SSD defines rest geometry and assigns weights $w_{b,i}^{SSD}$ for each bone and point (top). Deformation is achieved by rigidly transforming points with each bone and taking a weighed average (bottom). Shapes collapse for extreme poses. [b] SKD interpolates shape keys $k \in K$ with the same topology by weighted average of points with shape weights w_k^{SKD} (top). Weighting can driven by a function, e.g. α (bottom). [c] SKD computes rest geometry to address SSD collapse (top). Pose variables, e.g. angle θ between bones, can be mapped to SKD weights (bottom).

shape key vertices over all keys

$$\mathbf{r}_{s,i} = \sum_{k \in K} w_k^{SKD} \mathbf{r}_{k,i}; \quad \sum_{k \in K} w_k^{SKD} = 1 \tag{6.14}$$

where w_k^{SKD} are the shape key weights. The weights can be made to be a function of the skeleton pose. At one extreme, for every pose a different shape key could be defined, which would essentially model deformation by lookup table. However this would take a prohibitive amount of memory. Instead, additional shape keys are defined only when memory-efficient SSD results in problematic deformations.

SSD transforms geometry as a function of skeleton pose which effectively reduces the number of degrees of freedom needed to describe deformation. SSD is a modification of rigid registration Eq. 6.1. For every vertex $\mathbf{r}_{s,i}$ with respect to stitch basis s, the associated vertex $\mathbf{r}_{g,i}$ with respect to global basis g is computed by taking the weighted average of rigid transformations over all bones

$$\mathbf{r}_{g,i} = \sum_{b \in B} w_{b,i}^{SSD} A_{gb} \circ A_{bs}(\mathbf{r}_{s,i}); \quad \sum_{b \in B} w_{b,i}^{SSD} = 1 \tag{6.15}$$

where $w_{b,i}^{SSD}$ are the deformation weights for each bone $b \in B$ and vertex index $i \in I$. SSD strikes a good balance between computational complexity and realism. However, geometry influenced by a bone that is twisted or bent to an extreme degree relative to other bones causes pinching and collapsing of the resulting deformation. SKD is therefore applied before SSD to supply geometry suitable for deformation by the given skeleton pose.

While the governing equations of the SKD+SSD model are extremely simple, generating high quality data to feed the model is not a trivial task. The open source animation software Blender is used to facilitate this process. Blender is used to prepare a representation of a person by modeling surface geometry, defining a skeleton,

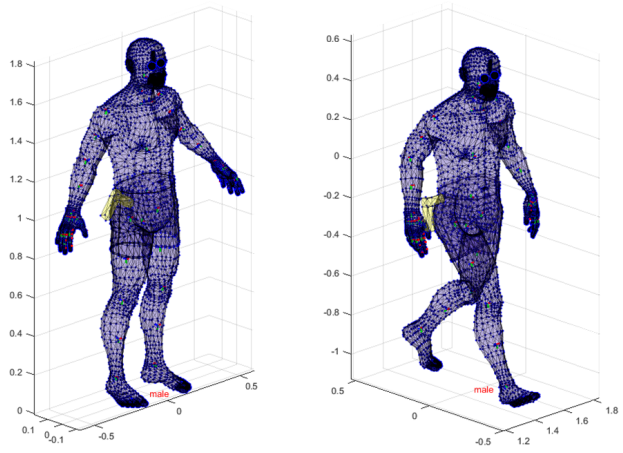

Figure 6.13: SKD+SSD Puppet — A life-like human model is posed with a skeleton and deformed with SKD+SSD. Note the threat object on the hip.

and weighting the geometry to the skeleton. Taken together, this data is called a *puppet* (Fig. 6.13). The puppet geometry can be realistically deformed by specifying a skeleton pose. The skeleton pose can be animated with Blender in simulation, or experimentally measured with depth cameras. For this pioneering work, research is restricted to simulations where the skeleton pose can be perfectly known.

Puppet and animation data are exported from Blender and imported into the imaging software. Imported geometric data is composed of vertices and triangular faces. Vertices are used to model point scatterers in simulation. Faces are supposed to represent surfaces, however if face vertices are farther apart than half the wavelength of the probing radiation, the face must be subdivided to correctly model a surface. In addition, the faces form a orientable surface. To stitch the volume immediately surrounding the body where worn objects are found, points are added to the geometry at each vertex along the average normal of adjacent faces in a "cactusing" procedure (Figure 6.14).

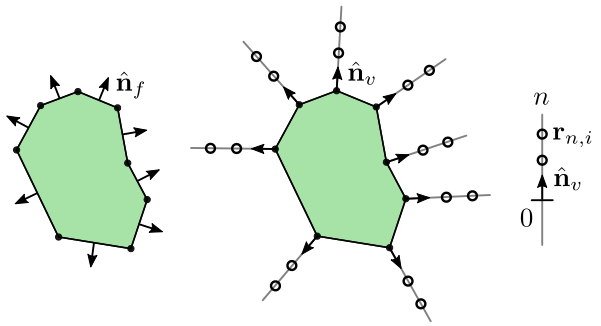

Figure 6.14: Cactusing — Geometry produced by computer graphics modeling software is usually an orientable surface composed of faces with normals $\hat{\mathbf{n}}_f \in S^2$ (left). To detect "worn" features above the surface, points are added along vertex normals $\hat{\mathbf{n}}_v \in S^2$ computed from face normals (center). The distribution of points is defined by a 1D point profile $\mathbf{r}_{n,i}$ (right).

A stitching simulation proceeds by selecting a frame of animation, assigning the corresponding skeleton pose to the puppet, transforming the puppet geometry to global basis g, and then applying the forward model. The inverse problem is then solved to reconstruct an image. To stitch, the skeleton pose can be reused to solve the registration problem. The stitching algorithm can then be applied. Stitch data can either be accumulated per vertex, or mapped to a regular grid relative to stitch basis s.

A simulated SKD+SSD stitch of a male puppet walking past an imager configured like the experimental system is shown in Figure 6.15. The puppet is equipped with a gun on his right thigh. When compared to the rigid subvolume stitch, the quality of the SKD+SSD stitch is much better: overlap is eliminated; joints are realistically deformed and stitched; the improved skeleton more accurately represents the pose of the scene; and the improved geometry better matches the human body. Whereas

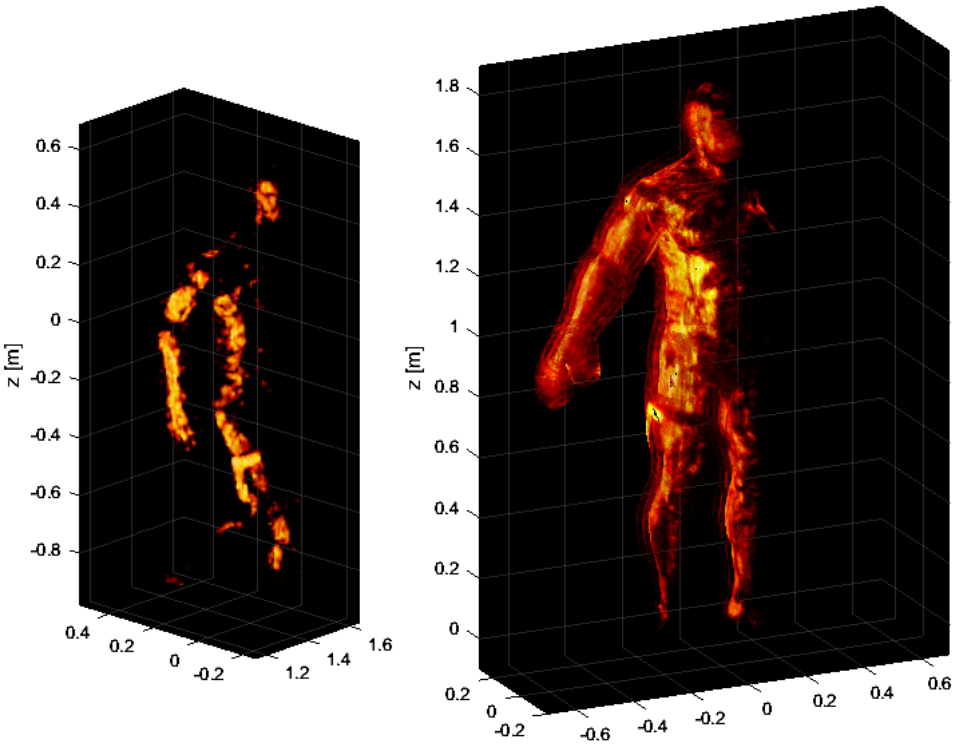

Figure 6.15: Simulated SKD+SSD Stitch — A male puppet wearing a threat object was animated walking past a simulated copy of the experimental imager. Specularity-limited coverage critically restricts the number of frames the threat is visible (left). Stitching with the same puppet greatly improves coverage, making the threat easily visible in a standardized pose (right).

ATD could miss the threat object in any single image, it is plausible that ATD would easily identify the threat on the person's thigh in the stitched image.

With perfect knowledge of the skeleton pose, this kind of simulation represents the gold standard for SKD+SSD stitching. The problem of experimentally estimating human pose has been extensively studied in the computer vision literature [HTTM12], but is still an active area of research. Combining the SKD+SSD deformation model with available experimental skeleton pose estimation is the next step in research.

With the introduction of SKD+SSD deformation and its incorporation into a full imaging system simulation suite, a host of future research questions can be answered without the need to construct physical experiments. For instance, the layout of the imager and trajectory a person takes through the imaging volume must be optimized to maximize coverage while minimizing system costs. Hallways, U-turns, dog-legs, and arches are all being considered, and can be rapidly evaluated on a computer. In addition, a calibration solution must be developed so images can be properly combined. It is hypothesized that a calibration object present in every image could be used to normalize the magnitude of the images. With good coverage of the scene established, a process that matches a person to a puppet from a library must be developed to stitch a wide variety of body shapes. The most challenging unsolved task is to estimate the skeleton pose of a real person in the scene which is beyond the limitations of the Kinect skeleton functionality utilized in experiment.

Chapter 7

Conclusion

7.1 Summary

This work discussed the design and implementation of an experimental walk-while-scan microwave imaging system. Ch. 1 motivated microwave imaging for security screening applications, and introduced a high-level description of the RF structured illumination imaging system. Ch. 2 developed the measurement model in detail. Ch. 3 derived the algorithms used to efficiently invert the measurement model. When combined, the measurement model and reconstruction algorithm define the imaging model. Ch. 4 describes the mathematics of registration, and develops the critical imager task of antenna registration. Ch. 5 describes how to integrate depth cameras into the experimental system. These depth cameras are used for several purposed, but their primary purpose is to reduce the size of the imaging model by providing prior information identifying the location of objects in the foreground. Ch. 6 describes a series of stitching experiments as a first attempt at image analysis. A deformable skeleton stitching model was proposed and demonstrated in simulation.

This work attempted to summarize the design and operation of the experimental imaging system. The scope of the project was extremely broad, and this was an ambitious undertaking. I was unable to met the goals I had set out for myself, but the story doesn't end here.

7.2 Future Research

7.2.1 Layout Analysis

A line of research partially resolved but not reported on was analyzing the optimal layout of several imaging system apertures. While the flat design of the experimental imager was an important design goal, it became obvious that there are serious drawbacks. Namely that only one side of a target is ever visible and that specularity reduces even that. The flexibility offered by a modular system is perfect for experimenting with different multiple aperture layouts; halls, tunnels, u-bends, dog-legs, etc.

Some interesting new questions can be asked when considering multiple apertures. First, it should be determined if two apertures should share the same radio or not. Diametrically opposed apertures might not seem to benefit from a shared radio, as the measurement model does not consider transmission measurements. However there may be opportunities to utilize opposing antennas, such as when a person is approaching two apertures. Knowing which measurements to take and when in this situation is a problem I call measurement planning.

7.2.2 Body Geometry and Pose Estimation

Deformation of the scene and specularity in the images complicate ATD, which motivated the deformable skeleton stitching model. However this model was only demonstrated in simulation. To demonstrate the model in experiment, the two related problems of body geometry estimation and pose estimation must be solved simultaneously. This is a complicated task in machine vision and machine learning. Sig-

nificant progress on generating realistic body geometries and measuring a person's pose in the case of multiple color cameras has been made. I believe that with the addition of depth cameras and the RF imaging system an iterative approach can be devised that cyclically estimates a person's body geometry from an image set and a pose estimate, and then estimates a person's pose from the same image set and a body geometry estimate.

Many exciting applications can be explored if this problem is solved, from taking suit measurements like a tailor, to gait analysis in physical therapy work.

7.2.3 Automatic Threat Detection

An ATD algorithm was never really developed for the imaging system. A standard approach for ATD is training machine learning algorithms for anomaly detection [LHG+18]. After an anomaly is detected, classification can be attempted. Conceptually, ATD could be trained on data of people in different poses with and without threat objects, however direct application of this idea is intractable because the space of human poses is enormous, so the stitching concept was proposed. In this way existing ATD algorithms can be applied. However state-of-the-art ATD algorithms are not generally available.

Generating synthetic training data sets has become an increasingly attractive option for machine learning tasks. Collecting training data is time consuming and expensive. However with the ability to generate realistic human bodies from body geometry and pose estimation, it is straight forward to generate realistic scenes and images of randomly generated people equipped with randomized objects of interest. This would greatly facilitate the development of ATD. Training data images can

even be labeled by rendering images using solid color codes. This could also sidestep touchy details when working with human data sets.

7.3 Vision

Modular screen-while-walk imaging systems invite a full reevaluation of security screening. Deployment options are expanded by relaxing assumptions about the configuration of the aperture and how people interact with the device. For instance, systems unobtrusively installed at portals and along corridors can opportunistically image passerbys. At a lower cost point, multiple checkpoints become feasible, which could cooperatively build models of people in an area. Increased screening throughput and distributed sensor networks would prevent vulnerable crowds from forming at checkpoints, and could move the secure area boundary outside critical infrastructure such as airport terminals.

Appendix A

Electromagnetism Review

This appendix assembles Maxwell's equations for electrodynamics from first principles, reviews advanced electromagnetic theorems, and derives a radiation model. The theoretical development closely follows Griffiths and Balanis. MENTION PEC, PMC

A.1 Electrostatics

Electrostatics is the physics of source charge distributions that are constant in time.

A.1.1 Coulomb's Law

Coulomb's law is the empirical equation for the force $\mathbf{F}$ on a point electric charge q at position $\mathbf{r}$ by a source point electric charge q' at source position $\mathbf{r}'$

$$\mathbf{F} = \frac{1}{4\pi\epsilon_0}\frac{\hat{\mathbf{R}}}{R^2}qq'. \tag{A.1}$$

Coulomb's law is linear and thus the forces of source charges obey superposition

$$\mathbf{F} = q\left[\frac{1}{4\pi\epsilon_0}\sum_i \frac{\hat{\mathbf{R}}_i}{R_i^2}q_i'\right]. \tag{A.2}$$

The bracketed factor is a vector field independent of q. Define the electric field as

$$\boxed{\mathbf{F} = q\mathbf{E}} \tag{A.3}$$

$$\mathbf{E}(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\sum_i \frac{\hat{\mathbf{R}}_i}{R_i^2}q_i'. \tag{A.4}$$

Charges that are small and numerous can be modeled as a continuum. Summation over point charges becomes integration of charge density over a geometric domain

$$\boxed{\mathbf{E}(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\int \frac{\hat{\mathbf{R}}}{R^2} dq'} \tag{A.5}$$

$$\boxed{dq' = \lambda(\mathbf{r}')dL' = \sigma(\mathbf{r}')dS' = \rho(\mathbf{r}')dV'.} \tag{A.6}$$

In particular, the electric field of a volume charge density over volume V is

$$\mathbf{E}(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\int_V \frac{\hat{\mathbf{R}}}{R^2}\rho(\mathbf{r}')dV'. \tag{A.7}$$

A.1.2 Divergence and Curl

Helmholtz decomposition implies a vector field is specified by its divergence and curl.

Consider the divergence of $\mathbf{E}$ for an electric volume density. Apply divergence, exchange the order of divergence and integration, and apply product rule (3)

$$\nabla\cdot\mathbf{E} = \frac{1}{4\pi\epsilon_0}\int_V \nabla\cdot\left[\frac{\hat{\mathbf{R}}}{R^2}\rho\right]dV' = \frac{1}{4\pi\epsilon_0}\int_V \left[\nabla\rho\cdot\frac{\hat{\mathbf{R}}}{R^2} + \rho\nabla\cdot\frac{\hat{\mathbf{R}}}{R^2}\right]dV'. \tag{A.8}$$

$\nabla\rho(\mathbf{r}') = \mathbf{0}$ because ρ is not a function of $\mathbf{r}$. Care must be taken to evaluate the second term with the identity $\nabla\cdot[\hat{\mathbf{R}}/R^2] = 4\pi\delta(\mathbf{R})$ (see Vector Calculus Appendix)

$$\nabla\cdot\mathbf{E} = \frac{1}{4\pi\epsilon_0}\int_V [0 + 4\pi\delta(\mathbf{R})\rho(\mathbf{r}')]dV' = \frac{\rho(\mathbf{r})}{\epsilon_0}. \tag{A.9}$$

The result is **Gauss's law** in differential form

$$\boxed{\nabla\cdot\mathbf{E} = \frac{\rho}{\epsilon_0}.} \tag{A.10}$$

Integrate this over a volume

$$\int_V \nabla\cdot\mathbf{E}dV = \frac{1}{\epsilon_0}\int_V \rho dV \tag{A.11}$$

define the total enclosed charge as

$$\boxed{Q \equiv \int_V \rho(\mathbf{r}) dV} \tag{A.12}$$

and apply the divergence theorem to obtain Gauss's law in integral form

$$\boxed{\oint_S \mathbf{E} \cdot d\mathbf{S} = \frac{Q}{\epsilon_0}.} \tag{A.13}$$

Compute the curl of $\mathbf{E}$ by first considering a single source electric point charge at $\mathbf{r}' = \mathbf{0}$ and form a line integral in spherical coordinates

$$\int_L \mathbf{E} \cdot d\mathbf{L} = \int_L \left[\frac{1}{4\pi\epsilon_0} \frac{q'}{r^2} \hat{\mathbf{r}} \right] \cdot [\hat{\mathbf{r}} dr + \hat{\boldsymbol{\theta}} r d\theta + \hat{\boldsymbol{\phi}} r \sin\theta d\phi] = \frac{q'}{4\pi\epsilon_0} \int_{r_a}^{r_b} \frac{dr}{r^2} = -\frac{1}{4\pi\epsilon_0} \frac{q'}{r} \Big|_{r=r_a}^{r_b}. \tag{A.14}$$

The line integral only depends on the end points. A closed loop evaluates to zero

$$\boxed{\oint_L \mathbf{E} \cdot d\mathbf{L} = 0} \tag{A.15}$$

and the curl theorem implies

$$\boxed{\nabla \times \mathbf{E} = \mathbf{0}.} \tag{A.16}$$

A.1.3 Scalar Potential

Line integrals of electrostatic fields only depend on the end points, not the path. Thus a scalar potential function can be defined relative to some reference point $\mathbf{r}_0$

$$\boxed{V(\mathbf{r}) \equiv -\int_{\mathbf{r}_0}^{\mathbf{r}} \mathbf{E}(\mathbf{r}') \cdot d\mathbf{L}'.} \tag{A.17}$$

Subtract V evaluated at $\mathbf{r}_a$ from $\mathbf{r}_b$, swap limits of integration, and combine integrals

$$V(\mathbf{r}_b) - V(\mathbf{r}_a) = -\int_{\mathbf{r}_0}^{\mathbf{r}_b} \mathbf{E} \cdot d\mathbf{L}' + \int_{\mathbf{r}_0}^{\mathbf{r}_a} \mathbf{E} \cdot d\mathbf{L}' = -\int_{\mathbf{r}_a}^{\mathbf{r}_b} \mathbf{E} \cdot d\mathbf{L}'. \tag{A.18}$$

Compare this to the gradient theorem to identify

$$\boxed{\mathbf{E} = -\nabla V.} \tag{A.19}$$

Insert this into Gauss's law to derive **Poisson's equation**

$$\boxed{\nabla^2 V = -\frac{\rho}{\epsilon_0}.} \tag{A.20}$$

The scalar potential relative to infinity of a source electric point charge at $\mathbf{r}' = \mathbf{0}$ is

$$V(\mathbf{r}) = -\int_{\infty}^{r} \left[\frac{1}{4\pi\epsilon_0}\frac{q'}{r'^2}\hat{\mathbf{r}}\right] \cdot [\hat{\mathbf{r}}dr'] = -\frac{q'}{4\pi\epsilon_0}\int_{\infty}^{r}\frac{dr'}{r'^2} = \frac{1}{4\pi\epsilon_0}\frac{q'}{r}. \tag{A.21}$$

Generalize this to arbitrary source positions

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\frac{q'}{R} \tag{A.22}$$

then to an arbitrary number of source charges

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\sum_i \frac{q_i'}{R_i} \tag{A.23}$$

then to a continuous charge distribution

$$\boxed{V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\int \frac{dq'}{R}.} \tag{A.24}$$

In particular, the scalar potential for a volume charge density is

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\int_V \frac{\rho(\mathbf{r}')}{R}dV'. \tag{A.25}$$

A.1.4 Polarization

Taylor expand R^{-1} about the origin (see Vector Calculus Appendix)

$$\frac{1}{R} = \frac{1}{r}\sum_{n=0}^{\infty}\left[\frac{r'}{r}\right]^n P_n(\cos\theta) \tag{A.26}$$

and insert this series into the equation for scalar potential of a volume charge density

$$\boxed{V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\sum_{n=0}^{\infty}\frac{1}{r^{[n+1]}}\int_V [r']^n P_n(\cos\theta)\rho(\mathbf{r}')dV'.} \tag{A.27}$$

Expand the first few terms

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\left[\frac{1}{r}\int_V \rho dV' + \frac{1}{r^2}\int_V r'\cos\theta\rho dV' + \ldots\right]. \tag{A.28}$$

The first term is the potential of an electric monopole moment at the origin

$$V_{mono}(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\frac{Q}{r}. \tag{A.29}$$

The second term is the potential of an electric dipole moment at the origin

$$V_{di}(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\frac{1}{r^2}\int_V r'\cos\theta\rho dV'. \tag{A.30}$$

This equation can be cast into vector form by observing $r'\cos\theta = \hat{\mathbf{r}}\cdot\mathbf{r}'$

$$V_{di}(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\frac{1}{r^2}\int_V \hat{\mathbf{r}}\cdot\mathbf{r}'\rho dV' = \frac{1}{4\pi\epsilon_0}\frac{1}{r^2}\hat{\mathbf{r}}\cdot\int_V \mathbf{r}'\rho dV' = \frac{1}{4\pi\epsilon_0}\frac{\hat{\mathbf{r}}}{r^2}\cdot\mathbf{p} \tag{A.31}$$

and the electric dipole moment is defined as

$$\boxed{\mathbf{p} \equiv \int_V \mathbf{r}'\rho(\mathbf{r}')dV'.} \tag{A.32}$$

Generalize the dipole potential to a point dipole at an arbitrary source position

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\frac{\hat{\mathbf{R}}}{R^2}\cdot\mathbf{p} \tag{A.33}$$

then to a continuous electric dipole moment volume density with $d\mathbf{p}(\mathbf{r}) = \mathbf{P}(\mathbf{r})dV$

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\int_V \frac{\hat{\mathbf{R}}}{R^2}\cdot\mathbf{P}(\mathbf{r}')dV'. \tag{A.34}$$

Use vector identity $\nabla R^n = nR^{n-1}\hat{\mathbf{R}}$ to rewrite $\hat{\mathbf{R}}/R^2 = \nabla' R^{-1}$

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\int_V [\nabla' R^{-1}]\cdot\mathbf{P}(\mathbf{r}')dV' \tag{A.35}$$

and integrate by parts by using product rule (3) and the divergence theorem

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\int_V [\nabla' \cdot [R^{-1}\mathbf{P}] - R^{-1}\nabla' \cdot \mathbf{P}]dV' \tag{A.36}$$

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\oint_S \frac{\mathbf{P} \cdot d\mathbf{S}'}{R} - \frac{1}{4\pi\epsilon_0}\int_V \frac{\nabla' \cdot \mathbf{P}}{R}dV'. \tag{A.37}$$

Compare these integrals to the scalar potential of a continuous charge distribution

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\oint_S \frac{\sigma_b}{R}dS' + \frac{1}{4\pi\epsilon_0}\int_V \frac{\rho_b}{R}dV' \tag{A.38}$$

and define bound electric surface and volume charge densities as

$$\boxed{\sigma_b = \mathbf{P} \cdot \hat{\mathbf{n}}} \tag{A.39}$$

$$\boxed{\rho_b = -\nabla \cdot \mathbf{P}.} \tag{A.40}$$

Electric charge can be split into bound charge and free (not bound) charge

$$\boxed{\rho = \rho_f + \rho_b.} \tag{A.41}$$

Insert this into Gauss's law

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} = \frac{1}{\epsilon_0}[\rho_f + \rho_b] = \frac{1}{\epsilon_0}[\rho_f - \nabla \cdot \mathbf{P}] \tag{A.42}$$

and isolate the free charge

$$\nabla \cdot [\epsilon_0\mathbf{E} + \mathbf{P}] = \rho_f. \tag{A.43}$$

Define the $\mathbf{D}$ field as

$$\boxed{\mathbf{D} \equiv \epsilon_0\mathbf{E} + \mathbf{P}.} \tag{A.44}$$

Gauss's law in matter in differential form is

$$\boxed{\nabla \cdot \mathbf{D} = \rho_f} \tag{A.45}$$

and in integral form is

$$\boxed{\oint_S \mathbf{D} \cdot d\mathbf{S} = Q_f.} \tag{A.46}$$

A.2 Magnetostatics

Continuity equation

$$\boxed{\oint_S \mathbf{J} \cdot d\mathbf{S} = -\frac{\partial Q}{\partial t}} \tag{A.47}$$

$$\int_V \nabla \cdot \mathbf{J} dV = -\frac{\partial}{\partial t} \int_V \rho dV = -\int_V \frac{\partial \rho}{\partial t} dV \tag{A.48}$$

$$\boxed{\nabla \cdot \mathbf{J} = -\frac{\partial \rho}{\partial t}} \tag{A.49}$$

Magnetostatics

$$\boxed{\nabla \cdot \mathbf{J} = 0} \tag{A.50}$$

A.2.1 Biot-Savart Law

$$\mathbf{F}_m = q\mathbf{v} \times \mathbf{B} \tag{A.51}$$

$$\boxed{\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int \frac{d\mathbf{I}' \times \hat{\mathbf{R}}}{R^2}} \tag{A.52}$$

$$\boxed{d\mathbf{I}' = \mathbf{I}(\mathbf{r}')dL' = \mathbf{K}(\mathbf{r}')dS' = \mathbf{J}(\mathbf{r}')dV'} \tag{A.53}$$

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_V \frac{\mathbf{J}(\mathbf{r}') \times \hat{\mathbf{R}}}{R^2} dV' \tag{A.54}$$

A.2.2 Divergence and Curl

Divergence of B

$$\nabla \cdot \mathbf{B} = \frac{\mu_0}{4\pi} \int_V \nabla \cdot \left[\mathbf{J} \times \frac{\hat{\mathbf{R}}}{R^2} \right] dV' = \frac{\mu_0}{4\pi} \int_V \left[\frac{\hat{\mathbf{R}}}{R^2} \cdot \nabla \times \mathbf{J} - \mathbf{J} \cdot \nabla \times \frac{\hat{\mathbf{R}}}{R^2} \right] dV' \tag{A.55}$$

$\nabla \times \mathbf{J}(\mathbf{r}') = \mathbf{0}$

$\nabla \times [\hat{\mathbf{R}}/R^2] = \mathbf{0}$ (see vector calc review)

$$\boxed{\nabla \cdot \mathbf{B} = 0} \tag{A.56}$$

$$\int_V \nabla \cdot \mathbf{B} dV = \int_V 0 dV \tag{A.57}$$

$$\boxed{\oint_S \mathbf{B} \cdot d\mathbf{S} = 0} \tag{A.58}$$

Curl of B

$$\nabla \times \mathbf{B} = \frac{\mu_0}{4\pi} \int_V \nabla \times \left[\mathbf{J} \times \frac{\hat{\mathbf{R}}}{R^2} \right] dV' = \frac{\mu_0}{4\pi} \int_V \left[\mathbf{J} \left[\nabla \cdot \frac{\hat{\mathbf{R}}}{R^2} \right] - [\mathbf{J} \cdot \nabla] \frac{\hat{\mathbf{R}}}{R^2} \right] dV' \tag{A.59}$$

$\nabla \times [\mathbf{A} \times \mathbf{B}] = \mathbf{A}[\nabla \cdot \mathbf{B}] - \mathbf{B}[\nabla \cdot \mathbf{A}] - [\mathbf{A} \cdot \nabla]\mathbf{B} + [\mathbf{B} \cdot \nabla]\mathbf{A}$

$\nabla \cdot [\hat{\mathbf{R}}/R^2] = 4\pi\delta(\mathbf{R})$

$$- [\mathbf{J} \cdot \nabla] \frac{\hat{\mathbf{R}}}{R^2} = [\mathbf{J} \cdot \nabla'] \frac{\hat{\mathbf{R}}}{R^2} \tag{A.60}$$

$$\hat{\mathbf{x}} \int_V [\mathbf{J} \cdot \nabla'] \frac{R_x}{R^3} dV' = \hat{\mathbf{x}} \int_V \left[\nabla' \cdot \left[\mathbf{J} \frac{R_x}{R^3} \right] - \frac{R_x}{R^3} \nabla' \cdot \mathbf{J} \right] dV' = \hat{\mathbf{x}} \oint_S \mathbf{J} \frac{R_x}{R^3} \cdot d\mathbf{S}' = \mathbf{0} \tag{A.61}$$

$\nabla \cdot [f\mathbf{A}] = \nabla f \cdot \mathbf{A} + f \nabla \cdot \mathbf{A}$

$$\int_V [\mathbf{J} \cdot \nabla] \frac{\hat{\mathbf{R}}}{R^2} dV' = \mathbf{0} \tag{A.62}$$

$$\nabla \times \mathbf{B} = \frac{\mu_0}{4\pi} \int_V \mathbf{J}(\mathbf{r}')[4\pi\delta(\mathbf{R})] dV' = \mu_0 \mathbf{J}(\mathbf{r}) \tag{A.63}$$

$$\boxed{\nabla \times \mathbf{B} = \mu_0 \mathbf{J}} \tag{A.64}$$

$$\int_S [\nabla \times \mathbf{B}] \cdot d\mathbf{S} = \mu_0 \int_S \mathbf{J} \cdot d\mathbf{S} \tag{A.65}$$

$$\boxed{I = \int_S \mathbf{J} \cdot d\mathbf{S}} \tag{A.66}$$

$$\boxed{\oint_L \mathbf{B} \cdot d\mathbf{L} = \mu_0 I} \tag{A.67}$$

A.2.3 Vector Potential

$$\boxed{\mathbf{B} = \nabla \times \mathbf{A}} \tag{A.68}$$

There is no restriction on divergence of A, and for now it's nice to get rid of it. Suppose $\nabla \cdot \mathbf{A}_o \neq 0$. Add a gradient (doesn't change the curl) $\mathbf{A} = \mathbf{A}_0 + \nabla\lambda$

$$\nabla \cdot \mathbf{A} = \nabla \cdot \mathbf{A}_0 + \nabla^2\lambda \tag{A.69}$$

$$\nabla^2\lambda = -\nabla \cdot \mathbf{A}_0 \tag{A.70}$$

$$\lambda = \frac{1}{4\pi}\int_V \frac{\nabla \cdot \mathbf{A}_0}{R} dV' \tag{A.71}$$

$$\boxed{\nabla \cdot \mathbf{A} = 0} \tag{A.72}$$

Poisson's equation x3

$$\nabla \times [\nabla \times \mathbf{A}] = \nabla\nabla \cdot \mathbf{A} - \nabla^2\mathbf{A} = \mu_0\mathbf{J} \tag{A.73}$$

$$\boxed{\nabla^2\mathbf{A} = -\mu_0\mathbf{J}} \tag{A.74}$$

vector potential of current distribution

$$\boxed{\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi}\int_V \frac{\mathbf{J}(\mathbf{r}')}{R} dV'} \tag{A.75}$$

$$\frac{1}{R} = \frac{1}{r}\sum_{n=0}^{\infty}\left[\frac{r'}{r}\right]^n P_n(\cos\theta) \tag{A.76}$$

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0 I}{4\pi}\oint_L \frac{d\mathbf{L}'}{R} \tag{A.77}$$

A.2.4 Magnetization

$$\boxed{\mathbf{A}(\mathbf{r}) = \frac{\mu_0 I}{4\pi}\sum_{n=0}^{\infty}\frac{1}{r^{[n+1]}}\oint_L [r']^n P_n(\cos\theta) d\mathbf{L}'} \tag{A.78}$$

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0 I}{4\pi}\left[\frac{1}{r}\oint_L d\mathbf{L}' + \frac{1}{r^2}\oint_L r'\cos\theta d\mathbf{L}' + \frac{1}{r^3}\oint_L \frac{r'^2}{2}[3\cos^2\theta - 1] d\mathbf{L}' + \ldots\right] \tag{A.79}$$

$$\oint_L d\mathbf{L}' = \mathbf{0} \tag{A.80}$$

$$\mathbf{A}_{mono}(\mathbf{r}) = \mathbf{0} \tag{A.81}$$

$$\mathbf{A}_{di}(\mathbf{r}) = \frac{\mu_0}{4\pi}\frac{I}{r^2}\oint_L r' \cos\theta d\mathbf{L}' \tag{A.82}$$

$\hat{\mathbf{r}} \cdot \mathbf{r}' = r' \cos\theta$

$\oint_L \mathbf{c} \cdot \mathbf{r} d\mathbf{L} = \int_S d\mathbf{S} \times \mathbf{c}$

$$\mathbf{A}_{di}(\mathbf{r}) = \frac{\mu_0}{4\pi}\frac{I}{r^2}\oint_L \hat{\mathbf{r}} \cdot \mathbf{r}' d\mathbf{L}' = \frac{\mu_0}{4\pi}\frac{I}{r^2}\left[\int_S d\mathbf{S}'\right] \times \hat{\mathbf{r}} \tag{A.83}$$

$$\mathbf{A}_{di}(\mathbf{r}) = \frac{\mu_0}{4\pi}\frac{\mathbf{m} \times \hat{\mathbf{r}}}{r^2} \tag{A.84}$$

$$\boxed{\mathbf{m} \equiv I\int_S d\mathbf{S}} \tag{A.85}$$

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi}\frac{\mathbf{m} \times \hat{\mathbf{R}}}{R^2} \tag{A.86}$$

$d\mathbf{m}' = \mathbf{M}(\mathbf{r}')dV'$

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi}\int_V \frac{\mathbf{M}(\mathbf{r}') \times \hat{\mathbf{R}}}{R^2} dV' \tag{A.87}$$

$[\hat{\mathbf{R}}/R^2] = \nabla' R^{-1}$

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi}\int_V \mathbf{M} \times \nabla' \frac{1}{R} dV' \tag{A.88}$$

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi}\int_V \left[\frac{1}{R}\nabla' \times \mathbf{M} - \nabla' \times \frac{\mathbf{M}}{R}\right] dV' \tag{A.89}$$

$\int_V \nabla \times \mathbf{A} dV = -\oint_S \mathbf{A} \times d\mathbf{S}$ (see vec calc appendix)

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi}\int_V \frac{\nabla' \times \mathbf{M}}{R} dV' + \frac{\mu_0}{4\pi}\oint_S \frac{\mathbf{M} \times d\mathbf{S}'}{R} \tag{A.90}$$

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi}\int_V \frac{\mathbf{J}_b}{R} dV' + \frac{\mu_0}{4\pi}\oint_S \frac{\mathbf{K}_b}{R} dS' \tag{A.91}$$

$$\boxed{\mathbf{J}_b = \nabla \times \mathbf{M}} \tag{A.92}$$

$$\boxed{\mathbf{K}_b = \mathbf{M} \times \hat{\mathbf{n}}} \tag{A.93}$$

$$\mathbf{J} = \mathbf{J}_f + \mathbf{J}_b \tag{A.94}$$

$$\nabla \times \mathbf{B} = \mu_0[\mathbf{J}_f + \mathbf{J}_b] = \mu_0[\mathbf{J}_f + \nabla \times \mathbf{M}] \tag{A.95}$$

$$\nabla \times \left[\frac{1}{\mu_0}\mathbf{B} - \mathbf{M}\right] = \mathbf{J}_f \tag{A.96}$$

$$\boxed{\nabla \times \mathbf{H} = \mathbf{J}_f} \tag{A.97}$$

$$\boxed{\mathbf{H} \equiv \frac{1}{\mu_0}\mathbf{B} - \mathbf{M}} \tag{A.98}$$

A.3 Electrodynamics

Define **emf** as the closed line integral of force per charge

$$\mathcal{E} \equiv \oint_L \mathbf{f} \cdot d\mathbf{L} \tag{A.99}$$

$$\Phi_B \equiv \int_S \mathbf{B} \cdot d\mathbf{S} \tag{A.100}$$

$$d\Phi_B = \Phi_B(t + dt) - \Phi_B(t) = \int_{dS} \mathbf{B} \cdot d\mathbf{S} \tag{A.101}$$

$$d\mathbf{S} = [\mathbf{v}dt] \times d\mathbf{L} \tag{A.102}$$

$$\frac{d\Phi_B}{dt} = \oint_L \mathbf{B} \cdot [\mathbf{v} \times d\mathbf{L}] \tag{A.103}$$

$$\frac{d\Phi_B}{dt} = -\oint_L [\mathbf{v} \times \mathbf{B}] \cdot d\mathbf{L} = -\oint_L \mathbf{f}_{mag} \cdot d\mathbf{L} = -\mathcal{E} \tag{A.104}$$

$$\mathcal{E} = -\frac{d\Phi_B}{dt} \tag{A.105}$$

Faraday hypothesized a change in magnetic flux induces an electric field emf

$$\boxed{\oint_L \mathbf{E} \cdot d\mathbf{L} = -\frac{d\Phi_B}{dt}.} \tag{A.106}$$

This is Faraday's law in integral form. Expand the definition for magnetic flux

$$\oint_L \mathbf{E} \cdot d\mathbf{L} = -\frac{d}{dt}\int_S \mathbf{B} \cdot d\mathbf{S} \tag{A.107}$$

apply the curl theorem and swap the order of differentiation and integration

$$\int_S [\nabla \times \mathbf{E}] \cdot d\mathbf{S} = -\int_S \frac{\partial \mathbf{B}}{\partial t} \cdot d\mathbf{S} \tag{A.108}$$

This equation has to hold for any surface S, so the integrands must be equal

$$\boxed{\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}.} \tag{A.109}$$

This is the differential form of Faraday's law.

Maxwell recognized in electrodynamics the divergence of Ampere's law doesn't necessarily equal zero because their are no constraints on $\nabla \cdot \mathbf{J}$

$$\nabla \cdot [\nabla \times \mathbf{B}] = \mu_0 \nabla \cdot \mathbf{J} \tag{A.110}$$

$$\mathbf{0} \neq \mu_0 \nabla \cdot \mathbf{J} \tag{A.111}$$

Maxwell used the continuity equation and Gauss's law

$$\nabla \cdot \mathbf{J} = -\frac{\partial \rho}{\partial t} = -\frac{\partial}{\partial t} \nabla \cdot [\epsilon_0 \mathbf{E}] \tag{A.112}$$

to propose a consistent modification to Ampere's law on theoretical grounds

$$\boxed{\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \epsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t}.} \tag{A.113}$$

This modification explains observation and couples the electric and magnetic fields.

A.3.1 Maxwell's Equations

The differential form of Maxwell's equations is

$$\boxed{\begin{cases} \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \epsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t} \\ \nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \\ \nabla \cdot \mathbf{B} = 0 \end{cases} \quad \text{BCs:} \begin{cases} \hat{\mathbf{n}} \times \Delta \mathbf{E} = \mathbf{0} \\ \hat{\mathbf{n}} \times \Delta \mathbf{B} = \mu_0 \mathbf{K} \\ \hat{\mathbf{n}} \cdot \Delta \mathbf{E} = \frac{\sigma}{\epsilon_0} \\ \hat{\mathbf{n}} \cdot \Delta \mathbf{B} = 0 \end{cases}} \tag{A.114}$$

and the electromagnetic (Lorentz) force law is

$$\boxed{\mathbf{f} = \mathbf{E} + \mathbf{v} \times \mathbf{B}.} \tag{A.115}$$

A.3.2 Maxwell's Equations in matter

Electrodynamics in matter requires considering current of changing polarization density

$$\boxed{\mathbf{J}_p \equiv \frac{\partial \mathbf{P}}{\partial t}} \tag{A.116}$$

$$\mathbf{J} = \mathbf{J}_f + \mathbf{J}_b + \mathbf{J}_p = \mathbf{J}_f + \nabla \times \mathbf{M} + \frac{\partial \mathbf{P}}{\partial t} \tag{A.117}$$

$$\nabla \times \mathbf{B} = \mu_0 \left[\mathbf{J}_f + \nabla \times \mathbf{M} + \frac{\partial \mathbf{P}}{\partial t} \right] + \epsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t} \tag{A.118}$$

$$\nabla \times \left[\frac{1}{\mu_0} \mathbf{B} - \mathbf{M} \right] = \mathbf{J}_f + \frac{\partial}{\partial t} [\epsilon_0 \mathbf{E} + \mathbf{P}] \tag{A.119}$$

$$\nabla \times \mathbf{H} = \mathbf{J}_f + \frac{\partial \mathbf{D}}{\partial t} \tag{A.120}$$

$$\boxed{\begin{cases} \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \times \mathbf{H} = \mathbf{J}_f + \frac{\partial \mathbf{D}}{\partial t} \\ \nabla \cdot \mathbf{D} = \rho_f \\ \nabla \cdot \mathbf{B} = 0 \end{cases} \quad \text{BCs:} \quad \begin{cases} \hat{\mathbf{n}} \times \Delta \mathbf{E} = \mathbf{0} \\ \hat{\mathbf{n}} \times \Delta \mathbf{H} = \mathbf{K}_f \\ \hat{\mathbf{n}} \cdot \Delta \mathbf{D} = \sigma_f \\ \hat{\mathbf{n}} \cdot \Delta \mathbf{B} = 0 \end{cases}} \tag{A.121}$$

Auxiliary fields

$$\boxed{\mathbf{D} \equiv \epsilon_0 \mathbf{E} + \mathbf{P}} \tag{A.122}$$

$$\boxed{\mathbf{H} \equiv \frac{1}{\mu_0} \mathbf{B} - \mathbf{M}} \tag{A.123}$$

A.3.3 Symmetric Maxwell's Equations in Matter

see forward model

$$\boxed{\begin{cases} \nabla \times \mathbf{E} = -\mathbf{J}_{mf} - \frac{\partial \mathbf{B}}{\partial t} \\ \nabla \times \mathbf{H} = \mathbf{J}_{ef} + \frac{\partial \mathbf{D}}{\partial t} \\ \nabla \cdot \mathbf{D} = \rho_{ef} \\ \nabla \cdot \mathbf{B} = \rho_{mf} \end{cases} \quad \text{BCs:} \begin{cases} \hat{\mathbf{n}} \times \Delta\mathbf{E} = -\mathbf{K}_{mf} \\ \hat{\mathbf{n}} \times \Delta\mathbf{H} = \mathbf{K}_{ef} \\ \hat{\mathbf{n}} \cdot \Delta\mathbf{D} = \sigma_{ef} \\ \hat{\mathbf{n}} \cdot \Delta\mathbf{B} = \sigma_{mf} \end{cases}} \tag{A.124}$$

in (nonmagentoelectric) linear matter

$$\boxed{\begin{cases} \nabla \times \mathbf{E} = -\mathbf{J}_{mf} - \frac{\partial}{\partial t}[\mu\mathbf{H}] \\ \nabla \times \mathbf{H} = \mathbf{J}_{ef} + \frac{\partial}{\partial t}[\epsilon\mathbf{E}] \\ \nabla \cdot [\epsilon\mathbf{E}] = \rho_{ef} \\ \nabla \cdot [\mu\mathbf{H}] = \rho_{mf} \end{cases} \quad \text{BCs:} \begin{cases} \hat{\mathbf{n}} \times \Delta\mathbf{E} = -\mathbf{K}_{mf} \\ \hat{\mathbf{n}} \times \Delta\mathbf{H} = \mathbf{K}_{ef} \\ \hat{\mathbf{n}} \cdot \Delta[\epsilon\mathbf{E}] = \sigma_{ef} \\ \hat{\mathbf{n}} \cdot \Delta[\mu\mathbf{H}] = \sigma_{mf} \end{cases}} \tag{A.125}$$

A.3.4 Time Harmonic Fields

see forward model

$$\boxed{\begin{cases} \nabla \times \tilde{\mathbf{E}} = -\tilde{\mathbf{J}}_{mi} - j\omega\tilde{\mu}\tilde{\mathbf{H}} \\ \nabla \times \tilde{\mathbf{H}} = \tilde{\mathbf{J}}_{ei} + j\omega\tilde{\epsilon}\tilde{\mathbf{E}} \\ \nabla \cdot [\tilde{\epsilon}\tilde{\mathbf{E}}] = \tilde{\rho}_{ei} \\ \nabla \cdot [\tilde{\mu}\tilde{\mathbf{H}}] = \tilde{\rho}_{mi} \end{cases} \quad \text{BCs:} \begin{cases} \hat{\mathbf{n}} \times \Delta\tilde{\mathbf{E}} = -\tilde{\mathbf{K}}_{mi} \\ \hat{\mathbf{n}} \times \Delta\tilde{\mathbf{H}} = \tilde{\mathbf{K}}_{ei} \\ \hat{\mathbf{n}} \cdot \Delta[\tilde{\epsilon}\tilde{\mathbf{E}}] = \tilde{\sigma}_{ei} \\ \hat{\mathbf{n}} \cdot \Delta[\tilde{\mu}\tilde{\mathbf{H}}] = \tilde{\sigma}_{mi} \end{cases}} \tag{A.126}$$

note in em review that in time harmonic formalism, dot product is no longer an inner product, but an instruction of how to combine harmonic quantities dot products

(and cross products) are rarely applied. carefully consider meaning every time dot or cross product is applied. grad, div, and curl are real operators applied to complex values, so act normally even on time harmonic quantities when in doubt, remember the physical situation is to take the real part of all fields and then applying operations

A.4 Theorems

A.4.1 Uniqueness

Time Harmonic Fields in lossy materials (take limit in lossless case)

$$\begin{cases} \nabla \times \mathbf{E}_1 = -\mathbf{J}_{mi} - j\omega\mu\mathbf{H}_1 \\ \nabla \times \mathbf{H}_1 = \mathbf{J}_{ei} + j\omega\epsilon\mathbf{E}_1 \end{cases} \qquad \begin{cases} \nabla \times \mathbf{E}_2 = -\mathbf{J}_{mi} - j\omega\mu\mathbf{H}_2 \\ \nabla \times \mathbf{H}_2 = \mathbf{J}_{ei} + j\omega\epsilon\mathbf{E}_2 \end{cases} \tag{A.127}$$

$$\begin{cases} \nabla \times \Delta\mathbf{E} = -j\omega\mu\Delta\mathbf{H} \\ \nabla \times \Delta\mathbf{H} = j\omega\epsilon\Delta\mathbf{E} \end{cases} \tag{A.128}$$

$$\nabla \cdot [\Delta\mathbf{E} \times \Delta\mathbf{H}^*] = -j\omega\mu^* |\Delta\mathbf{H}|^2 + j\omega\epsilon |\Delta\mathbf{E}|^2 \tag{A.129}$$

$$\int_V \mathfrak{Re}(\nabla \cdot [\Delta\mathbf{E} \times \Delta\mathbf{H}^*])dV = \int_V \mathfrak{Re}(-j\omega\mu^* |\Delta\mathbf{H}|^2 + j\omega\epsilon |\Delta\mathbf{E}|^2)dV \tag{A.130}$$

$\tilde{\epsilon} \equiv \epsilon + \kappa_e/[j\omega]$ and $\tilde{\mu} \equiv \mu + \kappa_m/[j\omega]$

$$\oint_S \mathfrak{Re}(\Delta\mathbf{E} \times \Delta\mathbf{H}^*) \cdot d\mathbf{S} = -\int_V [\kappa_m |\Delta\mathbf{H}|^2 + \kappa_e |\Delta\mathbf{E}|^2]dV \tag{A.131}$$

i feel like if i took the $\mathfrak{Im}$ instead this would always hold even in lossless media...

1) $\hat{\mathbf{n}} \times \mathbf{E}$ is specified on S, i.e. $\hat{\mathbf{n}} \times \Delta\mathbf{E} = \mathbf{0}$ on S

2) $\hat{\mathbf{n}} \times \mathbf{H}$ is specified on S, i.e. $\hat{\mathbf{n}} \times \Delta\mathbf{H} = \mathbf{0}$ on S

3) S is partitioned into S_i each of which has either $\hat{\mathbf{n}} \times \mathbf{E}$ or $\hat{\mathbf{n}} \times \mathbf{H}$ specified

Note, Maxwell's equations are not linearly independent, and neither are the BCs. By knowing the tangential components of the fields on a surface, the normal components can be deduced (how? continuity equation?).

A.4.2 Duality

Electromagnetic equations are symmetric in electric and magnetic quantities. Given any electromagnetic equation, a dual equation is derived by making the substitutions

$\downarrow$	ϵ	χ_e	κ_e	ρ_e	$\mathbf{J}_e$	$\mathbf{E}$	$\mathbf{D}$	$\mathbf{P}$	ϕ_e	$\mathbf{A}_e$
	μ	χ_m	κ_m	ρ_m	$\mathbf{J}_m$	$\mathbf{H}$	$\mathbf{B}$	$\mu_0\mathbf{M}$	ϕ_m	$\mathbf{A}_m$
$\downarrow$	μ	χ_m	κ_m	ρ_m	$\mathbf{J}_m$	$\mathbf{H}$	$\mathbf{B}$	$\mathbf{M}$	ϕ_m	$\mathbf{A}_m$
	ϵ	χ_e	κ_e	$-\rho_e$	$-\mathbf{J}_e$	$-\mathbf{E}$	$-\mathbf{D}$	$-\frac{1}{\epsilon_0}\mathbf{P}$	$-\phi_e$	$-\mathbf{A}_e$

(A.132)

A.4.3 Reciprocity

see forward model

note, need to first consider integration volume that encloses both current distributions. then it can be broken into two disjoint integration volumes over currents if desired (and as currently presented).

A.4.4 Image Theory

for PEC/PMC planes

$$\mathbf{r}' = \mathbf{r} - 2\hat{\mathbf{n}}[\hat{\mathbf{n}} \cdot \mathbf{r}] \tag{A.133}$$

$$\mathbf{J}_e(\mathbf{r}) = \mp\mathbf{J}_e(\mathbf{r}') \pm 2\hat{\mathbf{n}}[\hat{\mathbf{n}} \cdot \mathbf{J}_e(\mathbf{r}')] \tag{A.134}$$

$$\mathbf{J}_m(\mathbf{r}) = \pm\mathbf{J}_m(\mathbf{r}') \mp 2\hat{\mathbf{n}}[\hat{\mathbf{n}} \cdot \mathbf{J}_m(\mathbf{r}')] \tag{A.135}$$

for $\mathbf{r} \cdot \hat{\mathbf{n}} < 0$

this is a type of equivalence theorem where the PEC/PMC is replaced by the background medium and the BCs at the interface are satisfied.

A.4.5 Equivalence

see forward model

A.5 Radiation

This appendix derives the fields radiated by arbitrary source current distributions in a homogenous linear isotropic medium.

A.5.1 Symmetric Potentials

Assume linear media.

$$\boxed{\mathbf{E} = \mathbf{E}_e + \mathbf{E}_m} \tag{A.136}$$

$$\boxed{\mathbf{H} = \mathbf{H}_e + \mathbf{H}_m} \tag{A.137}$$

Assume isotropic homogeneous nonmagnetoelectric media.

Assume time harmonic formalism.

No magnetic sources

$$\begin{cases} \nabla \times \mathbf{E}_e = -j\omega\mu\mathbf{H}_e \\ \nabla \times \mathbf{H}_e = \mathbf{J}_{ei} + j\omega\epsilon\mathbf{E}_e \\ \nabla \cdot [\epsilon\mathbf{E}_e] = \rho_{ei} \\ \nabla \cdot [\mu\mathbf{H}_e] = 0 \end{cases} \tag{A.138}$$

$$\mu \mathbf{H}_e = \nabla \times \mathbf{A}_e \tag{A.139}$$

$$\nabla \times [\mathbf{E}_e + j\omega \mathbf{A}_e] = \mathbf{0} \tag{A.140}$$

$$\mathbf{E}_e + j\omega \mathbf{A}_e = -\nabla \phi_e \tag{A.141}$$

$$\mathbf{E}_e = -\nabla \phi_e - j\omega \mathbf{A}_e \tag{A.142}$$

Impose Lorenz gauge

$$\nabla \cdot \mathbf{A}_e + j\omega \epsilon \mu \phi_e = 0 \tag{A.143}$$

$$\phi_e = -\frac{1}{j\omega \epsilon \mu} \nabla \cdot \mathbf{A}_e \tag{A.144}$$

$$\nabla^2 \mathbf{A}_e + k^2 \mathbf{A}_e = -\mu \mathbf{J}_{ei} \tag{A.145}$$

$$\nabla^2 \phi_e + k^2 \phi_e = -\frac{\rho_{ei}}{\epsilon} \tag{A.146}$$

No electric sources

$$\begin{cases} \nabla \times \mathbf{E}_m = -\mathbf{J}_{mi} - j\omega \mu \mathbf{H}_m \\ \nabla \times \mathbf{H}_m = j\omega \epsilon \mathbf{E}_m \\ \nabla \cdot [\epsilon \mathbf{E}_m] = 0 \\ \nabla \cdot [\mu \mathbf{H}_m] = \rho_{mi} \end{cases} \tag{A.147}$$

$$\epsilon \mathbf{E}_m = -\nabla \times \mathbf{A}_m \tag{A.148}$$

$$\nabla \times [\mathbf{H}_m + j\omega \mathbf{A}_m] = 0 \tag{A.149}$$

$$\mathbf{H}_m + j\omega \mathbf{A}_m = -\nabla \phi_m \tag{A.150}$$

$$\mathbf{H}_m = -\nabla \phi_m - j\omega \mathbf{A}_m \tag{A.151}$$

Impose Lorenz gauge

$$\nabla \cdot \mathbf{A}_m + j\omega \epsilon \mu \phi_m = 0 \tag{A.152}$$

$$\nabla^2 \mathbf{A}_m + k^2 \mathbf{A}_m = -\epsilon \mathbf{J}_{mi} \tag{A.153}$$

$$\nabla^2 \phi_m + k^2 \phi_m = -\frac{\rho_{mi}}{\mu} \tag{A.154}$$

A.5.2 Potentials due to Sources

$$\nabla^2 \mathbf{A}_e + k^2 \mathbf{A}_e = -\mu \mathbf{J}_{ei} \tag{A.155}$$

$$\mathbf{J}_{ei} = \hat{\mathbf{z}} J_{e0} \delta(\mathbf{r}) \tag{A.156}$$

$$\nabla^2 A_{ez} + k^2 A_{ez} = -\mu J_{e0} \delta(\mathbf{r}) \tag{A.157}$$

$$\nabla^2 = \frac{1}{r^2} \frac{\partial}{\partial r} r^2 \frac{\partial}{\partial r} \tag{A.158}$$

$$A_{ez} = \frac{f(r)}{r} \tag{A.159}$$

$$\begin{aligned} \nabla^2 \left[\frac{f}{r} \right] &= \frac{1}{r^2} \frac{d}{dr} r^2 \frac{d}{dr} \left[\frac{f}{r} \right] \\ &= \frac{1}{r^2} \frac{d}{dr} r^2 \left[\frac{1}{r} \frac{df}{dr} - \frac{f}{r^2} \right] \\ &= \frac{1}{r^2} \frac{d}{dr} \left[r \frac{df}{dr} - f \right] \\ &= \frac{1}{r^2} \left[\frac{df}{dr} + r \frac{d^2 f}{dr^2} - \frac{df}{dr} \right] \\ &= \frac{1}{r} \frac{d^2 f}{dr^2} \end{aligned} \tag{A.160}$$

$$\frac{1}{r} \frac{d^2 f}{dr^2} + k^2 \frac{f}{r} = -\mu J_{e0} \delta(\mathbf{r}) \tag{A.161}$$

$$\frac{d^2 f}{dr^2} + k^2 f = -\mu J_{e0} r \delta(\mathbf{r}) \tag{A.162}$$

$$f(r) = C_1 e^{-jkr} + C_2 e^{jkr} \tag{A.163}$$

apply radiation BCs

$$f(r) = C_1 e^{-jkr} \tag{A.164}$$

$$A_{ez}(r) = C_1 \frac{1}{r} e^{-jkr} \tag{A.165}$$

$$\nabla \cdot \nabla A_{ez} + k^2 A_{ez} = -\mu J_{e0} \delta(\mathbf{r}) \tag{A.166}$$

$$\oint_S \nabla A_{ez} \cdot d\mathbf{S} + \int_V k^2 A_{ez} dV = -\int_V \mu J_{e0} \delta(\mathbf{r}) dV \tag{A.167}$$

$$\nabla A_{ez} = \hat{\mathbf{r}} \frac{d}{dr} \left[C_1 \frac{1}{r} e^{-jkr} \right] = \hat{\mathbf{r}} C_1 \left[-\frac{1}{r^2} - \frac{jk}{r} \right] e^{-jkr} \tag{A.168}$$

$$\begin{aligned} \int_S \nabla A_{ez} \cdot d\mathbf{S} &= \int_0^{2\pi} \int_0^{\pi} \left[\hat{\mathbf{r}} C_1 \left[-\frac{1}{a^2} - \frac{jk}{a} \right] e^{-jka} \right] \cdot [\hat{\mathbf{r}} a^2 \sin\theta d\theta d\phi] \\ &= \int_0^{2\pi} \int_0^{\pi} C_1[-1 - jka] e^{-jka} \sin\theta d\theta d\phi \\ &= C_1[-1 - jka] e^{-jka} \int_0^{\pi} \sin\theta d\theta \int_0^{2\pi} d\phi \\ &= C_1[-1 - jka] e^{-jka} [2][2\pi] \end{aligned} \tag{A.169}$$

$$\begin{aligned} \int_V k^2 A_{ez} dV &= \int_0^{2\pi} \int_0^{\pi} \int_0^{a} k^2 \left[C_1 \frac{1}{r} e^{-jkr} \right] [r^2 \sin\theta dr d\theta d\phi] \\ &= \int_0^{2\pi} \int_0^{\pi} \int_0^{a} k^2 C_1 r e^{-jkr} \sin\theta dr d\theta d\phi \\ &= C_1 k^2 \int_0^{a} r e^{-jkr} dr \int_0^{\pi} \sin\theta d\theta \int_0^{2\pi} d\phi \\ &= C_1 k^2 \int_0^{a} r e^{-jkr} dr [2][2\pi] \end{aligned} \tag{A.170}$$

$$-\int_V \mu J_{e0} \delta(\mathbf{r}) dV = -\mu J_{e0} \tag{A.171}$$

$$C_1 4\pi[-1 - jka] e^{-jka} + C_1 4\pi k^2 \int_0^{a} r e^{-jkr} dr = -\mu J_{e0} \tag{A.172}$$

$$\lim_{a \to 0} \left[C_1 4\pi[-1 - jka] e^{-jka} + C_1 4\pi k^2 \int_0^{a} r e^{-jkr} dr = -\mu J_{e0} \right] \tag{A.173}$$

$$-C_1 4\pi = -\mu J_{e0} \tag{A.174}$$

$$C_1 = \frac{\mu J_{e0}}{4\pi} \tag{A.175}$$

$$A_{ez} = \mu J_{e0} \frac{1}{4\pi r} e^{-jkr} \tag{A.176}$$

$$\mathbf{A}_e = \mu \mathbf{J}_{e0} \frac{1}{4\pi r} e^{-jkr} \tag{A.177}$$

$$\mathbf{A}_e = \mu \mathbf{J}_{e0} \frac{1}{4\pi R} e^{-jkR} \tag{A.178}$$

$$\mathbf{A}_e = \mu \int_V \frac{1}{4\pi R} e^{-jkR} \mathbf{J}_e dV' \tag{A.179}$$

$$\mathbf{A}_e = \mu \int_V G(\mathbf{r}, \mathbf{r}') \mathbf{J}_e(\mathbf{r}') dV' \tag{A.180}$$

$$G(\mathbf{r},\mathbf{r}') = \frac{1}{4\pi R}e^{-jkR} \tag{A.181}$$

$$\mathbf{A}_m = \epsilon \int_V G(\mathbf{r},\mathbf{r}')_{3D}\mathbf{J}_m(\mathbf{r}')dV' \tag{A.182}$$

A.5.3 Fields due to Potentials

$$\nabla R^n = nR^{n-1}\hat{\mathbf{R}} \tag{A.183}$$

$$\nabla e^{-jkR} = e^{-jkR}\nabla[-jkR] = -jke^{-jkR}\hat{\mathbf{R}} \tag{A.184}$$

$$\begin{aligned}
\mathbf{H}_e &= \frac{1}{\mu}\nabla \times \mathbf{A}_e \\
&= \frac{1}{\mu}\nabla \times \left[\mu \int_V G\mathbf{J}_e dV'\right] \\
&= \int_V \nabla G \times \mathbf{J}_e dV' \\
&= \int_V \nabla \left[\frac{1}{4\pi R}e^{-jkR}\right] \times \mathbf{J}_e dV'
\end{aligned} \tag{A.185}$$

$$\begin{aligned}
\nabla[[4\pi R]^{-1}e^{-jkR}] &= [4\pi]^{-1}[[\nabla R^{-1}]e^{-jkR} + R^{-1}[\nabla e^{-jkR}]] \\
&= [4\pi]^{-1}[[-R^{-2}\hat{\mathbf{R}}]e^{-jkR} + R^{-1}[-jke^{-jkR}\hat{\mathbf{R}}]] \\
&= -[4\pi]^{-1}[jkR^{-1} + R^{-2}]e^{-jkR}\hat{\mathbf{R}} \\
&= -G_0\hat{\mathbf{R}}
\end{aligned} \tag{A.186}$$

$$\boxed{G_0 \equiv [4\pi]^{-1}[jkR^{-1} + R^{-2}]e^{-jkR}} \tag{A.187}$$

$$\boxed{\mathbf{H}_e(\mathbf{r}) = -\int_V G_0\hat{\mathbf{R}} \times \mathbf{J}_e(\mathbf{r}')dV'} \tag{A.188}$$

$$\mathbf{E}_e = \frac{1}{j\omega\epsilon}[\nabla \times \mathbf{H}_e - \mathbf{J}_e] \tag{A.189}$$

$$\begin{aligned}\nabla \times \mathbf{H}_e &= \nabla \times \left[- \int_V G_0 \hat{\mathbf{R}} \times \mathbf{J}_e dV' \right] \\ &= - \int_V \nabla \times [G_0 \hat{\mathbf{R}} \times \mathbf{J}_e] dV' \\ &= - \int_V [[\nabla [G_0 R^{-1}]] \times [\mathbf{R} \times \mathbf{J}_e] + G_0 R^{-1} \nabla \times [\mathbf{R} \times \mathbf{J}_e]] dV'\end{aligned} \tag{A.190}$$

$$\begin{aligned}\nabla [G_0 R^{-1}] &= \nabla [[4\pi]^{-1} [jkR^{-2} + R^{-3}] e^{-jkR}] \\ &= [4\pi]^{-1} \nabla [[jkR^{-2} + R^{-3}] e^{-jkR}] \\ &= [4\pi]^{-1} [[\nabla [jkR^{-2} + R^{-3}]] e^{-jkR} + [jkR^{-2} + R^{-3}] [\nabla e^{-jkR}]] \\ &= [4\pi]^{-1} [[-j2kR^{-3} \hat{\mathbf{R}} - 3R^{-4} \hat{\mathbf{R}}] e^{-jkR} + [jkR^{-2} + R^{-3}] [-jke^{-jkR} \hat{\mathbf{R}}]] \\ &= [4\pi]^{-1} [-j2kR^{-3} - 3R^{-4} + k^2 R^{-2} - jkR^{-3}] e^{-jkR} \hat{\mathbf{R}} \\ &= [4\pi]^{-1} [k^2 R^{-2} - j3kR^{-3} - 3R^{-4}] e^{-jkR} \hat{\mathbf{R}} \\ &= -[4\pi]^{-1} [-k^2 R^{-1} + j3kR^{-2} + 3R^{-3}] e^{-jkR} R^{-1} \hat{\mathbf{R}} \\ &= -G_1 R^{-1} \hat{\mathbf{R}}\end{aligned} \tag{A.191}$$

$$\boxed{G_1 \equiv [4\pi]^{-1} [-k^2 R^{-1} + j3kR^{-2} + 3R^{-3}] e^{-jkR}} \tag{A.192}$$

$$\begin{aligned}\nabla \times [\mathbf{R} \times \mathbf{J}] &= \mathbf{R} [\nabla \cdot \mathbf{J}] - \mathbf{J} [\nabla \cdot \mathbf{R}] - [\mathbf{R} \cdot \nabla] \mathbf{J} + [\mathbf{J} \cdot \nabla] \mathbf{R} \\ &= -\mathbf{J} [\nabla \cdot \mathbf{R}] + [\mathbf{J} \cdot \nabla] \mathbf{R} \\ &= -\mathbf{J} [3] + [\mathbf{J}] \\ &= -2\mathbf{J}\end{aligned} \tag{A.193}$$

$$\hat{\mathbf{R}} \times [\hat{\mathbf{R}} \times \mathbf{J}] = \hat{\mathbf{R}} [\hat{\mathbf{R}} \cdot \mathbf{J}] - \mathbf{J} [\hat{\mathbf{R}} \cdot \hat{\mathbf{R}}] = \hat{\mathbf{R}} [\hat{\mathbf{R}} \cdot \mathbf{J}] - \mathbf{J} \tag{A.194}$$

$$\begin{aligned}
\nabla \times \mathbf{H}_e &= -\int_V [[-G_1 R^{-1}\hat{\mathbf{R}}] \times [\mathbf{R} \times \mathbf{J}_e] + G_0 R^{-1}[-2\mathbf{J}_e]]dV' \\
&= \int_V [G_1 \hat{\mathbf{R}} \times [\hat{\mathbf{R}} \times \mathbf{J}_e] + 2G_0 R^{-1}\mathbf{J}_e]dV' \\
&= \int_V [G_1[\hat{\mathbf{R}}[\hat{\mathbf{R}} \cdot \mathbf{J}_e] - \mathbf{J}_e] + 2G_0 R^{-1}\mathbf{J}_e]dV' \qquad \text{(A.195)} \\
&= \int_V [[2G_0 R^{-1} - G_1]\mathbf{J}_e + G_1 \hat{\mathbf{R}}[\hat{\mathbf{R}} \cdot \mathbf{J}_e]]dV' \\
&= \int_V [G_2 \mathbf{J}_e + G_1 \hat{\mathbf{R}}[\hat{\mathbf{R}} \cdot \mathbf{J}_e]]dV'
\end{aligned}$$

$$\boxed{G_2 \equiv [4\pi]^{-1}[k^2 R^{-1} - jkR^{-2} - R^{-3}]e^{-jkR}} \qquad \text{(A.196)}$$

$$\boxed{\mathbf{E}_e(\mathbf{r}) = \frac{1}{j\omega\epsilon}\int_V [G_2 \mathbf{J}_e(\mathbf{r}') + G_1 \hat{\mathbf{R}}[\hat{\mathbf{R}} \cdot \mathbf{J}_e(\mathbf{r}')]]dV' - \frac{1}{j\omega\epsilon}\mathbf{J}_e(\mathbf{r})} \qquad \text{(A.197)}$$

$$\boxed{\begin{aligned}
\mathbf{E}_e(\mathbf{r}) &= \frac{1}{j\omega\epsilon}\int_V [G_2 \mathbf{J}_e(\mathbf{r}') + G_1 \hat{\mathbf{R}}[\hat{\mathbf{R}} \cdot \mathbf{J}_e(\mathbf{r}')]]dV' - \frac{1}{j\omega\epsilon}\mathbf{J}_e(\mathbf{r}) && \text{(A.198a)} \\
\mathbf{E}_m(\mathbf{r}) &= \int_V G_0 \hat{\mathbf{R}} \times \mathbf{J}_m(\mathbf{r}')dV' && \text{(A.198b)} \\
\mathbf{H}_e(\mathbf{r}) &= -\int_V G_0 \hat{\mathbf{R}} \times \mathbf{J}_e(\mathbf{r}')dV' && \text{(A.198c)} \\
\mathbf{H}_m(\mathbf{r}) &= \frac{1}{j\omega\mu}\int_V [G_2 \mathbf{J}_m(\mathbf{r}') + G_1 \hat{\mathbf{R}}[\hat{\mathbf{R}} \cdot \mathbf{J}_m(\mathbf{r}')]]dV' - \frac{1}{j\omega\mu}\mathbf{J}_m(\mathbf{r}) && \text{(A.198d)}
\end{aligned}}$$

where

$$\boxed{\begin{aligned}
G_0 &\equiv [4\pi]^{-1}[jkR^{-1} + R^{-2}]e^{-jkR} && \text{(A.199a)} \\
G_1 &\equiv [4\pi]^{-1}[-k^2 R^{-1} + j3kR^{-2} + 3R^{-3}]e^{-jkR} && \text{(A.199b)} \\
G_2 &\equiv [4\pi]^{-1}[k^2 R^{-1} - jkR^{-2} - R^{-3}]e^{-jkR} && \text{(A.199c)}
\end{aligned}}$$

Sommerfeld Radiation

$$\lim_{r\to\infty} r[\mathbf{E} + \eta\hat{\mathbf{r}} \times \mathbf{H}] = 0 \qquad \text{(A.200)}$$

$$\lim_{r\to\infty} r[\mathbf{H} - \eta^{-1}\hat{\mathbf{r}} \times \mathbf{E}] = 0 \qquad \text{(A.201)}$$

where $\eta = \sqrt{\mu/\epsilon}$

Appendix B

Computer Graphics Review

B.1 Projective Coordinates

Computer graphics relies on **projective (homogeneous) coordinates** for efficiently transforming geometry. As the name suggests, projective coordinates represent geometry in $\mathbb{R}^n$ as the projection of higher dimensional geometry in $\mathbb{R}^{n+1}$, in a way like hands making shadow puppets on a wall.

A position vector $\mathbf{r} \in \mathbb{R}^3$ can be represented as a column vector of 3 coordinates w.r.t the standard basis. Equivalently it can be represented in projective coordinates as a column vector in $\mathbb{R}^4$ with the same 3 coordinates and the last equal to 1

$$\boxed{\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Leftrightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}. \tag{B.1}$$

Projective position vectors are defined to be similar if they are scalar multiples

$$\boxed{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \sim \begin{bmatrix} hx \\ hy \\ hz \\ h \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ h' \end{bmatrix} \quad \text{for} \quad h \neq 0.} \tag{B.2}$$

Direction vector $\mathbf{n} \in \mathbb{R}^3 - \{\mathbf{0}\}$ can also be represented in projective coordinates by

setting the last coordinate to 0

$$\boxed{\mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \Leftrightarrow \begin{bmatrix} n_x \\ n_y \\ n_z \\ 0 \end{bmatrix} \sim \begin{bmatrix} hn_x \\ hn_y \\ hn_z \\ 0 \end{bmatrix} = \begin{bmatrix} n'_x \\ n'_y \\ n'_z \\ 0 \end{bmatrix} \quad \text{for} \quad h \neq 0, \quad \mathbf{n} \in \mathbb{R}^3 - \{\mathbf{0}\}} \tag{B.3}$$

Projective coordinates make explicit the difference between position and direction vectors.

B.2 Transformation

The most general linear transformation from projective coordinates to projective coordinates is the linear system of equations

$$\begin{cases} x' = M_{11}x + M_{12}y + M_{13}z + M_{14}h \\ y' = M_{21}x + M_{22}y + M_{23}z + M_{24}h \\ z' = M_{31}x + M_{32}y + M_{33}z + M_{34}h \\ h' = M_{41}x + M_{42}y + M_{43}z + M_{44}h \end{cases} . \tag{B.4}$$

In matrix notation this is written as matrix multiplying vector

$$\begin{bmatrix} x' \\ y' \\ z' \\ h' \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix} \tag{B.5}$$

or more compactly as

$$\mathbf{r}' = \mathbf{M}\mathbf{r} \tag{B.6}$$

where

$$\boxed{\mathbf{M} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix}}. \tag{B.7}$$

Matrix $\mathbf{M}$ can transform many vectors at once by multiplying an augmented matrix constructed from those vectors

$$\boxed{\begin{bmatrix} \mathbf{r}'_1 & \mathbf{r}'_1 & \dots & \mathbf{r}'_n \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \dots & \mathbf{r}_n \end{bmatrix}}. \tag{B.8}$$

Normal vectors are vectors orthogonal to a plane. They are represented by projective direction vectors, but transform differently than position vectors. Consider a plane such that position vector $\mathbf{r}$ is in the plane and normal vector $\mathbf{n}$ is orthogonal to the plane

$$\mathbf{r} \cdot \mathbf{n} = 0. \tag{B.9}$$

When $\mathbf{r}$ is transformed by matrix $\mathbf{M}$, $\mathbf{n}$ must be transformed by matrix $\mathbf{N}$ to ensure the transformed tangent and normal spaces stay orthogonal

$$[\mathbf{Mr}] \cdot [\mathbf{Nn}] = 0 \tag{B.10}$$

or in matrix notation

$$[\mathbf{Nn}]^T [\mathbf{Mr}] = \mathbf{n}^T \mathbf{N}^T \mathbf{Mr} = 0. \tag{B.11}$$

Assuming $\mathbf{M}$ is invertible, this is satisfied when $\mathbf{N}^T = \mathbf{M}^{-1}$ because of Eq. B.9. Thus when position vectors are transformed by matrix $\mathbf{M}$, normal vectors are transformed by matrix $\mathbf{N} = [\mathbf{M}^{-1}]^T$

$$\boxed{\mathbf{n}' = \mathbf{Nn} \quad \text{where} \quad \mathbf{N} = [\mathbf{M}^{-1}]^T}. \tag{B.12}$$

B.3 Transformation Library

Only a small number of linear transformations on projective coordinates are needed to synthesize a computer graphics pipeline; scaling, rotation, and translation transformations are useful for posing geometry; windowing transformations map between axis-aligned rectangular regions; and orthographic and perspective transformations determine the camera model. Complicated linear transformation can be synthesized by composing these basic transformations. When this is impossible, a general linear transformation can always be supplied.

The following sections are derived for projective position vectors, but the results also apply to projective direction vectors. Projective normal vectors transform per B.12

B.3.1 Scaling

Scaling transformations scale (stretch/compress) axis-aligned coordinates. This is useful to specify the size of things. The matrix representation is easy to work out

$$\begin{bmatrix} S_x x \\ S_y y \\ S_z z \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{B.13}$$

where the scaling factors S_x, S_y, S_z scale their respective coordinates x, y, z. Isolating the transformation matrix define the projection scaling matrix

$$\boxed{\mathbf{M}_{\text{scale3D}}(S_x, S_y, S_z) \equiv \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}. \tag{B.14}$$

This matrix is invertible if the scaling factors are non-zero.

B.3.2 Rotation

Rotation transformations rotate coordinates about an axis through the origin. There are a variety of ways to parameterize a rotation in matrix notation. Regardless of the parameterization, the rotation matrix elements can always be written as

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad \text{where} \quad \det(\mathbf{R}) = 1 \tag{B.15}$$

therefore the projective rotation matrix is of the form

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{B.16}$$

The axis-angle representation is developed here for its robustness and simplicity. Consider the Rodrigues rotation formula, which describes the rotation of point $\mathbf{r} \in \mathbb{R}^3$ about unit vector $\hat{\mathbf{n}}$ by angle θ with the coordinate-free equation

$$\mathbf{r}' = \mathbf{r}_{\parallel} + \mathbf{r}_{\perp} \cos\theta + \hat{\mathbf{n}} \times \mathbf{r}_{\perp} \sin\theta \tag{B.17}$$

where $\mathbf{r}$ has been decomposed into components parallel and orthogonal to the axis. The parallel component $\mathbf{r}_{\parallel}$ in terms of the dot product is

$$\mathbf{r}_{\parallel} = \hat{\mathbf{n}}(\hat{\mathbf{n}} \cdot \mathbf{r}) = \hat{\mathbf{n}}\hat{\mathbf{n}}^T\mathbf{r} \tag{B.18}$$

while the orthogonal component $\mathbf{r}_{\perp}$ is

$$\mathbf{r}_{\perp} = \mathbf{r} - \mathbf{r}_{\parallel} = \mathbf{r} - \hat{\mathbf{n}}\hat{\mathbf{n}}^T\mathbf{r} = (\mathbf{1} - \hat{\mathbf{n}}\hat{\mathbf{n}}^T)\mathbf{r}. \tag{B.19}$$

The cross product (a bilinear operation) is linear if one of the vectors is held constant, and thus a cross product by this constant vector can be written as a matrix equation

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \mathbf{v} = \mathbf{M}_{\mathbf{u}\times}\mathbf{v}. \tag{B.20}$$

The rotation formula in matrix notation is thus

$$\mathbf{r}' = \hat{\mathbf{n}}\hat{\mathbf{n}}^T\mathbf{r} + (\mathbf{1} - \hat{\mathbf{n}}\hat{\mathbf{n}}^T)\mathbf{r}\cos\theta + \mathbf{M}_{\hat{\mathbf{n}}\times}(\mathbf{1} - \hat{\mathbf{n}}\hat{\mathbf{n}}^T)\mathbf{r}.\sin\theta \tag{B.21}$$

Grouping the terms

$$\mathbf{r}' = (\hat{\mathbf{n}}\hat{\mathbf{n}}^T(1 - \cos\theta) + \mathbf{1}\cos\theta + \mathbf{M}_{\hat{\mathbf{n}}\times}\sin\theta + \mathbf{0})\mathbf{r} \tag{B.22}$$

and Explicitly write out the matrix elements

$$\mathbf{R} = \begin{bmatrix} n_x n_x & n_x n_y & n_x n_z \\ n_y n_x & n_y n_y & n_y n_z \\ n_z n_x & n_z n_y & n_z n_z \end{bmatrix} (1 - \cos\theta) + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cos\theta + \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix} \sin\theta \tag{B.23}$$

and combine terms to get a rotation matrix function parameterized by an axis specified by a normal vector and an angle θ to rotate by

$$\mathbf{R}(\hat{\mathbf{n}}, \theta) = \begin{bmatrix} n_x^2 C + \cos\theta & n_x n_y C - n_z \sin\theta & n_x n_z C + n_y \sin\theta \\ n_y n_x C + n_z \sin\theta & n_y^2 C + \cos\theta & n_y n_z C - n_x \sin\theta \\ n_z n_x C - n_y \sin\theta & n_z n_y C + n_x \sin\theta & n_z^2 C + \cos\theta \end{bmatrix} \tag{B.24}$$

$$\boxed{C = 1 - \cos\theta} \tag{B.25}$$

Embedding the 3D rotation matrix into a 4D matrix, define

$$\boxed{\mathbf{M}_{\text{rot3D}}(\hat{\mathbf{n}}, \theta) = \begin{bmatrix} n_x^2 C + \cos\theta & n_x n_y C - n_z \sin\theta & n_x n_z C + n_y \sin\theta & 0 \\ n_y n_x C + n_z \sin\theta & n_y^2 C + \cos\theta & n_y n_z C - n_x \sin\theta & 0 \\ n_z n_x C - n_y \sin\theta & n_z n_y C + n_x \sin\theta & n_z^2 C + \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \tag{B.26}$$

B.3.3 Translation

Translation transformations translate coordinates. A major benefit of homogeneous coordinates is the ability to encode translations with linear transformations. Translations take the form

$$\begin{bmatrix} x + \Delta x \\ y + \Delta y \\ z + \Delta z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{B.27}$$

Isolating the transformation matrix, define

$$\boxed{\mathbf{M}_{\text{trans3D}}(\Delta x, \Delta y, \Delta z) \equiv \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}} \tag{B.28}$$

B.3.4 Windowing

Windowing transformations map one axis-aligned rectangular region to another axis-aligned rectangular region. This transformation is used to build projection matrices, and map output to pixels.

The windowing transformation is constructed from a sequence of simpler transformations; first translate the starting region to the origin, then scale the region to reshape it, and finally translate the region to the final position. Specifically (in 2D)

$$\mathbf{M}_{\text{win2D}} = \mathbf{M}_{\text{trans2D}}(x'_0, y'_0)\mathbf{M}_{\text{scale2D}}\left(\frac{x'_1 - x'_0}{x_1 - x_0}, \frac{y'_1 - y'_0}{y_1 - y_0}\right)\mathbf{M}_{\text{trans2D}}(-x_0, -y_0) \tag{B.29}$$

Expanding the matrix definitions and simplifying

$$\mathbf{M}_{\text{win2D}} = \begin{bmatrix} 1 & 0 & x'_0 \\ 0 & 1 & y'_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x'_1 - x'_0}{x_1 - x_0} & 0 & 0 \\ 0 & \frac{y'_1 - y'_0}{y_1 - y_0} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.30}$$

$$\mathbf{M}_{\text{win2D}} = \begin{bmatrix} 1 & 0 & x'_0 \\ 0 & 1 & y'_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x'_1 - x'_0}{x_1 - x_0} & 0 & -x_0\frac{x'_1 - x'_0}{x_1 - x_0} \\ 0 & \frac{y'_1 - y'_0}{y_1 - y_0} & -y_0\frac{y'_1 - y'_0}{y_1 - y_0} \\ 0 & 0 & 1 \end{bmatrix} \tag{B.31}$$

$$\mathbf{M}_{\text{win2D}} = \begin{bmatrix} \frac{x_1'-x_0'}{x_1-x_0} & 0 & x_0' - x_0\frac{x_1'-x_0'}{x_1-x_0} \\ 0 & \frac{y_1'-y_0'}{y_1-y_0} & y_0' - y_0\frac{y_1'-y_0'}{y_1-y_0} \\ 0 & 0 & 1 \end{bmatrix} \tag{B.32}$$

results in

$$\mathbf{M}_{\text{win2D}} = \begin{bmatrix} \frac{x_1'-x_0'}{x_1-x_0} & 0 & \frac{x_0'x_1-x_0x_1'}{x_1-x_0} \\ 0 & \frac{y_1'-y_0'}{y_1-y_0} & \frac{y_0'y_1-y_0y_1'}{y_1-y_0} \\ 0 & 0 & 1 \end{bmatrix} \tag{B.33}$$

Generalizing to 3D, define

$$\boxed{\mathbf{M}_{\text{win3D}}(x_0, x_1, y_0, y_1, z_0, z_1, x_0', x_1', y_0', y_1', z_0', z_1') \equiv \begin{bmatrix} \frac{x_1'-x_0'}{x_1-x_0} & 0 & 0 & \frac{x_0'x_1-x_0x_1'}{x_1-x_0} \\ 0 & \frac{y_1'-y_0'}{y_1-y_0} & 0 & \frac{y_0'y_1-y_0y_1'}{y_1-y_0} \\ 0 & 0 & \frac{z_1'-z_0'}{z_1-z_0} & \frac{z_0'z_1-z_0z_1'}{z_1-z_0} \\ 0 & 0 & 0 & 1 \end{bmatrix}} \tag{B.34}$$

B.3.5 Orthographic Projection

Orthographic projection is a special windowing transformation that maps an axis-aligned rectangular region of world space to the unit cube

$$\mathbf{M}_{\text{ortho}} = \mathbf{M}_{\text{win3D}}(l, r, b, t, n, f, -1, 1, -1, 1, -1, 1) \tag{B.35}$$

Evaluating this equation, define

$$\boxed{\mathbf{M}_{\text{ortho}}(l, r, b, t, n, f) \equiv \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}} \tag{B.36}$$

B.3.6 Perspective Projection

Perspective projection models a pinhole camera.

Consider 2D where z is depth. In this case, x needs to be divided by z.

$$\begin{cases} x_s = \frac{nx_r}{z_r} \\ z_s = A + \frac{B}{z_r} \\ h_s = 1 \end{cases} \tag{B.37}$$

$$\begin{cases} n = A + \frac{B}{n} \\ f = A + \frac{B}{f} \end{cases} \tag{B.38}$$

$$f = A + \frac{B}{f} = \left[n - \frac{B}{n}\right] + \frac{B}{f} = n + B\left[\frac{1}{f} - \frac{1}{n}\right] = n + B\frac{n-f}{nf} \tag{B.39}$$

$$B = -nf \tag{B.40}$$

$$A = n + f \tag{B.41}$$

$$\begin{cases} x_s = \frac{nx_r}{z_r} \\ z_s = n + f - \frac{nf}{z_r} \\ h_s = 1 \end{cases} \tag{B.42}$$

$$\begin{cases} x_q = nx_r \\ z_q = z_r[n+f] - nf \\ h_q = z_r \end{cases} \tag{B.43}$$

$$\begin{bmatrix} x_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{x_q}{h_q} \\ \frac{z_q}{h_q} \\ 1 \end{bmatrix} \sim \begin{bmatrix} x_q \\ z_q \\ h_q \end{bmatrix} = \begin{bmatrix} n & 0 & 0 \\ 0 & n+f & -nf \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_r \\ z_r \\ 1 \end{bmatrix} \tag{B.44}$$

$$\mathbf{s} \sim \mathbf{q} = \mathbf{M}_{\text{perspect2D}}\mathbf{r} \tag{B.45}$$

$$\mathbf{M}_{\text{frustum2D}} = \mathbf{M}_{\text{ortho2D}}(l, r, n, f)\mathbf{M}_{\text{perspect2D}}(n, f) \tag{B.46}$$

$$\mathbf{M}_{\text{frustum2D}} = \begin{bmatrix} \frac{2}{r-l} & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n & 0 & 0 \\ 0 & n+f & -nf \\ 0 & 1 & 0 \end{bmatrix} \tag{B.47}$$

$$\mathbf{M}_{\text{frustum2D}} = \begin{bmatrix} \frac{2n}{r-l} & -\frac{r+l}{r-l} & 0 \\ 0 & \frac{f+n}{f-n} & -\frac{2nf}{f-n} \\ 0 & 1 & 0 \end{bmatrix} \tag{B.48}$$

$$\boxed{\mathbf{M}_{\text{frustum}}(l, r, b, t, n, f) \equiv \begin{bmatrix} \frac{2n}{r-l} & 0 & -\frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & -\frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -\frac{2nf}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix}} \tag{B.49}$$

B.4 Transformation Stack

Scaling, rotation, and translation transformations (in that order) are commonly used to specify the pose (position and orientation) of spaces and geometry.

$$\mathbf{M}_{\text{pose3D}} = \mathbf{M}_{\text{trans3D}}\mathbf{M}_{\text{rot3D}}\mathbf{M}_{\text{scale3D}} \tag{B.50}$$

matrix stack

$$\mathbf{M} = \mathbf{M}_{\text{windowing}}\mathbf{M}_{\text{projection}}\mathbf{M}_{\text{modelview}} \tag{B.51}$$

1) apply modelview 2) compute lighting in eye coordinates 3) apply projection

B.5 Clipping

$$\hat{\mathbf{n}} \cdot \mathbf{q} = d \tag{B.52}$$

give algorithm on how to cut triangle up. can either be pass, fail, one vertex passes, or two vertices pass per face

B.6 Barycentric Coordinates

Let $B = \{\mathbf{r}_0, \ldots, \mathbf{r}_N\}$ be an affinely independent set of $N+1$ vectors in $\mathbb{R}^M$. Consider

$$\mathbf{r} = \sum_{i=0}^{N} \mathbf{r}_i \beta_i \quad \text{where} \quad \sum_{i=0}^{N} \beta_i = 1 \tag{B.53}$$

Use the constraint to eliminate β_0 from the linear combination. First solve for β_0

$$\beta_0 = 1 - \sum_{i=1}^{N} \beta_i \tag{B.54}$$

then insert this result into the linear combination

$$\mathbf{r} = \mathbf{r}_0 \left[1 - \sum_{i=1}^{N} \beta_i \right] + \sum_{i=1}^{N} \mathbf{r}_i \beta_i \tag{B.55}$$

and simplify to get

$$\mathbf{r} - \mathbf{r}_0 = \sum_{i=1}^{N} \left[\mathbf{r}_i - \mathbf{r}_0 \right] \beta_i. \tag{B.56}$$

The sum has the form of matrix-vector multiplication

$$\mathbf{r} - \mathbf{r}_0 = \mathbf{M}\boldsymbol{\beta} \tag{B.57}$$

where the columns of matrix $\mathbf{M}$ are the affine basis vectors

$$\mathbf{M} = \begin{bmatrix} \mathbf{r}_1 - \mathbf{r}_0 & \mathbf{r}_2 - \mathbf{r}_0 & \ldots & \mathbf{r}_N - \mathbf{r}_0 \end{bmatrix} \tag{B.58}$$

and the elements of column vector $\boldsymbol{\beta}$ are the barycentric coefficients

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & \beta_2 & \dots & \beta_N \end{bmatrix}^T . \tag{B.59}$$

In the case where $N = M$, affine independence guarantees matrix $\mathbf{M}$ is invertible, thus $\boldsymbol{\beta}$ has the unique solution

$$\boldsymbol{\beta} = \mathbf{M}^{-1}[\mathbf{r} - \mathbf{r}_0]. \tag{B.60}$$

The case of $N = M = 2$ is of primary importance in this work and fully expanded here. Matrix $\mathbf{M} \in \mathbb{R}^{2\times 2}$

$$\mathbf{M} = \begin{bmatrix} \mathbf{r}_1 - \mathbf{r}_0 & \mathbf{r}_2 - \mathbf{r}_0 \end{bmatrix} . \tag{B.61}$$

The inverse of a 2x2 matrix is

$$\mathbf{M}^{-1} = \frac{1}{\det(\mathbf{M})} \begin{bmatrix} M_{22} & -M_{12} \\ -M_{21} & M_{11} \end{bmatrix} \tag{B.62}$$

so the solution to the matrix equation is

$$\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \frac{1}{(x_1 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_1 - y_0)} \begin{bmatrix} y_2 - y_0 & -(x_2 - x_0) \\ -(y_1 - y_0) & x_1 - x_0 \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} . \tag{B.63}$$

The constraint on the coefficients gives β_0

$$\boxed{\beta_0 = 1 - \beta_1 - \beta_2} \tag{B.64}$$

while applying the inverse matrix gives the equations for β_1

$$\boxed{\beta_1 = \frac{(x - x_0)(y_2 - y_0) - (x_2 - x_0)(y - y_0)}{(x_1 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_1 - y_0)}} \tag{B.65}$$

and β_2

$$\boxed{\beta_2 = \frac{-(x - x_0)(y_1 - y_0) + (x_1 - x_0)(y - y_0)}{(x_1 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_1 - y_0)}}. \tag{B.66}$$

Barycentric coordinates linearly interpolate the properties of vertices. Geometrically the vector sum equals the linear interpolation of the vertex points

$$\boxed{\text{lerp2}(p_0, p_1, p_2, \beta_1, \beta_2) \equiv p_0(1 - \beta_1 - \beta_2) + p_1\beta_1 + p_2\beta_2} \tag{B.67}$$

B.7 Perspective Interpolation

Efficient rasterization requires iterating over pixels in screen coordinates. Transforming vertices into screen coordinates and performing linear interpolation with barycentric coordinates is part of rasterization.

$$\mathbf{r} = \sum_i \mathbf{r}_i \beta_i \tag{B.68}$$

Apply the transformation matrix $\mathbf{M}$ to this equation

$$\mathbf{Mr} = \sum_i \mathbf{Mr}_i \beta_i \tag{B.69}$$

to get clip coordinates

$$\mathbf{q} = \sum_i \mathbf{q}_i \beta_i. \tag{B.70}$$

Clip coordinates are transformed vertices before the projective divide.

Clip coordinates are similar to screen coordinates by the projective divide

$$\frac{\mathbf{q}}{h} = \frac{\sum_i \mathbf{q}_i \beta_i}{\sum_i h_i \beta_i} \tag{B.71}$$

screen coordinates

$$\mathbf{s} = \sum_i \mathbf{s}_i \beta_i' \tag{B.72}$$

$$\frac{\sum_i \mathbf{q}_i \beta_i}{\sum_i h_i \beta_i} = \sum_i \mathbf{s}_i \beta_i' \tag{B.73}$$

$$\sum_i \mathbf{q}_i \beta_i = \sum_i h_i \beta_i \sum_i \mathbf{s}_i \beta_i' \tag{B.74}$$

$$\sum_i \mathbf{s}_i h_i \beta_i = \sum_i h_i \beta_i \sum_i \mathbf{s}_i \beta_i' \tag{B.75}$$

Screen space barycentric coordinates

Consider the world-space linear interpolation of property p

$$p = p_0 + t(p_1 - p_0) \tag{B.76}$$

The line is defined by

$$\mathbf{r} = \mathbf{r}_0 + t(\mathbf{r}_1 - \mathbf{r}_0) \tag{B.77}$$

$$\mathbf{Mr} = \mathbf{Mr}_0 + t(\mathbf{Mr}_1 - \mathbf{Mr}_0) \tag{B.78}$$

$$\mathbf{q} = \mathbf{q}_0 + t(\mathbf{q}_1 - \mathbf{q}_0) \tag{B.79}$$

$$\frac{\mathbf{q}}{h} = \frac{\mathbf{q}_0 + t(\mathbf{q}_1 - \mathbf{q}_0)}{h_0 + t(h_1 - h_0)} \tag{B.80}$$

$$\mathbf{s} = \mathbf{s}_0 + u(\mathbf{s}_1 - \mathbf{s}_0) \tag{B.81}$$

$$\frac{\mathbf{q}_0 + t(\mathbf{q}_1 - \mathbf{q}_0)}{h_0 + t(h_1 - h_0)} = \mathbf{s}_0 + u(\mathbf{s}_1 - \mathbf{s}_0) \tag{B.82}$$

$$\mathbf{q}_0 + t(\mathbf{q}_1 - \mathbf{q}_0) = (h_0 + t(h_1 - h_0))(\mathbf{s}_0 + u(\mathbf{s}_1 - \mathbf{s}_0)) \tag{B.83}$$

$$h_0\mathbf{s}_0 + t(h_1\mathbf{s}_1 - h_0\mathbf{s}_0) = h_0\mathbf{s}_0 + uh_0(\mathbf{s}_1 - \mathbf{s}_0) + t(h_1 - h_0)\mathbf{s}_0 + tu(h_1 - h_0)(\mathbf{s}_1 - \mathbf{s}_0) \tag{B.84}$$

$$th_1(\mathbf{s}_1 - \mathbf{s}_0) - tu(h_1 - h_0)(\mathbf{s}_1 - \mathbf{s}_0) = uh_0(\mathbf{s}_1 - \mathbf{s}_0) \tag{B.85}$$

$$t(h_1 - u(h_1 - h_0)) = uh_0 \tag{B.86}$$

$$t = \frac{uh_0}{h_1 + u(h_0 - h_1)} \tag{B.87}$$

$$p = p_0 + t(p_1 - p_0) \tag{B.88}$$

$$p = p_0 + \frac{uh_0}{h_1 + u(h_0 - h_1)}(p_1 - p_0) \tag{B.89}$$

$$p = \frac{p_0(h_1 + u(h_0 - h_1)) + uh_0(p_1 - p_0)}{h_1 + u(h_0 - h_1)} \tag{B.90}$$

$$p = \frac{p_0 h_1 + u(p_1 h_0 - p_0 h_1)}{h_1 + u(h_0 - h_1)} \tag{B.91}$$

$$p = \frac{\frac{p_0}{h_0} + u\left(\frac{p_1}{h_1} - \frac{p_0}{h_0}\right)}{\frac{1}{h_0} + u\left(\frac{1}{h_1} - \frac{1}{h_0}\right)} \tag{B.92}$$

$u = \beta_1$

$$\boxed{p = \frac{\text{lerp}\left(\frac{p_0}{h_0}, \frac{p_1}{h_1}, \beta_1\right)}{\text{lerp}\left(\frac{1}{h_0}, \frac{1}{h_1}, \beta_1\right)}} \tag{B.93}$$

$$\boxed{p = \frac{\text{lerp2}\left(\frac{p_0}{h_0}, \frac{p_1}{h_1}, \frac{p_2}{h_2}, \beta_1, \beta_2\right)}{\text{lerp2}\left(\frac{1}{h_0}, \frac{1}{h_1}, \frac{1}{h_2}, \beta_1, \beta_2\right)}} \tag{B.94}$$

B.8 Face Rasterization

Algorithm 8: Triangle Rasterization

Result: Rasterize triangle

```
for x ∈ X do
    for y ∈ Y do
        (alpha, beta) = Rect2ToBary2(r0, r1, r2, x, y);
        if α ∈ [0, 1] ∧ β ∈ [0, 1] then
            p = lerp2(p0 / h0, p1 / h1, p2 / h2, alpha, beta);
            h = lerp2(1 / h0, 1 / h1, 1 / h2, alpha, beta);
            draw(x, y, p / h);
        end
    end
end
```

note: this algo doesn't try to fix double draw

B.9 Pipeline

- apply $\mathbf{M}_{\text{modelview}}$
- world space computation (e.g. light, ignored in this work)
- apply $\mathbf{M}_{\text{perspective}}$
- clip
- projective divide
- apply $\mathbf{M}_{\text{viewport}}$
- rasterize

B.10 Mesh Data Structure

$$\mathbf{V} = \begin{bmatrix} \mathbf{r}_0 & \mathbf{r}_1 & \dots & \mathbf{r}_v \end{bmatrix} \tag{B.95}$$

$$\mathbf{F} = \begin{bmatrix} v_{00} & v_{01} & \dots & v_{0f} \\ v_{10} & v_{11} & \dots & v_{1f} \\ v_{20} & v_{21} & \dots & v_{2f} \end{bmatrix} \tag{B.96}$$

NOTE THAT VF MESH SUCKS AT SUBDIVISION!!

Bibliography

[AED+16] Tamer A. A. Alzohairy, Emad El-Dein, et al. Image Mosaicing based on Neural Networks. *International Journal of Computer Applications*, 975(1):8887, feb 2016.

[BL07] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[Boa01] Amir Boag. A fast multilevel domain decomposition algorithm for radar imaging. *IEEE Transactions on Antennas and Propagation*, 49(4):666–671, 2001.

[Bro92] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4):325–376, December 1992.

[BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: speeded up robust features. In *European Conference on Computer Vision*, pages 404–417. Springer, 2006.

[CCT+08] Wai Lam Chan, Kriti Charan, Dharmpal Takhar, Kevin F. Kelly, Richard G. Baraniuk, and Daniel M. Mittleman. A single-pixel terahertz imaging system based on compressed sensing. *Applied Physics Letters*, 93(12):121105, 2008.

[Cre14] Creaform. MaxSHOT-3D optical coordiante measuring system. Brochure, 2014.

[CSKR14] Hyunggi Cho, Young-Woo Seo, B. V. K. Vijaya Kumar, and Raguna-than Raj Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1836–1843. IEEE, 2014.

[Ebe02] David Eberly. Rotation representations and performance issues. Self, 2002.

[FB87] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in Computer Vision*, pages 726–740. Elsevier, 1987.

[FC78] Edward E. Fenimore and Thomas M. Cannon. Coded aperture imaging with uniformly redundant arrays. *Applied optics*, 17(3):337–347, 1978.

[FYI+15] Thomas Fromenteze, Okan Yurduseven, Mohammadreza F. Imani, Jonah Gollub, Cyril Decroze, David Carsenat, and David R. Smith. Computational imaging using a mode-mixing cavity at microwave frequencies. *Applied Physics Letters*, 106(19):194104, 2015.

[GA16] Frank Gumbmann and Sherif Sayed Ahmed. Walk through screening with multistatic mmW technology. In *SPIE Millimetre Wave and Terahertz Sensors and Technology IX*, volume 9993, page 999306. International Society for Optics and Photonics, 2016.

[GAGF17] Raffaele Gravina, Parastoo Alinia, Hassan Ghasemzadeh, and Giancarlo Fortino. Multi-sensor fusion in body sensor networks: state-of-the-art and research challenges. *Information Fusion*, 35:68–80, 2017.

[GWSG11] Daniel Göhring, Miao Wang, Michael Schnürmacher, and Tinosch Ganjineh. Radar/lidar sensor fusion for car-following on highways. In *International Conference on Automation, Robotics and Applications (ICARA)*, pages 407–412. IEEE, 2011.

[GYT+17] J. N. Gollub, O. Yurduseven, K. P. Trofatter, D. Arnitz, M. F. Imani, T. Sleasman, M. Boyarsky, A. Rose, A. Pedross-Engel, H. Odabasi, et al. Large metasurface aperture for millimeter wave computational imaging at the human-scale. *Scientific Reports*, 7:42650, 2017.

[HDM+13] John Hunt, Tom Driscoll, Alex Mrozack, Guy Lipworth, Matthew Reynolds, David Brady, and David R. Smith. Metamaterial apertures for computational imaging. *Science*, 339(6117):310–313, 2013.

[HGD+14] John Hunt, Jonah Gollub, Tom Driscoll, Guy Lipworth, Alex Mrozack, Matthew S. Reynolds, David J. Brady, and David R. Smith. Metamaterial microwave holographic imaging system. *Journal of the Optical Society of America A*, 31(10):2109–2119, 2014.

[HTTM12] Michael B. Holte, Cuong Tran, Mohan M. Trivedi, and Thomas B. Moeslund. Human pose estimation and activity recognition from multi-view videos: comparative explorations of recent developments. *IEEE Journal of selected topics in signal processing*, 6(5):538–552, 2012.

[LCF00] John P. Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 165–172. ACM Press/Addison-Wesley Publishing Co., 2000.

[LCL+14] Qingquan Li, Long Chen, Ming Li, Shih-Lung Shaw, and Andreas Nüchter. A sensor-fusion drivable-region and lane-detection system for

autonomous vehicle navigation in challenging road scenarios. *IEEE Transactions on Vehicular Technology*, 63(2):540–555, 2014.

[LHG+18] Kevin J. Liang, Geert Heilmann, Christopher Gregory, Souleymane O. Diallo, David Carlson, Gregory P. Spell, John B. Sigman, Kris Roe, and Lawrence Carin. Automatic threat recognition of prohibited items at aviation checkpoint with x-ray imaging: a deep learning approach. In *SPIE Anomaly Detection and Imaging with X-Rays (ADIX) III*, volume 10632, page 1063203. International Society for Optics and Photonics, 2018.

[LMH+13] Guy Lipworth, Alex Mrozack, John Hunt, Daniel L. Marks, Tom Driscoll, David Brady, and David R. Smith. Metamaterial apertures for coherent computational imaging on the physical layer. *Journal of the Optical Society of America A*, 30(8):1603–1612, 2013.

[Low99] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. Ieee, 1999.

[LRY+15] Guy Lipworth, Alec Rose, Okan Yurduseven, Vinay R. Gowda, Mohammadreza F. Imani, Hayrettin Odabasi, Parker Trofatter, Jonah Gollub, and David R. Smith. Comprehensive simulation platform for a metamaterial imaging system. *Applied Optics*, 54(31):9343–9353, 2015.

[MA13] V. R. S. Mani and S. Arivazhagan. Survey of medical image registration. *Journal of Biomedical Engineering and Technology*, 1(2):8–25, mar 2013.

[MF12] Andrew Maimone and Henry Fuchs. Reducing interference between multiple structured light depth sensors using motion. In *2012 IEEE Virtual Reality Workshops (VRW)*, pages 51–54. IEEE, 2012.

[MP94] Steve Mann and Rosalind W. Picard. Virtual bellows: Constructing high quality stills from video. In *Proceedings of 1st International Conference on Image Processing*, volume 1, pages 363–367. IEEE, IEEE Comput. Soc. Press, 1994.

[MYS17a] Daniel L. Marks, Okan Yurduseven, and David R. Smith. Cavity-backed metasurface antennas and their application to frequency diversity imaging. *Jornal of the Optical Society of America A*, 34(4):472–480, 2017.

[MYS17b] Daniel L. Marks, Okan Yurduseven, and David R. Smith. Fourier accelerated multistatic imaging: a fast reconstruction algorithm for multiple-input-multiple-output radar imaging. *IEEE Access*, 5:1796–1809, 2017.

[Nay11] Shree K. Nayar. Computational cameras: approaches, benefits and limits. *Columbia University Computer Science Technical Reports*, page 6, 2011.

[OIL+16] Hayrettin Odabasi, Mohammadreza F. Imani, Guy Lipworth, Jonah Gollub, and David R. Smith. Investigation of alignment errors on multistatic microwave imaging based on frequency-diverse metamaterial apertures. *Progress In Electromagnetics Research*, 70:101–112, 2016.

[PEAG+18] Andreas Pedross-Engel, Daniel Arnitz, Jonah N. Gollub, Okan Yurduseven, Kenneth P. Trofatter, Mohammadreza F. Imani, Timothy Sleasman, Michael Boyarsky, Xiaojie Fu, Daniel L. Marks, et al. Orthogonal coded active illumination for millimeter wave, massive-mimo computational imaging with metasurface antennas. *IEEE Transactions on Computational Imaging*, 4(2):184–193, 2018.

[PMFS+16] Laura Pulido-Mancera, Thomas Fromenteze, Timothy Sleasman, Michael Boyarsky, Mohammadreza F. Imani, Matthew Reynolds, and David Smith. Application of range migration algorithms to imaging with a dynamic metasurface antenna. *Journal of the Optical Society of America B*, 33(10):2082–2092, 2016.

[Roh05] F. Rohlf. Procrustes problems by john c. gower; garmt b. dijksterhuis. *Journal of the American Statistical Association*, 100, 01 2005.

[RS16] Rohde and Schwarz. Rohde and Schwarz QPS quick personnel security scanners. Brochure, 2016.

[SAS07] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 357–360. ACM, ACM Press, 2007.

[SBM12] David M Sheen, Bruce Bernacki, and Doug McMakin. Advanced millimeter-wave imaging enhances security screening. *Spie Newsroom*, 12, 2012.

[Sch66] Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[SFIGS15] Timothy Sleasman, Mohammadreza F. Imani, Jonah N. Gollub, and David R. Smith. Dynamic metamaterial aperture for microwave imaging. *Applied Physics Letters*, 107(20):204104, 2015.

[SHM+16] David M. Sheen, Thomas E. Hall, Douglas L. McMakin, A. Mark Jones, and Jonathan R. Tedeschi. Three-dimensional radar imaging techniques

and systems for near-field applications. In *SPIE Radar Sensor Technology XX*, volume 9829. International Society for Optics and Photonics, 2016.

[SIY+17] Timothy Sleasman, Mohammadreza F. Imani, Okan Yurduseven, Kenneth P. Trofatter, Vinay R. Gowda, Daniel L. Marks, Jonah N. Gollub, and David R. Smith. Near field scan alignment procedure for electrically large apertures. *IEEE Transactions on Antennas and Propagation*, 65(6):3257–3262, 2017.

[SLK15] Hamed Sarbolandi, Damien Lefloch, and Andreas Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. *Computer Vision and Image Understanding*, 139:1–20, oct 2015.

[SMC+96] David M. Sheen, Douglas L. McMakin, H. Dale Collins, Thomas E. Hall, and Ronald H. Severtsen. Concealed explosive detection on personnel using a wideband holographic millimeter-wave imaging system. In *SPIE Signal Processing, Sensor Fusion, and Target Recognition V*, volume 2755, pages 503–514. International Society for Optics and Photonics, 1996.

[SMH01] David M. Sheen, Douglas L. McMakin, and Thomas E. Hall. Three-dimensional millimeter-wave imaging for concealed weapon detection. *IEEE Transactions on Microwave Theory and Techniques*, 49(9):1581–1592, 2001.

[SSHD17] Josef Steinbaeck, Christian Steger, Gerald Holweg, and Norbert Druml. Next generation radar sensors in automotive sensor fusion systems. In *IEEE Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6. IEEE, 2017.

[SSSS97] Richard Szeliski, Heung-Yeung Shum, Heung-Yeung Shum, and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 251–258. ACM Press/Addison-Wesley Publishing Co., 1997.

[TMHF99] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment – a modern synthesis. In *International Workshop on Vision Algorithms*, pages 298–372. Springer, 1999.

[YGGS16] Okan Yurduseven, Vinay R. Gowda, Jonah N. Gollub, and David R. Smith. Printed aperiodic cavity for computational and microwave imaging. *IEEE Microwave and Wireless Components Letters*, 26(5):367–369, 2016.

[YGMS16] Okan Yurduseven, Jonah N. Gollub, Daniel L. Marks, and David R. Smith. Frequency-diverse microwave imaging using planar mills-cross cavity apertures. *Optics Express*, 24(8):8907–8925, 2016.

[YGT+16] Okan Yurduseven, Jonah N. Gollub, Kenneth P. Trofatter, Daniel L. Marks, Alec Rose, and David R. Smith. Software calibration of a frequency-diverse, multistatic, computational imaging system. *IEEE Access*, 4:2488–2497, 2016.

[Zha12] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.