Optimization-based Enhancement of THz Data and Image

Optimierungsbasierte Verbesserung von THz Daten und Bildern

DISSERTATION
zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften

von
Tak Ming Wong

eingereicht bei der
Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen

Siegen 2022

Betreuer und erster Gutachter
Prof. Dr.-Ing. Andreas Kolb
Universität Siegen

Zweiter Gutachter
Prof. Dr.-Ing. Peter Haring Bolívar
Universität Siegen

Tag der mündlichen Prüfung
05.05.2023

ABSTRACT

THz imaging is significantly limited in its spatial resolution due to the substantially longer wavelength of the associated frequencies, which has made the problem of imaging beyond diffraction limit to be an emerging challenge in the THz research community.

In this dissertation, an optimization-based THz data and imaging enhancement concept is introduced. In this context, inverse problems in THz data and image enhancement, such as parameter estimation, image reconstruction, denoising and deblurring, are expressed as mathematical optimization problems, in which the core components are a physical model and an optimizer. Instead of solely maximizing the subjective improvement in terms of the visual perception, the optimizer minimizes an objective measure between the THz physical model and the measured THz data. This concept enables various kinds of computational optimization methods, for example, classical gradient descent based optimizers and modern neural network based optimizers, to solve the non-convex optimization problems and to estimate the material-related THz parameters from the measured THz data.

Experiments show that this concept is beneficial for resolution enhancement, the ability to find energy minima, the requirements of measured data size, the robustness of parameter estimation, and computational resources. This study demonstrates the advantages brought by the cross-disciplinary collaboration between the fields of THz imaging and visual computing.

ZUSAMMENFASSUNG

Die räumliche Auflösung der THz-Bildgebung ist aufgrund der wesentlich längeren Wellenlänge der zugehörigen Frequenzen erheblich eingeschränkt, was das Problem der Bildgebung jenseits der Beugungsgrenze zu einer neuen Herausforderung für die THz-Forschung gemacht hat.

In dieser Dissertation wird ein optimierungsbasiertes Konzept zur Verbesserung von THz-Daten und Bildgebung vorgestellt. In diesem Zusammenhang werden inverse Probleme der THz-Daten- und Bildverbesserung, wie z. B. Parameterschätzung, Bildrekonstruktion, Rauschunterdrückung und Entschärfung, als mathematische Optimierungsprobleme ausgedrückt, wobei die Kernkomponenten ein physikalisches Modell und ein Optimierer sind. Anstatt nur die subjektive Verbesserung in Bezug auf die visuelle Wahrnehmung zu maximieren, minimiert der Optimierer ein objektives Maß zwischen dem physikalischen THz-Modell und den gemessenen THz-Daten. Dieses Konzept ermöglicht es verschiedenen Arten von rechnergestützten Optimierungsmethoden, z. B. klassischen gradientenabstiegsbasierten Optimierern und modernen neuronalen Netzwerk-basierten Optimierern, die nicht-konvexen Optimierungsprobleme zu lösen und die materialbezogenen THz-Parameter aus den gemessenen THz-Daten zu schätzen.

Experimente zeigen, dass dieses Konzept im Hinblick auf viele Teilbereiche von Vorteil ist: für die Verbesserung der Auflösung, die Fähigkeit, Energieminima zu finden, die Anforderungen an die Größe der gemessenen Daten, die Robustheit der Parameterschätzung und die Rechenressourcen. Diese Studie zeigt die Vorteile, die sich aus der disziplinübergreifenden Zusammenarbeit zwischen den Bereichen THz-Bildgebung und Visual Computing ergeben.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to those who have given me constant support throughout this amazing journey, without whom I would not have been able to complete this dissertation.

First of all, I am extremely grateful to my Ph.D. supervisor Prof. Dr. Andreas Kolb, who gave me such a great opportunity to work with his research group and trusted me unconditionally all these years. I would like to thank him sincerely for his unwavering support and astounding guidance, both personally and professionally.

I am also greatly thankful to Prof. Dr. Peter Haring Bolívar for his continuous support and patience, especially during our long, yet remarkably valuable discussions about the THz systems and models. Thanks should also go to members of Institute of High Frequency and Quantum Electronics (HQE), Matthias Kahl, Aya Souliman and Daniel Stock, for their inspiring THz work.

This endeavor would not have been possible without the contribution of my co-authors: Hartmut Bauermeister, Matthias Kahl, Daniel Stock, Anna Katharina Wigger; and I would like to express my appreciation for the great advice I received from Prof. Dr. Michael Möller. I would like to extend my sincere thanks to members of the doctoral committee, Prof. Dr. Peter Haring Bolívar, Prof. Dr. Andreas Kolb, Prof. Dr. Michael Möller and Prof. Dr. Kristof Van Laerhoven, for their effort to review this dissertation.

Special thanks to Deutsche Forschungsgemeinschaft (DFG), House of Young Talents (HYT) and Zentrum für Sensorsysteme (ZESS) for the financial support.

It is my honour to work with members of the Computer Graphics and Multimedia Systems group: Rustam Akhunov, Shida Beigpour, David Bulczak, Paramanand Chandramouli, Gianni Franchi, Andreas Görlitz, Willi Gräfrath, Hendrik Hochstetter, Nadine Hoffmann, Markus Kluge, Christian Köhler, Martin Lambers, Dmitri Presnov, Hamed Sarbolandi, Christoph Schikora, Ulrich Schipper, Hendrik Sommerhoff, Sarah Wagener and Rene Winchenbach.

Thanks to members of DFG Research Training Group 1564 - Imaging New Modalities, for our great knowledge-exchange sessions: Prof. Dr. Volker Blanz, Jonas Geiping, Mai Lan Ha, Miguel Heredia Conde, Thomas Klinkert, Christoph Pomrehn and Davoud Shahlaei.

Thanks to members of Computer Vision group, for all of the great time we shared: Hartmut Bauermeister, Hannah Dröge, Kanchana Vaishnavi Gandikota and Jonas Geiping.

I would like to acknowledge the collaboration with all SHK students, for their hard work in this research project: Holger Baars, Hannah Dröge, Robin Häbel, Jürgen Perwuschin and Christina Runkel.

I would also like to thank my friends Alex Murphy and Chloe Wong for their help and support. Last but not least, from the very bottom of my heart, I would like to thank my parents, family, friends, and my wife Jennie for their endless encouragement.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Adam	Adaptive Moment Estimation
AE	Autoencoder
AWGN	Additive White Gaussian Noise
BN	Batch Normalization
CNN	Convolutional Neural Network
CPU	Central Processing Unit
dB	Decibel
DOP	Deep Optimization Prior
DTFT	Discrete Time Fourier Transform
DTIFT	Discret Time Inverse Fourier Transform
FC	Fully Connected
FCN	Fully Convolutional Networks
FFT	Fast Fourier Transform
FMCW	Frequency Modulated Continuous Wave
GB	Gigabytes
GPU	Graphics Processing Unit
GT	Ground truth
IFFT	Inverse Fast Fourier Transform
LBFGS	Limited-memory BFGS algorithm
LeReLU	Leaky Rectified Linear Units
LR	Lucy-Richardson
MTF	Modulation Transfer Function
PPAE	Per-pixel Model-based Auto-encoder
PSF	Point Spread Function
PSNR	Peak Signal to Noise Ratio
Rx	Receiver
RMSE	Root Mean Square Error
SD	Standard Deviation
SNR	Signal to Noise Ratio
TDS	Time Domain Spectroscopy
THZ	Terahertz
TRA	Trust-Region Algorithm
TV	Total Variation
Tx	Transmitter

1

INTRODUCTION

1.1 MOTIVATION

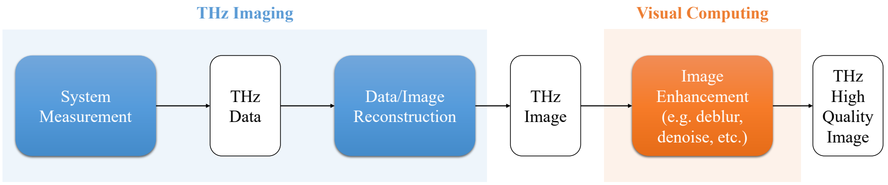
Since the beginning of THz imaging in the early 1990s [HN 95], it has shown great potential in contact-free analysis, non-destructive testing, and stand-off detection in various application fields, where a wide variety of designs were proposed for the individual application, such as semiconductor industry, biology, medicine, material analysis, quality control, and security [Sieg 02; CDM 07; Jans+ 10]. In many of these application fields, THz imaging is competing with established imaging methodologies, such as optical inspection and X-ray imaging. In comparison with imaging in the optical or X-ray parts of the electromagnetic spectrum, THz imaging is significantly limited in its spatial resolution due to the substantially longer wavelength of the associated frequencies. While systematic enhancement is limited by the dimension and precision of electronic and optical components, the computational enhancement approaches are beneficial due to the rapid development of modern computer science and applied mathematics fields, especially the state-of-arts computational optimization technologies.

1.2 CHALLENGES

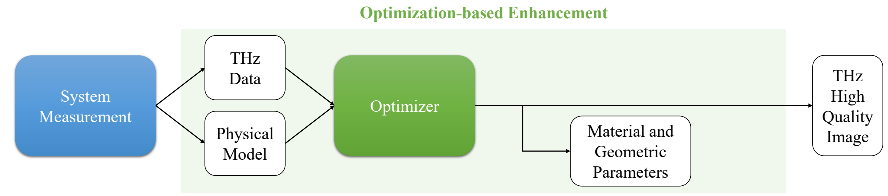
In principle, the systematic resolution of a THz imaging system is limited by its physical diffraction limit. However, as advanced computational imaging techniques have shown in recent years, it is indeed possible to resolve a target visually beyond the systematic resolution. This technology, which is commonly known as the super-resolution beyond the diffraction limit, raises a challenge in the THz community, whether a computational method can apply to THz imaging and can enhance the visual resolution.

One technical barrier of computational THz imaging comes from the very sophisticated THz physical model, which includes (but is not limited to) scattering effect, optical refraction, single-path, and multi-path reflection. This physical model is generally non-linear, e.g. multiplied with a periodic function and commonly raises highly ill-posed and non-convex problems from an image enhancement perspective.

Moreover, computational methods, such as machine learning methods, often require hundred-thousands of images and datasets for testing and verification. Although THz systems have shown signifi-



(a) Classical image-based enhancement method considers a THz image as a medium of information exchange. Visual computing expertise takes the THz image as input images for further enhancement (e.g. deblurring, denoising, super-resolution, etc.).



(b) Inverse problems in the optimization-based THz data and image enhancement method (e.g. parameter estimation, image reconstruction, etc.) are expressed as mathematical optimization problems, in which the core components are a THz physical model and an optimizer.

Figure 1.1: Overview of the optimization-based THz data and image enhancement concept.

cant breakthroughs, researchers have yet to resolve the difficulties to obtain large-scale THz image datasets. This limited size of THz data presents another technical barrier from a visual computing perspective, that it would restrain from using methods that rely on large datasets.

Besides the restricted data size, the robustness of computational enhancement methods is still in lack of study, whether the enhanced images are solely improved visually or are enhanced by their physical quantities. In addition, there is also a high practical interest to have methods that can be run and implemented in reasonably limited computational resources, such as time and memory.

These challenges can be difficult to overcome when existing THz computational enhancement methods are *image-based* methods, i.e. THz imaging systems reconstruct THz images from the measured THz data as the results before visual computing algorithms take these results (without further knowledge on the THz physical model) as input images for the visual enhancement, e.g. deblurring, denoising, super-resolution, etc. In this image-based mentality as depicted in Fig. 1.1a, the separation of either field would greatly limit the possibility of in-depth co-design work. Therefore, the study of a cross-disciplinary joint approach of *optimization-based* enhancement is necessary, as it would review the benefit of this collaborative philosophy.

1.3 CONTRIBUTION

In this dissertation, an optimization-based THz data and image enhancement conceptual model is presented. In this context, as illustrated in Fig. 1.1b, inverse problems in THz data enhancement (e.g. parameter estimation, image reconstruction, denoising, deblurring etc.) are expressed as mathematical optimization problems, in which the core components are a physical model and an optimizer. Instead of solely maximizing the subjective improvement in terms of visual perception, the optimizer minimizes an objective measure between the THz physical model (i.e. the forward model) and the realistic measured THz data. This optimization-based approach enables various kinds of optimizers, e.g. classical gradient-descent-based optimizers, to estimate the proper material-related THz parameters from the measured THz data by minimizing an objective discrepancy (like ℓ^2-squared loss, mean-square-error), i.e. *optimizing* in mathematics terminology. The measurement THz datasets are provided by colleagues from the Institute of High Frequency and Quantum Electronics (HQE), University of Siegen using their FMCW THz imaging systems, and the THz physical models are also mainly derived by HQE. The following contributions to the mentioned challenges are presented:

- To enhance the spatial resolution of the FMCW THz system, a novel method has been proposed by firstly reconstructing the THz signal in depth-direction which yields a significant improvement in depth estimation and signal parameter extraction. The resultant intensity THz image allows a 2D blind deconvolution process which enhances the lateral THz image resolution beyond the diffraction limit. The method has been published in [Wong+ 19a] and [Stoc+ 19].

- For the THz parameter estimation problem, a model-based autoencoder approach has been developed, in which the encoder neural network predicts suitable model parameters, and the decoder is fixed to the THz physical model. This allows the encoding network to be trained in an unsupervised machine learning approach, which makes the neural network feasible without ground-truth labeled THz datasets and improves the computation time much faster than classical optimizers. This autoencoder has been published in [Wong+ 19b].

- A deep optimization prior approach, i.e. the reparameterization of a pixel-wise non-convex THz model parameter estimation problem via a spatially coupled 3D neural network, has been proposed. Theoretically, the surjective reparameterization does not eliminate critical points of the non-convex cost function but yields a different gradient descent path, which allows this approach to find significantly better (local) minima compared to classical optimizers. The experiment demonstrates that this approach robustly reconstructs THz model parameters in low SNR and shot noise situations, and the network is highly memory efficient. Combining this approach with the modern blind deblurring method improves the lateral resolution enhancement robustly under low SNR noise conditions in comparison to the per-pixel curve fitting method. This approach has been published in [Wong+ 22a] and [Wong+ 22b].

- A first insight into the joint material and geometric parameter estimation for the THz multi-path reflection model is introduced. This approach derives the comprehensive formulas that transfer the measured data from the numerical aspect to the physical space, and interpret the THz multi-path reflection as a functional model of material and geometric parameters. The model allows expressing the parameter estimation as the direct optimal of the optimization problem, in which the theoretical analysis and experiments demonstrate that the optimizer based on the signal magnitude is generally better than the optimizer based on the complex parts of the signal. However, experiments show that so far this approach can not solve the optimization problem with sufficient robustness, and further investigation of the optimization method is needed.

1.4 OUTLINE

In chapter 2, the fundamental knowledge of this dissertation, which includes the THz system, numerical optimization, and machine learning, is provided. Chapter 3 introduces the computational THz image enhancement method to enhance the spatial resolution of the FMCW THz system. The model-based autoencoder approach for THz image reconstruction is described in chapter 4. In chapter 5, the deep optimization prior approach is explained in detail. Chapter 6 introduces the joint material and geometric parameter estimation for the THz multipath reflection model preliminarily. Finally, chapter 7 summarizes the dissertation and briefly discusses potential future works.

2

FUNDAMENTALS

This chapter provides some necessary fundamental knowledge for this dissertation. In Sec. 2.1, an overview of THz 3-D imaging is given, including the definitions of THz radiation, properties of THz imaging systems, methods of FMCW technology, and the principle of 3-D imaging. In Sec. 2.2, more details of the FMCW THz signals and the corresponding signal processing are expressed in the form of a mathematical model. Once the THz imaging is introduced, an introduction to convex and non-convex optimization theory is provided in Sec. 2.3, where different commonly used optimization methods are briefly described. Sec. 2.4 overviews the basic principle of machine learning and some terminology of modern deep neural network architectures. At the end of this chapter, Sec. 2.5 describes the details of the experimental system setup and the corresponding datasets used in this dissertation.

2.1 THZ 3-D IMAGING

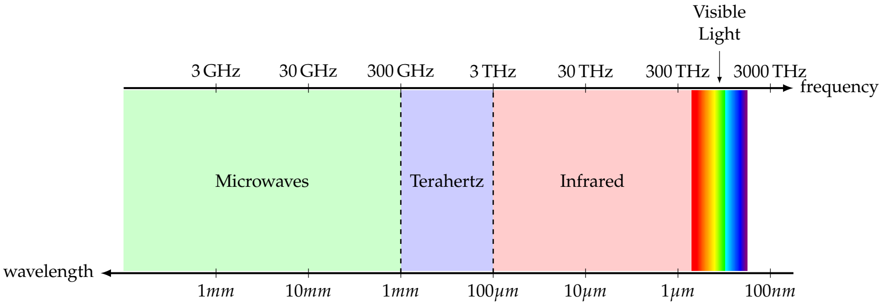


Figure 2.1: Terahertz radiation is commonly referred to the band between microwaves and infrared in the electromagnetic spectrum.

THZ RADIATION The term *terahertz radiation* (THz, or T-ray) is commonly referred to the electromagnetic spectrum around 10^{12} Hz. This term is analogous to microwaves, infrared and X-rays radiations, although it names a spectral band by its frequency unit [Lee 09]. In general, *Terahertz band* is referred to the radiation band applied to *sub-millimeter* waves (see Fig. 2.1) that fills the wavelength range between $1mm$ and $100\mu m$ (300 GHz $-$ 3 THz in frequency) [Sieg 02]. This THz band lies between microwaves and the infrared band, where the microwaves frequency band ends at 300 GHz. In contrast, when the frequency goes beyond 3 THz, the spectral band crosses into the far infrared radiation (around 10 THz) [Lee 09]. In the early era of THz research, the optical imaging system could only handle the higher frequency spectrum, while the electronic system could only work on the lower THz frequency. During the last one or two decades of development, both optical and electronic fields were trying to fill this technological vacancy (also called *terahertz gap*) of efficient optical and electronic devices for imaging in this THz spectrum. This is because suitable electronic circuits that handle high-frequency signals and appropriate optical systems for low frequency were difficult to build in a robust, inexpensive, and coherent manner [Cham 04]. To have a more comprehensive review of modern THz imaging technologies, the reader is referred to [MJN 96], [Sieg 02] and [Tono 07].

Within this sub-millimeter band, THz radiation is well known for its non-ionizing property while non-destructively penetrating many visually opaque materials [Cham 04]. This free-of-hazardous characteristic generates great interest in contact-free material analysis and non-destructive testing in application areas such as bio-chemical material analyses, civil security, package inspection, etc. in the semiconductor industry and the medical community [HN 95; MJN 96; Sieg 02; Cham 04].

2.1.1 *THz Imaging*

Most of the modern THz systems can be categorized by their radiation sources: active systems irradiate the object by the THz source integrated with the active system, while passive systems use the THz radiation from the natural environment only [Lee 09].

One important property of the THz imaging system is the coherence of the radiation source and the sensor-received signal. In a coherent radiation system, the phase differences and frequencies are all constantly detected and hence invariant during the acquisition time, while incoherent systems can only obtain signals with varying phase differences and frequencies and thus can only obtain the intensities of the signal. Hence, coherent systems can be used to reconstruct the phase and amplitude of the scattered electronic field, offering the possibility of obtaining additional information via more sophisticated reconstruction techniques such as depth information or material parameters of the objects. In general, atmospheric THz radiation in passive systems can only provide an incoherent radiation source; hence only active THz imaging systems with coherent THz radiation sources are considered in this dissertation.

In addition to radiation coherence, the radiation mode is another essential property, which mainly falls into two categories: pulsed and continuous wave (CW) technologies [Tono 07]. The THz pulsed method transmits femtosecond THz pulses to radiate the object, which is commonly used in THz Time-Domain Spectroscopy (THz-TDS)

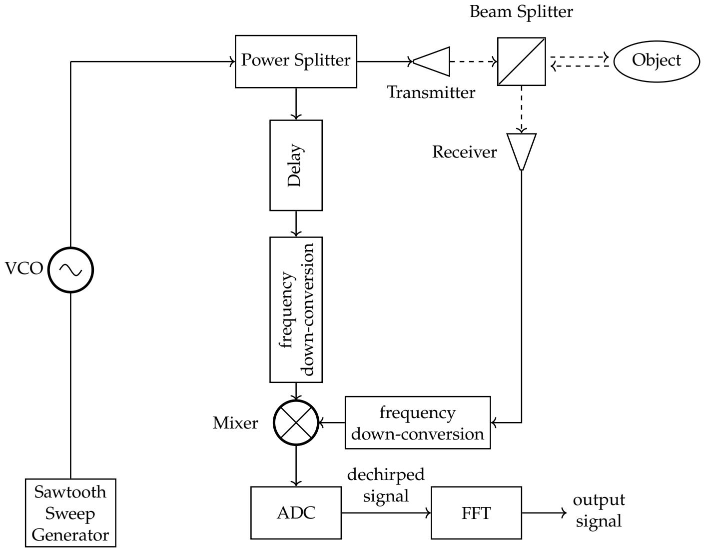


Figure 2.2: Overview of FMCW imaging system.

systems [HN 95; MJN 96]. In contrast, the CW method radiates the object continuously during the acquisition period.

Generally, the CW system is easier to integrate than the pulsed system, which makes the CW method a much more affordable system design, especially on applications operating with higher frequencies.

This dissertation builds on Frequency Modulated Continuous Wave (FMCW) data with the benefit of enabling the reconstruction of depth information and material properties, which are provided by the Institute of High Frequency and Quantum Electronics (HQE).

2.1.2 *Frequency Modulated Continuous Wave (FMCW)*

In the radar community, the major advantages of the FMCW radar system are commonly known as the wide dynamic range, low noise, and high average radiation power [Dani 96]. To capture a THz 3-D image, the remote sensing community combined linear FMCW technology and Synthetic Aperture Radar (SAR) signal processing technique [Ding+ 13; MHL 07], which can avoid building large lenses at the THz range. Instead of synthesizing a linear array with a broadband radar, one common alternative method is to mechanically move SAR transmitter-receiver (*transceiver*) sensors in a grid-like approach, which makes the imaging system economically efficient.

Fig. 2.2 illustrates an overview of FMCW radar imaging system and Fig. 2.3 shows the corresponding FMCW data processing [Stov 92; CGM 95; Dani 96; Ding+ 13].

LINEAR FM WAVEFORM A typical FMCW radar system transmits a linear frequency-modulated (FM) wave on a repetitive approach, which is due to the ease of signal generation and processing. A common name for this signal is *chirp* because a signal of rising frequency over time is similar to the chirping sound of a bird. One common transmitter (Tx) signal generation method is using a continuously varying carrier frequency voltage-controlled oscillator (VCO) controlled by a *sawtooth-like sweep* generator over a chosen frequency range $[f_{\min}, f_{\max}]$ and a sweep duration period T_p [Coop+ 17]. Let $S_\mathrm{T}(t)$ be the time-domain signal generated by the transmitter (Tx):

$$f_\mathrm{T}(t) = f_c + K_r t \tag{2.1}$$

$$S_\mathrm{T}(t) = a_\mathrm{T} \cdot \exp\left[i2\pi \int_0^t f_\mathrm{T}(t)dt\right] = a_\mathrm{T} \cdot \exp\left[i2\pi \left(f_c t + \frac{K_r t^2}{2}\right)\right] \tag{2.2}$$

where $f_\mathrm{T}(t)$ is the instantaneous frequency of the transmitter chirp signal $S_\mathrm{T}(t)$, a_T is the constant to determine the amplitude of the transmitter signal, f_c is the center frequency, and the starting frequency, i.e. $f_\mathrm{T}(0) = f_c$), K_r is a linear *chirp rate*. The bandwidth B of the pulse is $K_r T_p$.

Notice that instead of a real signal, the transmitter chirp signal $S_\mathrm{T}(t)$ is modeled as a complex signal with real and imaginary channels, where this analytical signal is generated by Hilbert Transform (or by Discrete Hilbert Transform [Oppe 99] for transformation in discrete time).

DECHIRPING After the signal is returned from the object (the scatterer), the received signal from the Rx receiver $S_\mathrm{R}(t)$ is delayed and attenuated:

$$S_\mathrm{R}(t) = a_\mathrm{R} \cdot \exp\left[i2\pi \left(f_c(t-\tau) + \frac{K_r(t-\tau)^2}{2}\right)\right] \tag{2.3}$$

where a_R is the amplitude of the receiver signal. Since the sweep frequency f_c is operating in THz and practically components are impossible to detect in this frequency range, the Tx and Rx signals are converted into the MHz frequency range by the *frequency down-conversion* (see Fig. 2.2), where the down-converted carrier frequency is denoted by f_{dc}. This down-converted receiver signal is mixed with a sample of down-converted transmitted waveform delayed by τ and results in a difference of frequency. This process is called *dechirping* or *deramping* of FM signal, while radar systems using this process is also known as *dechirp-on-receive* systems.

Then, by a proper analog-to-digital (ADC) conversion, the resultant dechirped signal has a frequency related to its time delay relative to the delayed chirp signal, where the frequency is named as *intermediate frequency (IF)* to distinguish from the wideband frequency of the

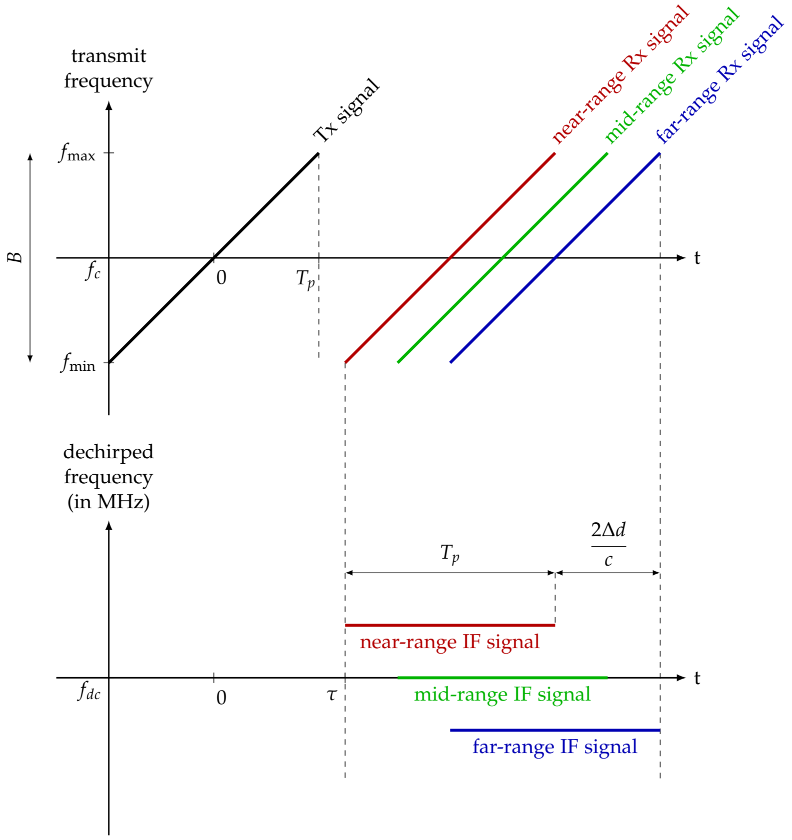


Figure 2.3: Dechirp processing of FMCW data from transmit time-frequency domain signal to dechirped intermediate frequency (IF) signal.

transceiver signal. The resultant IF signal $S_{\mathrm{IF}}(t)$ is the mixing product of the transmitted signal with the complex conjugate of the received signal:

$$S_{\mathrm{IF}}(t) = S_{\mathrm{T}}(t) \cdot \overline{S_{\mathrm{R}}(t)} = a_{\mathrm{IF}} \cdot \exp\left[i2\pi (K_r \tau t + f_{dc}\tau - \frac{K_r \tau^2}{2}) \right] \tag{2.4}$$

where $\overline{S_{\mathrm{R}}}$ denotes the complex conjugate of S_{R}.

2.1.3 *3-D Imaging*

By applying a Fourier Transform, the dechirped IF signal is converted to a range profile [MV 89], which represents the intensity of reflection at these frequencies respectively. Because of the linear FM waveform, these frequency profiles can also be interpreted as the profiles using different times of traveling, and hence are further considered as the reflective intensity at different depth positions. Therefore, these range profiles can also be expressed as *the spatial depth domain signal*, which is an essential component of FMCW THz imaging in the depth direction.

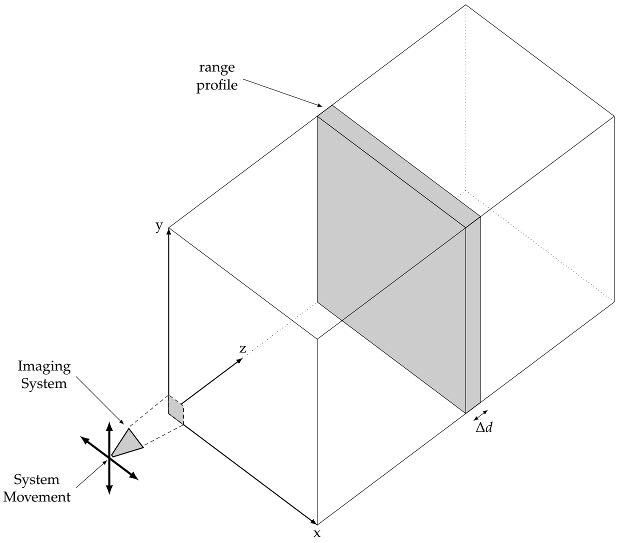


Figure 2.4: THz 3-D image is captured by grid based azimuth x, y movement of the FMCW imaging system. The data at each depth position forms 2-D images (*range profile*) with depth resolution Δd.

By repeating this imaging procedure for each azimuth x, y position as illustrated in Fig. 2.4, the entire THz 3-D image is captured.

By this FMCW 3-D imaging methodology, the system depth resolution Δd is:

$$\Delta d = \frac{c}{2B} \tag{2.5}$$

where c is the speed of light and $B = f_{\max} - f_{\min}$ is the system bandwidth. Notice that the system azimuth x, y resolution (also referred to as the *lateral* resolution) is independent of this FMCW scanning approach but depends on the physical movement and optical components.

2.2 FMCW THZ SINGLE-PATH REFLECTION MODEL

SAMPLING Consider the sampling rate is L samples per second, the discrete IF signal $S_{\mathrm{IF}}[n]$ is derived from (2.4):

$$S_{\mathrm{IF}}[n] = S_{\mathrm{IF}}\left(\frac{n}{L} \cdot T_s\right) = a_{\mathrm{IF}} \cdot \exp\left[i2\pi(K_r \tau T_s \cdot \frac{n}{L} + f_{dc}\tau - \frac{K_r \tau^2}{2})\right] \tag{2.6}$$

where $T_s = T_p / L$ is the sampling period. In sampling theory, a proper choice of sampling rate is critical to avoid aliasing.

Since the term $K_r \tau^2 / 2$ has a practically neglectable value compared to the term $f_{dc}\tau$ (because the time delay τ is commonly measured in the nano-second scale), by assuming $\mu = K_r T_s \tau$ and $\phi_\tau \approx 2\pi\,(f_{dc}\tau)$, the discrete IF signal a before deramp Fourier Transform is expressed by:

$$a[n] = a_{\mathrm{IF}} \cdot \exp\left(i2\pi\mu\frac{n}{L}\right) \cdot \exp\left(i\phi_\tau\right) \tag{2.7}$$

Note that the discrete signal n in (2.7) represents the received IF signal in the frequency domain (i.e. its frequency profile), which is no longer denoting the continuous time domain signal in (2.2) nor the discrete time domain signal n in (2.6).

WINDOWING FUNCTION The THz signal in the depth direction is expressed as a modulated signal with a *sinc*-envelope in the spatial model of the THz image in (2.7). To effectively suppress the sidelobe of the sinc-envelope, one of the common signal processing approaches is to introduce a windowing function, which is to multiply the Hamming window $H[n]$ to the discrete IF signal in (2.6) before the deramp Fast Fourier Transform (FFT):

$$a_{\mathrm{Hamming}}[z] = \mathcal{F}\left\{S_{\mathrm{IF}}[n] \cdot H[z]\right\} \tag{2.8}$$

$$\text{where} \quad H[n] = 0.54 - 0.46\cos\left(2\pi \cdot \frac{n}{L}\right) \tag{2.9}$$

However, this common signal processing practice introduces an extra term to the spatial model, and it appears that the envelope of the spatial domain signal is no longer a straightforward *sinc* function after the deramping Fourier Transform. Therefore, to keep the simplicity of the THz physical model, neither this Hamming nor any other windowing function is included in the time domain signal for the single-path reflection model in Chapters 3, 4 and 5.

ZERO-PADDING The signal processing community often includes the zero-padding technique, which appends (*pads*) extra zeros to the end of the frequency domain signal in (2.7) before the deramp Fourier Transform. To express this zero-padding technique on the discrete signal (2.7), let $E_N[z]$ be the zero-padded discrete IF signal by the zero-padding factor N_p:

$$a_N[n] = \begin{cases} S_{\mathrm{IF}}[n], & \text{if } 0 \leq n < L \\ 0, & \text{otherwise for } n < L \cdot N_p \end{cases} \tag{2.10}$$

Hence, after the Fourier Transform, the signal length is increased from L to $L \cdot N_p$. This zero-padding technique introduces more interpolation points between sampling points while retaining the same signal power and structure at these sampling positions. These extra interpolation points are very useful, especially in curve fitting and optimization approaches.

DISCRETE TIME FOURIER TRANSFORM (DTFT) For the deramping transform, the Discrete Time Fourier Transform (DTFT) should be used. Given an ideal square discrete time domain signal $x[n]$:

$$x[n] = \begin{cases} 1, & 0 \leq n \leq M-1 \\ 0, & \text{otherwise} \end{cases} \tag{2.11}$$

$$\text{for } 0 \leq n \leq N-1 \tag{2.12}$$

After DTFT, the corresponding discrete frequency signal is:

$$X[k] = \frac{\sin[2\pi kM/(2N)]}{\sin[2\pi k/(2N)]} \cdot e^{-i\frac{2\pi k}{N}\left(\frac{M-1}{2}\right)} \tag{2.13}$$

Moreover, by considering the time-shift property and the linearity property,

$$\begin{aligned} x[n] \cdot e^{j2\pi k_0 n/N} &\xrightarrow{\mathcal{F}} X[k-k_0] \\ ax[n-n_0] &\xrightarrow{\mathcal{F}} aX[k] \end{aligned} \tag{2.14}$$

the discrete IF signal (2.7) is transformed to the discrete frequency signal by DTFT.

$$\mathcal{F}\left\{a[n]\right\} = a_{\text{IF}} \cdot \frac{\sin\frac{2\pi(k-\mu)M}{2N}}{\sin\frac{2\pi(k-\mu)}{2N}} \cdot \exp\left[-i\frac{2\pi(k-\mu)}{N}\left(\frac{M-1}{2}\right)\right] \cdot \exp\left(i\phi_\tau\right) \tag{2.15}$$

Because of the linear FM waveform, this discrete frequency domain signal is linearly related to the time-of-travel of the THz radiation in the air. As the speed-of-light in the air is known, this time-of-travel can be further interpreted as the spatial displacement from the transmitter to the object. Hence, these frequency profiles can be expressed as the *range profiles* in terms of the depth positions of the object.

Therefore, these discrete frequencies k are interpreted as depth positions z, and the spatial domain signal of the FMCW THz signal is expressed as $A[z] = \mathcal{F}\left\{a[n]\right\}$. By assuming $M = L$ and $N = L \cdot N_p$, the combination of (2.10), (2.13) and (2.15) yields

$$\begin{aligned} A[z] &= a_{\mathrm{IF}} \cdot \frac{\sin \frac{2\pi(z-\mu)L}{2LN_p}}{\sin \frac{2\pi(z-\mu)}{2LN_p}} \cdot \exp\left[-i\frac{2\pi(z-\mu)}{LN_p}\left(\frac{L-1}{2}\right)\right] \cdot \exp\left(i\phi_\tau\right) \\ &= a_{\mathrm{IF}} \cdot \frac{\sin \frac{\pi(z-\mu)}{N_p}}{\sin \frac{\pi(z-\mu)}{LN_p}} \cdot \exp\left[-i\left(\frac{\pi(z-\mu)}{N_p} \cdot \frac{L-1}{L}\right)\right] \cdot \exp\left(i\phi_\tau\right) \end{aligned}$$

As $\mathrm{sinc}(t) = \dfrac{\sin(\pi t)}{\pi t} \approx \dfrac{\sin(\pi t)}{L \sin(\pi t/L)}$, and $(L-1)/L \approx 1$ when L is large, the discrete spatial domain signal can be simplified as:

$$A[z] \approx a_{\mathrm{IF}} \cdot L \cdot \mathrm{sinc}\,\frac{z-\mu}{N_p} \cdot \exp\left[-i\left(\frac{\pi}{N_p} \cdot (z-\mu)\right)\right] \cdot \exp\left(i\phi_\tau\right)$$

Therefore, by substituting $\hat{e} = a_{\mathrm{IF}} \cdot L$, $\sigma = 1/N_p$, $\omega = \pi/N_p$ and $\phi = \omega\mu + \phi_\tau$, the discrete spatial domain FMCW THz model is:

$$A[z] \approx \hat{e} \cdot \mathrm{sinc}\left(\sigma(z-\mu)\right) \cdot e^{-i(\omega z - \phi)} \tag{2.16}$$

where $\hat{e}, \mu, \phi$ represent the physical properties of the object in the z-direction. In this *single-path reflection* model, only one reflective impulse (e.g. the impulse reflected on metal) is received by the receiver. It is also possible to have a *multi-path reflection* model as the superposition of multiple received impulses (for example, the dielectric material in THz radiation such as silicon), which is discussed in Chapter 6.

2.3 OPTIMIZATION

OPTIMIZATION PROBLEM In mathematics, an *optimization* problem [BBV 04; NW 06] is formulated as

$$\begin{aligned} &\text{minimize} && f_0(u), \quad u \in \mathbb{R}^n \\ &\text{subject to} && f_i(u) \leq b_i, \quad i = 1, \ldots, m \end{aligned} \tag{2.17}$$

where the vector $u = (u_1, \ldots, u_n)$ is the *optimization variable*, the function $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*, functions $f_i : \mathbb{R}^n \to \mathbb{R}$ are the *constraint functions* based on *constraints* b_i.

The solution of this optimization problem u^* is called *optimal*, when it has the lowest objective value among all vectors that satisfy constraints such that

$$f_0(u) \geq f_0(u^*), \qquad \text{for any } u \text{ with } f_i(u) \leq b_i$$

From this general form of the optimization problem (2.17), in a particular case, a *convex* optimization problem is defined as the

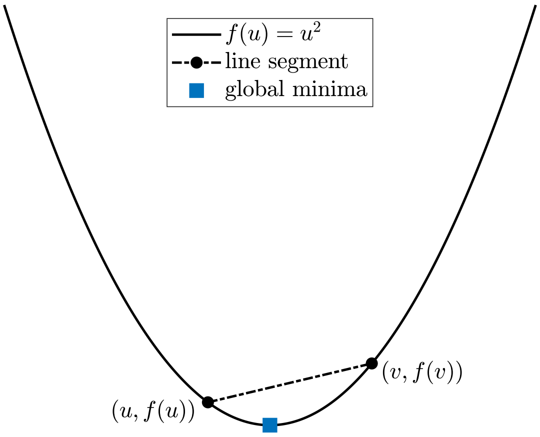


Figure 2.5: Example of a convex function. The line segment between any two points on the function lies above the function itself. The global minimizer (and also the solution) u^* when $\nabla f(u^*) = 0$ is the *global minima* of the convex function f.

optimization problem with convex set C and convex function f, where a set $C \subset \mathbb{R}^n$ is called convex if

$$\alpha u + (1 - \alpha)v \in C \tag{2.18}$$

for all $u, v \in \mathbb{R}^n$ and all $\alpha \in [0, 1]$. Given a convex set $C \in \mathbb{R}^n$, a convex function f satisfies the inequality

$$f(\alpha u + (1 - \alpha)v) \leq \alpha f(u) + (1 - \alpha)f(v) \tag{2.19}$$

for all $u, v \in C$ and all $0 \leq \alpha \leq 1$. As an example illustrated in Fig. 2.5, in geometry, this inequality implies that the line segment between $(u, f(u))$ and $(v, f(v))$ lies above the function f.

One important property of convex optimization is that given f is convex and differentiable at $u \in \mathbf{dom}\, f$, if $\nabla f(u) = 0$, then u is a *global minimizer* of the function f (*first-order condition*). This global minimizer serves as the desired solution u^*, while sometimes it is also called *global optima* or *global minima* (Fig. 2.5).

In the visual computing community, the convex function is usually referred to as a suitable discrepancy measure *energy function*, or also commonly known as *loss function*. One classical optimization example is the image denoising problem:

$$\begin{aligned} u^* &= \arg\min_u \sum f(u) = \arg\min_u \sum \mathcal{H}_g(u) + \alpha \mathcal{R}(u) \\ &= \arg\min_u \sum \|u - g\|_2^2 + \alpha \|Du\|_2^2 \end{aligned} \tag{2.20}$$

where $u \in \mathbb{R}^{n_x \times n_y}$ is the latent clear image, $g \in \mathbb{R}^{n_x \times n_y}$ is the measured noisy image. The energy function f usually consists of two components: the data term $\mathcal{H}_g$ measures the similarity of the desired solution

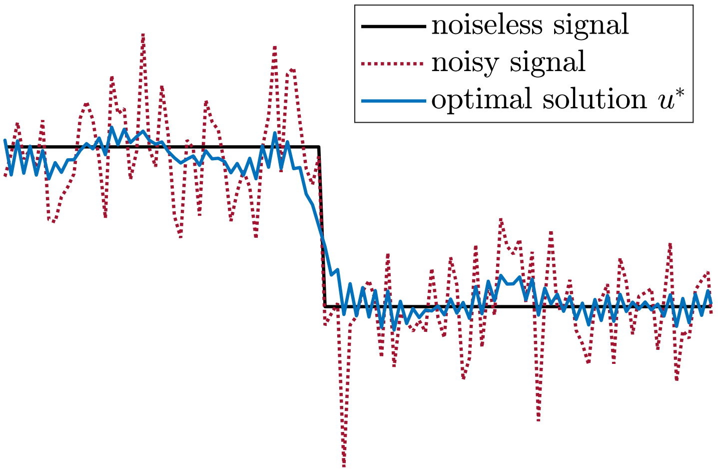


Figure 2.6: Example of a denoising problem by solving the optimization problem Eq. 2.20. L_2 regularizer is not good at preserving details with sharp edges.

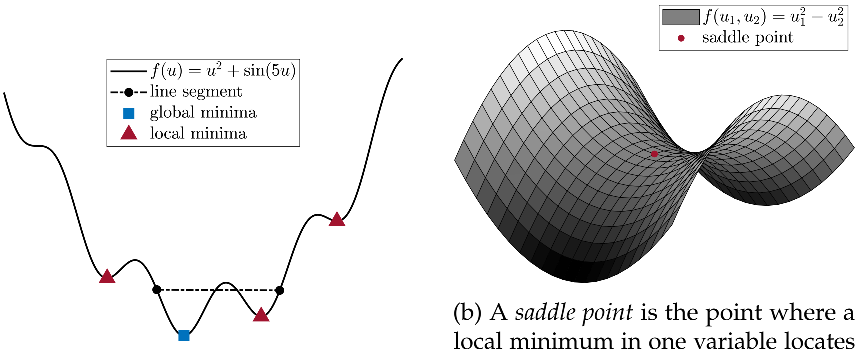
u to the data g; the regularization term $\mathcal{R}$ enforces the smoothness of the solution u, where the level of smoothness is controlled by the regularization coefficient α, and D denotes the finite difference matrix to compute the partial derivative ∇u.

Fig. 2.6 plots an example of a 1D signal denoising problem (i.e. a simplified case of image denoising problem), where the optimal solution u^* is calculated by the close form solution of L_2 regularization:

$$u^* = (I + \alpha D^T D)^{-1} g \tag{2.21}$$

where I denotes the identity matrix. This example demonstrates that the L_2 regularizer $\|Du\|_2^2$ is not optimal for denoising near the sharp edge as it introduces a blurry solution. In fact, in the visual computing area, a L_1 regularizer (i.e. $\|Du\|_1$) is more commonly adopted for the preservation of edges and also for reducing shot noise, and occasionally L_0 or $L_{p<1}$ regularizers are used. However, it appears that L_1 or L_0 regularization is a more sophisticated optimization problem, because the norms may not be differentiable all over the set, and it is a non-convex optimization problem.

In comparison with convex optimization, the non-convex optimization problem is indeed a more challenging problem, to which there is more than one local solution for the problem. In Fig. 2.7, examples of non-convex optimization problems are illustrated, while there are more than one *local minima* (Fig. 2.7a) or *saddle points* (also called *critical points*, see Fig. 2.7b) where $\nabla f(u) = 0$ without indicating a local minimum. The aim of solving the non-convex optimization problem is to find the global minima instead of local minima or critical points.



(a) The minimizers at local domain are called *local minima*.

(b) A *saddle point* is the point where a local minimum in one variable locates at a local maximum in another variable.

Figure 2.7: Examples of non-convex functions in single variable and multiple variables.

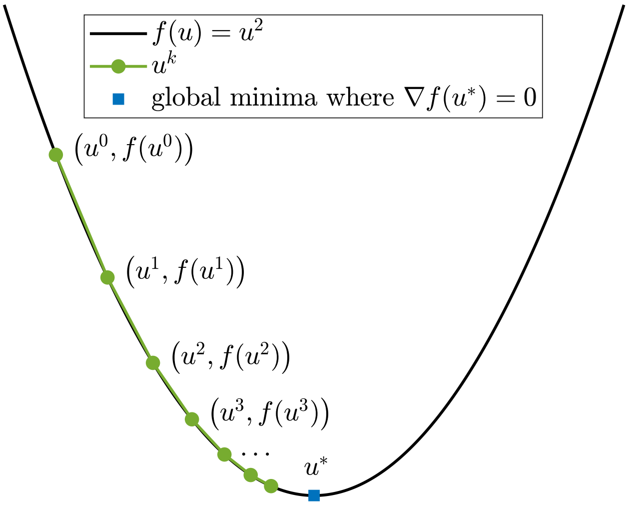
2.3.1 *Gradient Descent Method*

To solve an optimization problem, one of the most commonly used methods is the gradient descent method due to its simplicity. Gradient descent is an iterative procedure starting with an initial point $u^0 \in \mathbb{R}^n$,

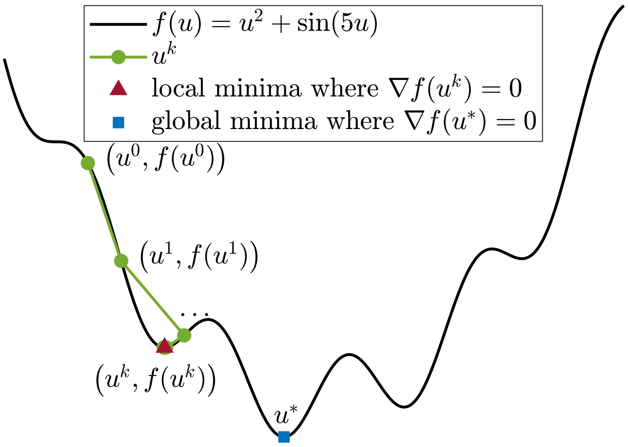
$$u^{k+1} = u^k - \lambda^k \cdot \nabla f(u^k), \quad \text{for } k = 0, 1, 2, \dots \tag{2.22}$$

where k denotes the k-th iterations, the gradient ∇f is the vector of partial derivatives of function f and $\lambda^k \in \mathbb{R}^+$ is the distance of travel at the k-th step (named as *step-size*). Since the gradient descent method (2.22) involves the first-order derivative ∇f only, it is often named the *first-order* gradient descent method. Fig. 2.8 shows examples of the gradient descent method for convex and non-convex optimization problems respectively. Conceptually, the gradient descent method starts with an initial estimation of variables, and then *descends* to the possible desired minima according to the direction (*gradient*) of the current position and the distance (*step size*) of one step. In the convex optimization problem (Fig. 2.8a), the gradient descent method converges to the global minima for an infinite number of iteration k. In practice, a *stopping criterion* is usually introduced to stop the iteration when the gradient is below a certain threshold (i.e. $\|\nabla f(u)\|_2 \leq \varepsilon$ where ε is small and positive).

However, as shown in Fig. 2.8a, the gradient descent method can converge to the local minima for non-convex optimization problems, which depends on the selection of the initial point (*initiliazation*) and the step size. In general, the traditional gradient descent method (2.22) is slow in convergence or unreliable for non-convex functions [GBC 16], even if there are some state-of-arts acceleration methods like momentum [Poly 63; Nest 03].



(a) The gradient descent method converges to the global minima for convex optimization problems.



(b) The gradient descent method can converge to the local minima for non-convex optimization problems.

Figure 2.8: Examples of the gradient descent method to solve optimization problems.

2.3.2 *Stochastic Gradient Descent (SGD) Method*

To solve non-convex optimization problems, the Stochastic Gradient Descent (SGD) method is an extension of the gradient descent method. As [SB 14] explained, SGD randomly selects a vector v^k for each iteration k:

$$u^{k+1} = u^k - \lambda^k \cdot v^k, \quad \text{s.t. } \mathbb{E}[v^k | u^k] \in \nabla f(u^k) \tag{2.23}$$

where $\mathbb{E}[v^k | u^k]$ denotes the expected value of the random vector v^k. As an illustration, SGD does not require the update direction for each iteration to exactly match the gradient itself, as shown in Fig. 2.9. Instead, the direction is allowed with a variation, and its *expected value* (i.e. $\mathbb{E}[v^k | u^k]$) at each iteration must match the gradient direction (i.e.

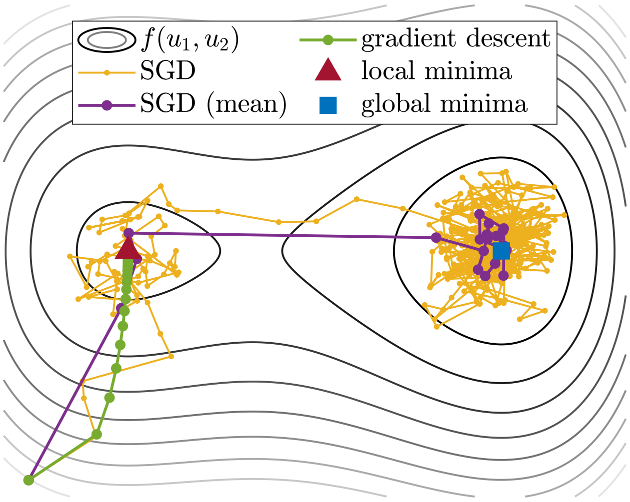


Figure 2.9: An illustration of optimization steps by the gradient descent method (green) and the Stochastic Gradient Descent method (SGD, yellow) on the non-convex function $f(u_1, u_2) = (u_1^4/4 - u_1^3/3 - u_1^2) + (u_2^2)$. The mean of 20 steps of SGD (purple) shows that SGD only requires the expected value of update steps to match the gradient direction.

$\nabla f(u^k)$). SGD is a crucial algorithm for solving non-convex optimization problems, especially in modern machine learning, because SGD allows the optimization on a subset of the variable (*batch*) which makes it computationally efficient in terms of memory and time, even with a large batch size and a large number of neurons.

In recent years, Adam [KB 14] and AdamW [LH 18] are two of the most popular methods among the variants of SGD algorithms, which are both first-order stochastic gradient descent methods with low order moment estimation and exponential decays of weights.

2.3.3 *Second-order Optimization Methods*

Besides the first-order condition $\nabla f(u^*) = 0$, the *second-order condition* $\nabla^2 f(u^*) > 0$ also determines the local minima u^*, if the Hessian $\nabla^2 f$ exists and is continuous in the local neighborhood [NW 06]. By considering the second-order Taylor series approximation of $f(u^k)$ near u^k,

$$u^{k+1} = u^k - \left(\nabla^2 f(u^k)\right)^{-1} \nabla f(u^k) \tag{2.24}$$

this iterative method finds the local minima of the function f by using the local approximation of the Taylor expansion. This method is commonly named Newton's Method, which is probably the most important *second-order* optimization method. Conceptually, in addition to the direction of step (gradient), the second-order method considers the *curvature* information of the function landscape. Besides the second-order Hessian matrix, an extension of Newton's Method called

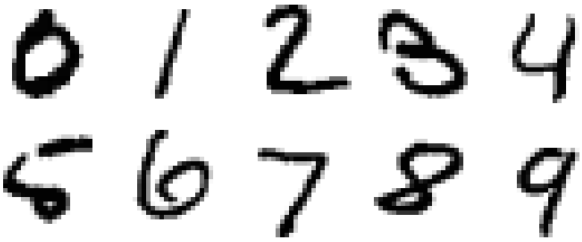


Figure 2.10: Examples of hand-written digits from the MNIST dataset. Courtesy from [LeCu 98].

the Trust Region approach [CL 96] introduces a boundary of steps (i.e. $\|u^{k+1} - u^k\|_2 \leq \Delta_k$) bounded by a trust-region radius Δ_k at each iteration.

Newton's method is well-known for its fast convergence rate, typically quadratic when compared to first-order methods. Also, due to the additional curvature information, the second-order method may avoid the local minima or saddle points for non-convex optimization problems, giving it a significant advantage over gradient descent methods. However, the explicit computation of the Hessian matrix $\nabla^2 f$ is a computationally expensive process ($\nabla^2 f \in \mathbb{R}^{n \times n}$ for $u \in \mathbb{R}^n$). To overcome this memory limitation, the Quasi-Newton method replaces the computation of the Hessian matrix with an approximation matrix, which is updated after each step to take account of the additional knowledge gained during the step. Until now, even the most commonly used quasi-Newton method, the Limited-memory BFGS algorithm (LBFGS [LN 89]), is still difficult to handle large-scale non-convex optimization problems such as the one arising in deep learning.

2.4 MACHINE LEARNING

SUPERVISED AND UNSUPERVISED LEARNING Since the first machine learning program was developed in the 1950s [Samu 59], machine learning has been a prominent area in the field of pattern recognition and artificial intelligence. Machine Learning (ML) is broadly defined in modern research [Mitc 97] as a computational algorithm that improves its performance at some tasks through some kind of *experience*. Consider the example of recognizing handwritten digits in Fig. 2.10 [Bish 06], the aim is to build a machine that takes images as input and that produces the identity of the digit $0, \ldots, 9$ as the output. Applications in which the *training data* comprises examples of the input vectors along with their corresponding target vectors (*labels*) are known as *supervised learning* problems. *Classification* problems are situations, such as the digital recognition example, in which the goal is to assign each input vector to one of a finite number of discrete categories. If the desired output consists of one or more continuous

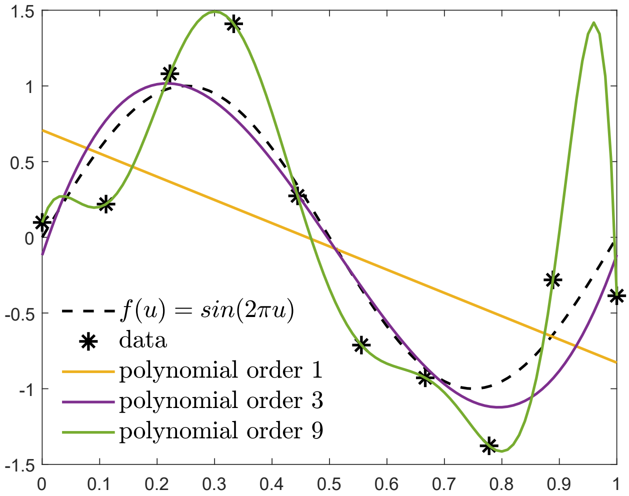


Figure 2.11: A plot of polynomial curve fitting with various orders to the data.

variables, then the task is called *regression*. An example of a regression problem would be the prediction of the yield in a chemical manufacturing process in which the inputs consist of the concentrations of reactants, the temperature, and the pressure.

Besides, when the training data consists of a set of input vectors without any corresponding target values, these problems are commonly referred to as *unsupervised learning* problems. Some examples of unsupervised learning are *clustering* which discovers groups of similar examples within the data, or *dimensional reduction* which projects the data from a high-dimensional space down to a lower dimensional space while retaining most of the relevant information.

Consider the regression problem illustrated in Fig. 2.11, which does a polynomial curve fitting to the measured data using a 1D function f with different orders. Notice that the low order (e.g. first order in yellow color) polynomials curve gives relatively poor fits to the data, and hence bad representations of the measurement. The appropriate order (e.g. third order in purple color) polynomials give the best overall fitting to the data. When the polynomial order is further increased (e.g. ninth order in green color), the polynomial curve has more freedom of movement, passes exactly through each data point, and minimizes the error. However, the fitted curve oscillates wildly and gives a very bad representation of the underlying function. This undesired behavior is known as *over-fitting* in machine learning. To solve this over-fitting problem, the machine learning theory introduces *validation datasets*, which select and exclude a subset of data from the training dataset. Instead of training this validation dataset, the machine evaluates the loss of this validation dataset during the training process. When there is an over-fitting like in Fig. 2.11, instead of a decreasing loss during training, the validation loss should be increasing and greater than the training loss, while this loss difference is commonly referred to

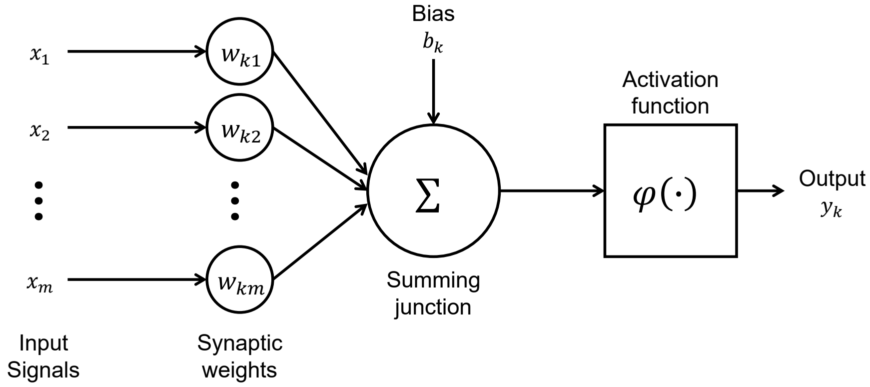


Figure 2.12: Nonlinear model of a neuron, labeled k.

as a *generalization gap*. Therefore, the samples of data are commonly separated into *training, validation, and testing* sets, where testing sets are used solely for evaluation.

2.4.1 *Neural Network*

Artificial neural networks, often known as *neural networks*, were one of the most prominent machine learning approaches. Historically, neural networks arose from a biological motivation of mimicking the behavior of neurons, such that the knowledge is stored by the inter-neuron connection, known as synaptic weights [Hayk 10]. Fig. 2.12 illustrates the mathematical model of neurons, which was first introduced in 1958 [Rose 58] and was later known as Rosenblatt's perception. In mathematical terms, the k-th neuron is expressed as:

$$y_k = \varphi\left(\sum_{j=1}^{m} w_{kj} x_j + b_k\right) \tag{2.25}$$

where $x_1, x_2, \ldots, x_m$ are the input signals; $w_{k1}, w_{k2}, \ldots, w_{km}$ are the respective synaptic *weights* of the neuron; φ is an activation function for limiting the amplitude of the neuron; b_k denotes the *bias* to control the net input of the activation function; and y_k is the output signal of the neuron. In machine learning terminology, generally the network learnable parameters θ mainly consist of the weights w and biases b of all learnable neurons.

Starting from this neuron model, a neuron network is defined as a direct graph consisting of nodes (neurons) with interconnecting synaptic and activation links [Hayk 10]. Fig. 2.13 illustrates an example of a fully connected neural network, where the neurons are organized in the form of layers. In this neural network, an input layer of source nodes connects to the hidden layers before projecting onto the output layers, where the term *hidden* refers to the fact that this part of the neural network is invisible from either the input or output of the

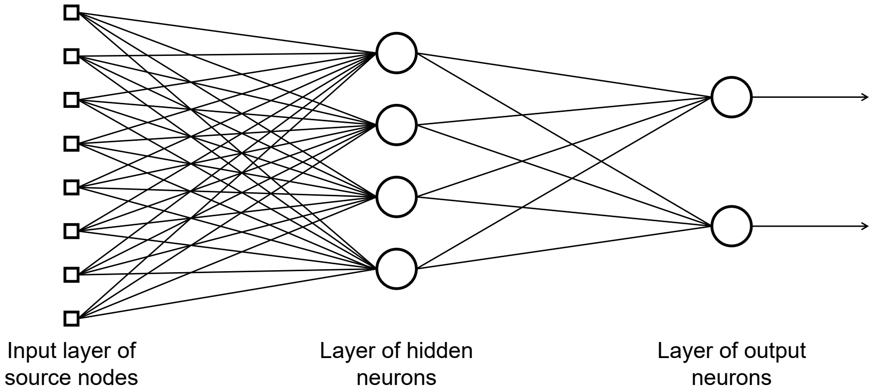


Figure 2.13: The network architecture of a fully connected neural network with one hidden layer and one output layer.

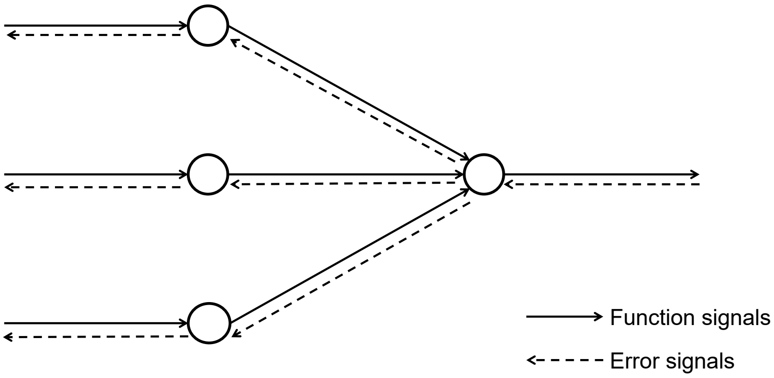


Figure 2.14: Illustration of forward propagation of function signals and backpropagation of error signals.

network. By adding one or more hidden layers, the network can extract higher-order statistics from its input. The network is described as *fully connected* only if every neuron in each layer of the network is connected to every other node in the adjacent forward layer.

To train a feed-forward neural network, the *back-propagation* is a widely used algorithm in machine learning. The basic idea is to compute partial derivatives of the loss function $\mathcal{L}$ realized by the network with respect to all adjustable weight vectors w, which implies that *training the neural network is to solve an optimization problem* by updating the network weights using the gradient descent method.

First, considering the single neuron model (2.25), the function signals y_j appearing at the output of neuron j, the error signal e_j and the loss function $\mathcal{L}$ (takes square loss as a simple example) are:

$$v_j = \sum_{i=0}^{m} w_{ji} \cdot y_i \tag{2.26}$$

$$y_j = \varphi\left(v_j\right) \tag{2.27}$$

$$e_j = d_j - y_j \tag{2.28}$$

$$\mathcal{L} = \frac{1}{2} \cdot e_j^2 \tag{2.29}$$

where y_i are the input signals of neuron j, d_j is the desired response vector (i.e. annotated label), m is the total number of inputs applied to neuron j, and the weight w_{j0} equals the bias b_j applied to neuron j. By applying the chain rule, the partial derivative of the loss function with respect to the network weights is:

$$\frac{\partial \mathcal{L}}{\partial w_{ji}} = \frac{\partial \mathcal{L}}{\partial e_j} \cdot \frac{\partial e_j}{\partial y_j} \cdot \frac{\partial y_j}{\partial v_j} \cdot \frac{\partial v_j}{\partial w_{ji}} \tag{2.30}$$

$$= -e_j \cdot \varphi'(v_j) \cdot y_i \tag{2.31}$$

Assuming all functions and weights are *differentiable* in the context, the gradient ∇w_{ji} applied to w_{ji} is:

$$\nabla w_{ji} = -\tau \cdot \frac{\partial \mathcal{L}}{\partial w_{ji}} = \tau \cdot \delta_j \cdot y_i \tag{2.32}$$

$$\text{where} \quad \delta_j = e_j \cdot \varphi'(v_j) \tag{2.33}$$

where τ is the *learning rate* parameter of the back-propagation algorithm and δ_j denotes the local gradient. The minus sign accounts for gradient descent in weight space (i.e. seeking the direction for weight change that reduces the value of $\mathcal{L}$).

From (2.33), a key factor involved in the calculation of the weight adjustment ∇w_{ji} is the error signal e_j. In this context, two distinct cases can be identified: neuron j is an output node and neuron j is a hidden neuron. In case one where neuron j is an output node, this is simple to handle because the error signal e_j can be computed directly by (2.28). In case two where neuron j is a hidden neuron, by considering the error signal flow to the output neuron k, the back-propagation formula of the local gradient δ_j yields:

$$\delta_j = \varphi'(v_j) \cdot \sum_k \delta_k w_{kj}, \qquad \text{where neuron } j \text{ is hidden} \tag{2.34}$$

More details of intermediate mathematical equations are referred to [Hayk 10, Section 4.4]. Since the term $\sum_k \delta_k w_{kj}$ only contains terms in later layers, the local gradient δ_j can be calculated from output to input on the network [Ripl 07], which is the weighted sum of the δ-s computed for the neurons in the next hidden or output layer that is connected to neuron j [Hayk 10].

Conceptually, as a feed-forward network depicted in Fig. 2.14, the forward function signal is an input stimulus signal that comes from the input layer of the network propagating forward (neuron-by-neuron) through the layers to the output end; the error signal, which is induced when the error function is computed by the difference between the output nodes of the network and the labels, originates at an output neuron of the network and *propagates backward* (layer-by-layer) until the input layer.

2.4.2 *Deep Learning*

In the early stages development of neural networks, networks were generally believed to be very difficult to train, which could be simply because these networks were computationally too costly to allow experiments with the hardware available at the time. However, with the rapid development of parallelable and distributable computational devices, neural networks were getting more layers (*deeper* and bigger) and more complicated structures, especially because of the breakthroughs made since 2006 [HOT 06]. Nowadays, the term *deep learning*, which commonly refers to machine learning using a deep neural network, extends beyond the neuroscience perspective. Instead, deep learning is mostly interpreted from the perspective of function approximation problems, which mainly focuses on applied mathematics fields such as linear algebra, probability, information theory, and numerical optimization. A more detailed history of deep learning development can be found in [GBC 16, Section 1.2].

Deep Learning has revolutionized the field of computer vision over the past decade, pushing previously infeasible tasks such as faithful image classification beyond a human accuracy level on challenging data sets like ImageNet [Deng+ 09]. In addition, the availability of a significant amount of training data has enabled deep learning techniques to dominate several areas of image reconstruction, such as denoising [Zhan+ 17], deblurring [Xu+ 14], super-resolution [Dong+ 15], inpainting [Path+ 16], or the reconstruction of images from different modalities such as medical [Litj+ 17] or time-of-flight data [Su+ 18].

While a lot of earlier research has focused on manually constructing parts of the mapping represented by the network, and merely learning simple parts, e.g. linear, of the overall function, the success of *deep learning* is based on two fundamental pillars: First, the realization that a large number of image processing related tasks can be represented extremely well by the network that is a *deeply nested* composition of convolutions (i.e. local linear filters) and simple component-wise non-linearities (e.g. setting negative entries to zero).

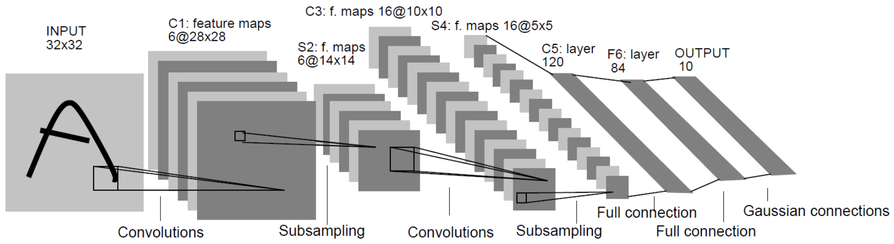


Figure 2.15: The architecture of LeNet-5, a renowned convolutional neural network for digits recognition. Courtesy of [LeCu+ 98]. The input image is sent through a sequence of nested, parameterized *convolutions* followed by a simple activation such as the *rectified linear unit* (ReLU) and sub-sampling layers (also called *pooling*). Modern deep networks have far more than 7 convolutional layers, often more than 100 channels per layer, and additionally use skip connections and batch normalization (not depicted here).

Second, the computational power to train such deep networks on the desired task in an end-to-end fashion.

Fig. 2.15 illustrates the architecture of LeNet-5 [LeCu+ 98], a renowned design of Convolutional Neural Networks (CNN) for classifying images: The input image is filtered with several convolutions with filter parameters named *learnable kernels* (i.e. synaptic weights and bias in classical neural networks terminology). These learnable convolutional filters are commonly named *convolutional layers*.

For the activation functions, modern deep neural networks utilize simple activation functions, especially the *rectified linear unit* [GBB 11] (ReLU, i.e. $\varphi(x) = \max(0, x)$) for its computational simplicity, or the Leaky ReLU [MHN+ 13] to ensure a non-zero gradient over the entire domain.

A technique called *stride* and sub-sampling operations called *pooling layers* compose many *layers* of such convolutions and simple activation functions, along with slowly decreasing the resolution of the structure by simply omitting every other computation. Finally, the likelihood of each class is calculated as weighted linear combinations of all outputs of the long chain of convolutions, i.e. the *fully connected layers* (structured like a classic feed-forward neural network in Fig. 2.13). Skip connections [He+ 16] and batch normalization techniques [IS 15] are two extremely significant extensions of the basic prototype demonstrated in Fig. 2.15 that enable training through ever-deeper networks.

For computational tasks where the output image needs to retain the spatial structure of the input image, it is common to first decrease the spatial resolution, similar to the classification network above, and then increase it again, possibly with the help of previously computed high resolution features, as is done, for example, in the celebrated U-net architecture [RFB 15] (shown in Fig. 2.16). In such architectures, the resolution-decreasing part is frequently referred to as an *encoder*,

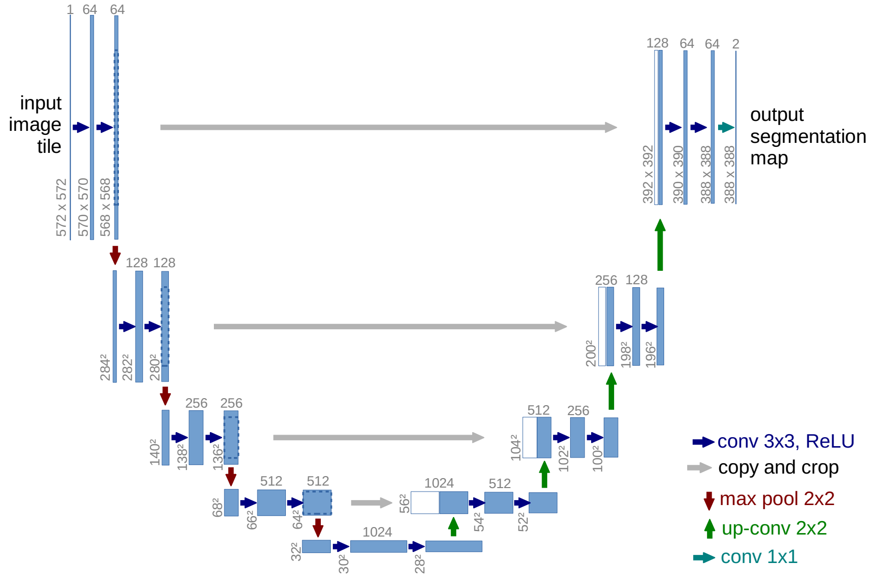


Figure 2.16: An architecture of classic U-Net. Courtesy of [RFB 15].

compressing the representation of an image to a few key parameters, and the resolution-increasing part is referred to as the *decoder*.

Recently, the seminal *deep image prior* work [UVL 18b] proposed to use an implicit regularization of a reconstruction process through a suitable parameterization via a neural network, which is now commonly known as *network reparameterization*. This network reparameterization approach tries to benefit from the beneficial structure of deep neural networks with less or no information about the desired predictions, which is also known under the names of *weakly-supervised* or *unsupervised* learning. This technique is gaining attention in convex optimization applications like image denoising, reconstruction, super-resolution, and inpainting.

2.5 EXPERIMENTAL SETUP

In this section, details of the all-electronic THz imaging system are given in Sec. 2.5.1. The details of measurement and synthetic datasets used in this chapter are provided in Sec. 2.5.2.

2.5.1 *All-electronic FMCW THz Imaging System*

The all-electronic FMCW THz imaging system is based on hollow-waveguide multipliers and mixers, operating in a frequency modulated continuous wave (FMCW) mode for measuring depth information [Ding+ 13]. The components operate at a center frequency of 577GHz with a bandwidth of 126GHz.

Fig. 2.17 illustrates the working mechanism of the imaging system, and Fig. 2.18 shows the imaging system unit. Regarding the structural

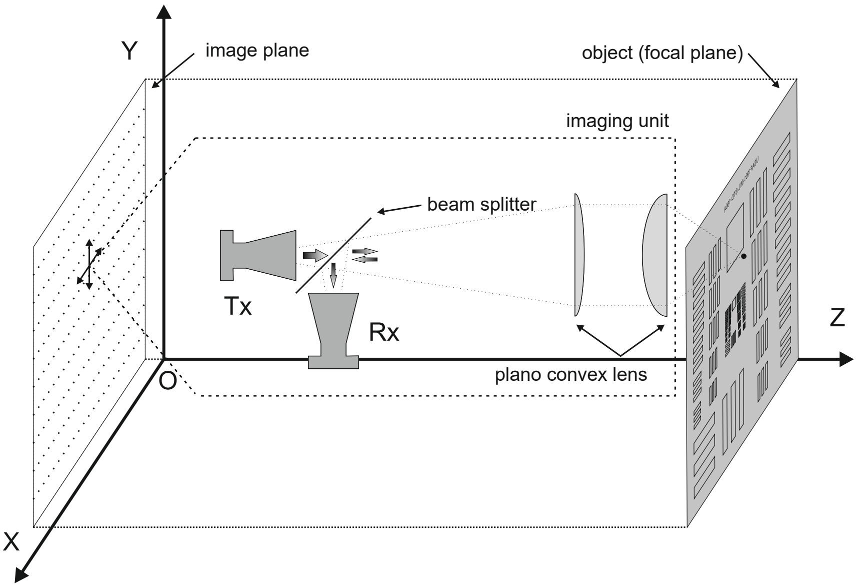


Figure 2.17: An illustration of the working mechanism of the THz 3D imaging system, where the transmitter (Tx) and the receiver (Rx) are mounted on the same platform.

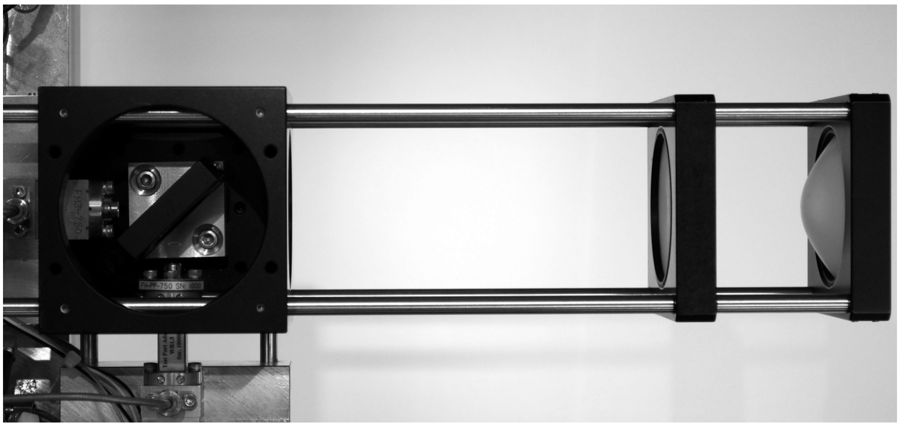


Figure 2.18: Photo of the THz 3D imaging system.

setup, both the transmitter (Tx) and the receiver (Rx) are mounted in a monostatic geometry, i.e. share the same observation direction. The imaging unit, consisting of Tx, Rx, and optical components, is moved along the x and y direction using stepper motors and linear stages. The beam splitter and two hyperbolic lenses focus the beam radiated from the transmitter to the target and then focus the beam reflected from the target back to the receiver.

For the FMCW operation, a voltage controlled oscillator (VCO) is tuned from 14.28 – 17.78GHz (see Fig. 2.19). The signal at the output of the VCO is distributed to the Tx and the Rx using a power splitter. For transmission, the signal is then upconverted to 514 – 640GHz

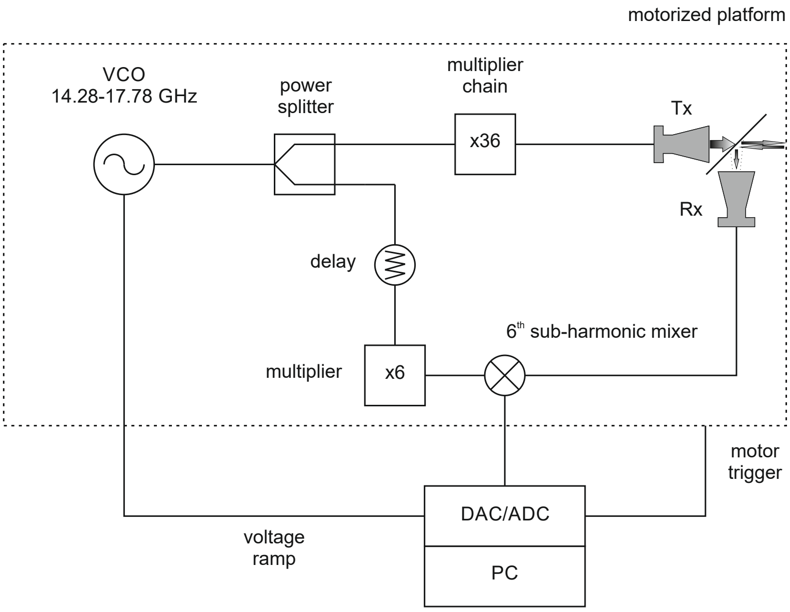


Figure 2.19: Schematic of the THz 3D imaging system setup and the host PC.

with a chain of multipliers. The signal is digitized with 10 MS (megasamples) per second sampling rate and transferred to the host PC after being down-converted to the intermediate frequency (IF) range using a sub-harmonic mixer, which is fed with the 6th harmonic of the VCO signal An additional delay between the Tx and Rx path cause a frequency offset in the intermediate frequency signal for proper data acquisition.

This imaging unit takes a depth profile of the object at each lateral position to acquire a full 3D image. The data is acquired with a lateral step size of $262.5\mu m$ in the xy-direction. During measurements, the motor controller and the data acquisition are synchronized to enable on-the-fly measurements. An adequate integration time and velocity are chosen to provide enough time for the acquisition of 1400 samples per depth profile and 36 averages per sample. The total per-pixel acquisition time for such an averaged depth profile is $5ms$.

The resolution of the setup is measured as $793.7\mu m$ using a metallic USAF 1951 Resolving Power Test Target scaled to the THz frequency range, which is close to the theoretical ideal expectation of $622\mu m$ (numerical aperture NA$= 0.508$ @ 578GHz). This is achieved by measuring minimum dimension which obtains more than 3dB intensity difference. Due to the monostatic scanning approach, no optical magnification or vignetting occurs. The system's depth resolution Δd is defined by

$$\Delta d = \frac{c_0}{2 \times B} \tag{2.35}$$

where c_0 is the speed of light in air, and B is the system bandwidth. Therefore, by calculation, this imaging system has $\Delta d = 1210\mu m$ depth resolution.

SYSTEM CALIBRATION Before each experimental measurement, system calibration is performed to ensure that the measured signals are correlated to the reference target [Ding+ 13]. In all experiments of this thesis, an isotropic metallic target is chosen as a reference target, which has a strong and stable reflection upon the THz radiation. Hence, a single measurement of the metallic target is recorded as the S_{max} signal, while a free space measurement is recorded as the S_{min} signal. The calibration procedure is formulated as:

$$S_{norm} = \frac{S_{measure} - S_{min}}{S_{max} - S_{min}} \tag{2.36}$$

where S_{norm} is the normalized stored signal and $S_{measure}$ is the measurement signal before calibration.

SYSTEM SETUP FOR MULTI-LAYER REFLECTION DATASETS For multi-layer reflection datasets (see Sec. 2.5.2.2), some of the settings are adjusted: the system is operating from 499.96 GHz to 733.38 GHz. The data is acquired by 1400 samples per depth profile and 106 averages per sample, while the lateral step size remains at $262.5\mu m$ in the xy-direction.

2.5.2 *Datasets*

In this dissertation, evaluations are based on measurement datasets and synthetic datasets based on single-path reflection (Sec. 2.5.2.1) as well as multi-path reflection (Sec. 2.5.2.2), where the number of samples is denoted by N_x for horizontal, N_y for vertical, and N_z for depth direction respectively.

2.5.2.1 *Single-path Reflection Datasets*

MEASUREMENT DATASETS Evaluations in Chapters 3, 4 and 5 are based on the following measurement datasets:

- **MetalPCB**: A nearly planar "USAF" target is etched on a circuit board (Fig. 2.20). The dataset has been acquired using the setup described in Sec. 2.5.1 and has the resolutions $N_x = 446$, $N_y = 446$, $N_z = 1400$. The lateral per-pixel distance (i.e. the sensor mechanical movement distance) is $262.5\mu m$. The target is etched using the standard size scale of USAF target *MIL-STD-150A* [Stan 59].

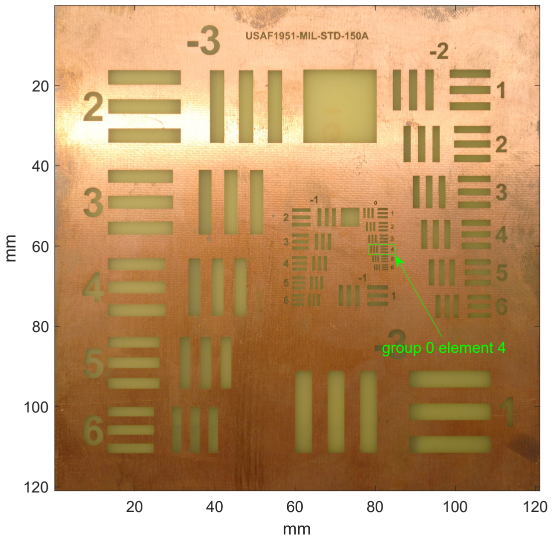


Figure 2.20: **MetalPCB** datasets are measured by metal on a PCB board test target fabricated according to USAF 1951 *MIL-STD-150A* standard, where the group and row numbers indicate the lines per millimeter according to [Stan 59]. Group 0 Element 4 is indicated.

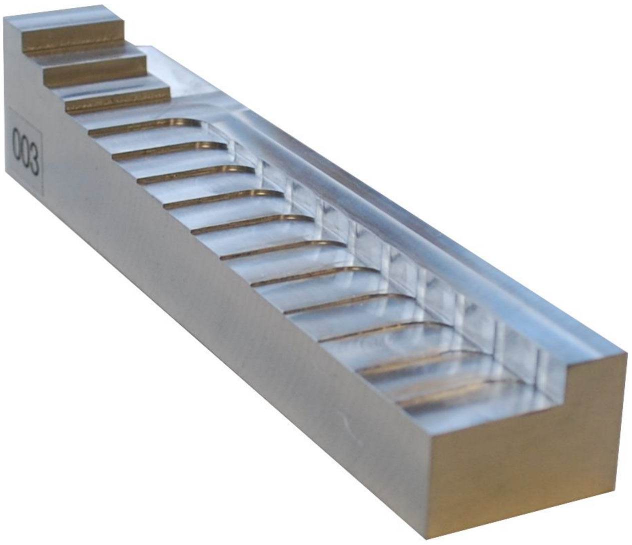
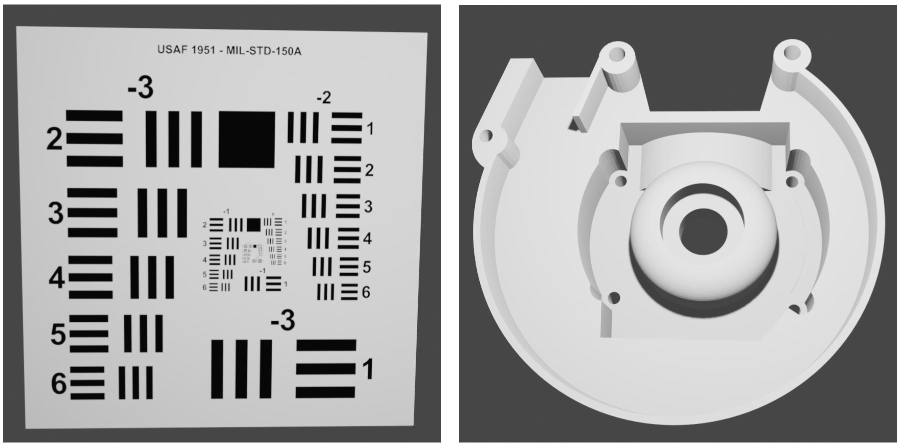


Figure 2.21: **StepChart** datasets are measured by a metallic step object with a reference zero.

- **StepChart**: A metallic step chart with steps varying from $4000\mu m$ to $50\mu m$, and a reference plane to locate the reference zero position (Fig. 2.21). The dataset has also been acquired using the setup in Sec. 2.5.1 with the resolutions $N_x = 575$, $N_y = 113$, $N_z = 1400$. The lateral per-pixel distance is $262.5\mu m$.

It should be noted that the THz system has a lateral resolution of $622\mu m$ as its ideal diffraction limit, an experimentally measured lateral



(a) **SynthUSAF** (b) **SynthObj**

Figure 2.22: Synthetic datasets are generated by 3D objects from a USAF resolution target and an engine bearing part from MVTec ITODD datasets [Dros+ 17]

resolution of $793.7\mu m$ and a depth accuracy of $\Delta d = 1210\mu m$ as its diffraction limits (see Sec. 2.5.1).

Due to the memory limitation of machine learning approaches (see Chapters 4 and 5), **MetalPCB** datasets are cropped out by 91 measurements in the spatial depth direction centered around the main lobe for experiments in Chapters 4 and 5, i.e. $N_z = 91$. Details of the cropping window are described in Chapter 3 Sec. 3.3.1.

SYNTHETIC DATASETS In addition to measurement datasets, the following synthetic datasets are simulated for experiments in Chapter 5:

- **SynthUSAF**: The ground truth THz model parameters are synthetically generated using the THz model equation (2.16) based on a tiled planar USAF object (Fig. 2.22a).
- **SynthObj**: The ground truth THz model parameters are synthetically generated from a 3D object (Fig. 2.22b) from MVTec ITODD datasets [Dros+ 17].

To simulate different noise levels, two synthetic noise models are used:

- **+AWGN**: adding Additive White Gaussian Noise (AWGN) by -20 to $10dB$ peak-SNR (PSNR) to the frequency domain signal, with a static background noise level and a varying signal power.
- **+ShotNoise**: On top of the AWGN, a random salt-and-pepper noise signal with a power of $60dB$ higher is added for 10% of the pixels.

All synthetic datasets are generated using the size of cropped **MetalPCB**, i.e. $N_x = 446$, $N_y = 446$, $N_z = 91$.

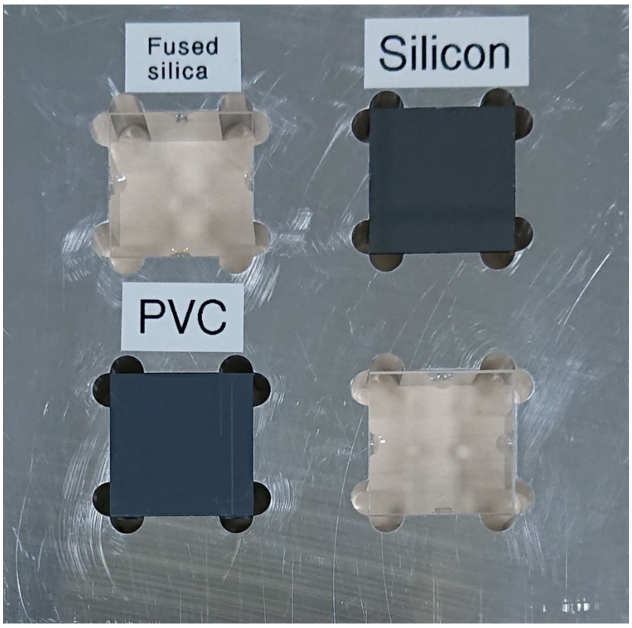


Figure 2.23: **MPTarget** datasets are measured by a metallic frame containing four different dielectric materials: Fused Silica, Silicon, PVC, and COC (bottom right). *No supporting structure* is under any of these four materials (i.e. only *air* behind materials).

2.5.2.2 *Multi-path Reflection Datasets*

For multi-path reflection datasets, the number of samples is denoted by N_x for the horizontal direction, N_y for the vertical direction, L for the frequency domain respectively.

MEASUREMENT DATASETS Evaluations in Chapter 6 are based on the following measurement dataset:

- **MPTarget**: A metallic frame containing four different dielectric materials: Fused Silica, Silicon, PVC, and COC (Fig. 2.23). Note that there is *no* supporting structure beneath these four materials, i.e. there is *only air* behind the materials. The datasets have been acquired by the setup described in Sec. 2.5.1 and has the resolutions $N_x = 413$, $N_y = 413$, and the number of samples in the frequency domain $L = 1400$.

Fig. 2.24 shows the sum of the measured signal magnitudes in the **MPTarget** datasets, where signal magnitudes are calibrated to the metallic region. The reduced signal magnitudes by these four materials indicate that the four dielectric materials are partially transmitting and partially absorbing the THz radiation energy in the operating THz frequency range.

To have a reference value of the material parameters, the complex refractive index of these four materials are measured by the Institute of High Frequency and Quantum Electronics (HQE) from 200 GHz

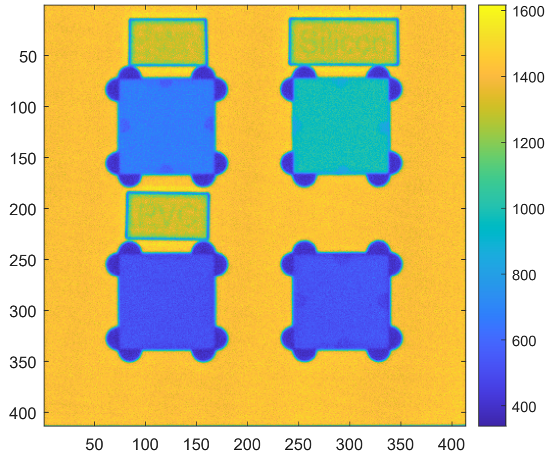


Figure 2.24: The sum of the measured signal magnitudes in **MPTarget** datasets.

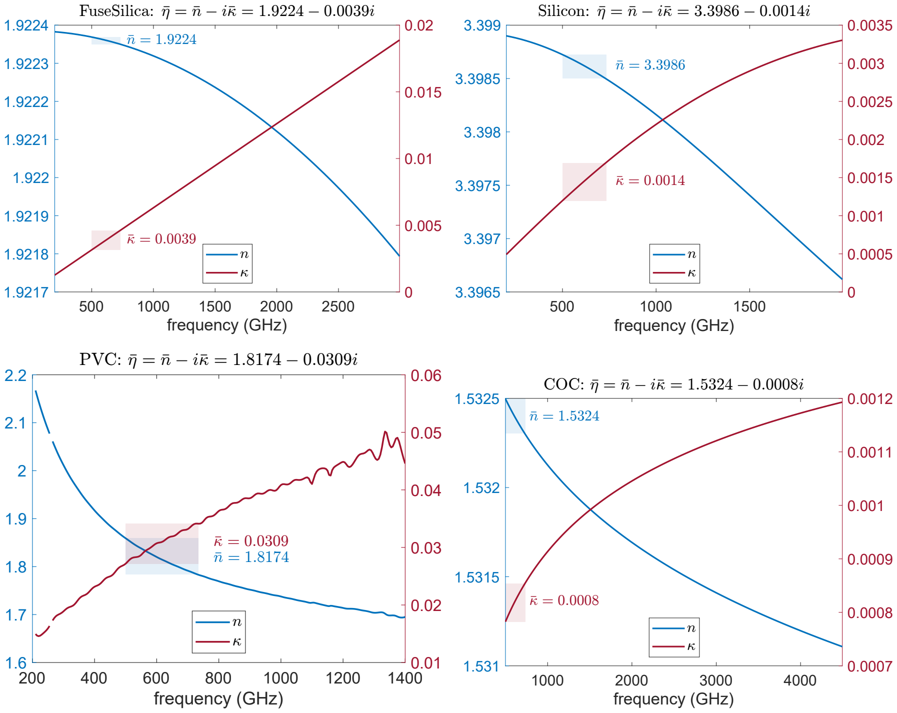


Figure 2.25: Measured complex refractive index of all four dielectric materials in the **MPTarget** datasets from 200 GHz to 4.495 THz. The measured refractive index within the operating frequency (from 499.96 GHz to 733.38 GHz) are averaged on real and imaginary parts individually and are taken as the reference complex refractive index $\overline{\eta}$ for comparison in Chapter 6.

to 4.495 THz respectively and are plotted in Fig. 2.25. The measured refractive index within the operating frequency, which is 499.96 GHz – 733.38 GHz, is averaged on real and imaginary parts individually and is taken as the reference complex refractive index $\overline{\eta}$ for comparison in Chapter 6. For the material thickness, around 10 measurements (mechanically by the HQE) are taken across each material respectively,

Material	refractive index $\overline{\eta}$	thickness
FuseSilica	$1.9224 - 0.0039i$	$1016\mu m$
Silicon	$3.3986 - 0.0014i$	$1005\mu m$
PVC	$1.8174 - 0.0309i$	$5010\mu m$
COC	$1.5324 - 0.0008i$	$3058\mu m$

Table 2.1: Reference material and geometric parameters of all four materials in the **MPTarget** datasets: the complex refractive index $\overline{\eta}$ and the measured thickness in μm.

and then the average thickness is calculated as the reference value. Table 2.1 summarizes the reference value of material and geometric parameters in the **MPTarget** datasets.

SYNTHETIC DATASETS The theoretical analysis and evaluation in Chapter 6 are based on these synthetic datasets:

- **SynthMPT**: The ground truth material and geometric parameters are chosen by the reference values of all four dielectric materials in Table 2.1. The THz data is simulated based on the multi-path reflection model (see Sec. 6.2 and (6.8)).

- **SynthMPT$*$Hamm**: The ground truth material and geometric parameters are chosen by the reference values of all four dielectric materials in Table 2.1. The THz data is simulated based on the multi-path reflection model and the Hamming window. More details will be given in Chapter 6.

3

COMPUTATIONAL IMAGE ENHANCEMENT

3.1 MOTIVATION

A wide range of technological approaches to realize THz imaging systems have been demonstrated [HN 95; Coop+ 11; Ding+ 13; Kahl+ 12] in recent years, that frequency modulated THz signals allow to sense temporal or phase shifts to the object's surface, making the 3D THz imaging possible. Despite the fact that in most of these approaches, THz imaging is performed close to the diffraction limit, the comparatively low spatial resolution associated with the long THz radiation wavelengths significantly hampers the application range. There is a huge interest to increase the spatial resolution of THz imaging beyond the diffraction limit, to make this technique competitive in comparison to established methods of other imaging modalities, such as optical or X-ray.

Hence, THz imaging below the diffraction limit is an emerging area [CDM 07], which can roughly be classified into two alternatives: by system enhancements or by computational approaches. System enhancements include for example, interferometric sensing [JDM 01] to increase the depth sensitivity in THz time-of-flight imaging, or near-field sensing approaches [CKC 03] which demonstrate a nanometer scale lateral resolution by circumventing the diffraction limit. Computational image enhancement techniques aim at improving the resolution by utilizing numerical procedures and additional signal or system information, e.g. oversampled THz imaging signals, without introducing additional equipment effort and cost.

Depending on the THz image acquisition mode, there are several approaches for computational image enhancement. THz imaging super-resolution (also referred to as high-resolution or image restoration) is often associated with spatial resolution enhancement in the xy-direction [XFL 14; Li+ 08; Ding+ 10; Hou+ 14; AA 16]. In contrast, depth resolution enhancement is associated with improvement in azimuth direction (z-direction) [Walk+ 12; CHP 10; Taka+ 09].

The majority of research focuses on the lateral resolution of 2D THz images, where the Lucy-Richardson deconvolution algorithm [Lucy 74; Rich 72] is one of the most frequently used methods: Knobloch et al. [Knob+ 02] firstly applied Lucy-Richardson deconvolution on THz images; Li et al. [Li+ 08] proposed to use the Lucy-Richardson deconvolution algorithm for a coherent THz 2D imaging system; Ding et al. [Ding+ 10] used the Lucy-Richardson deconvolution for a THz reflective 2D imaging system; Xu et al. [XFL 14] proposed a THz time-domain spectroscopy im-

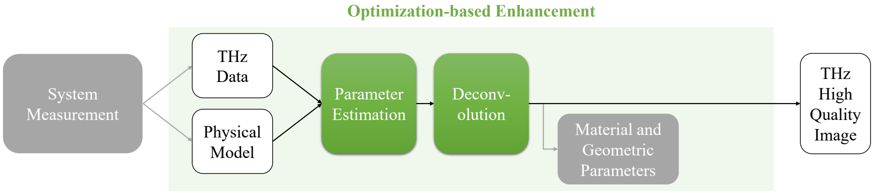


Figure 3.1: The computational THz image enhancement approach concentrates on the optimization-based enhancement, the quality of THz image, the THz data processing, and the physical model (3.3).

age high-resolution reconstruction model incorporating a 2D wavelet decomposition and a Lucy-Richardson deconvolution to reconstruct a high-resolution THz image from four low-resolution images and to reconstruct a high-resolution image from a single degraded 2D low-resolution image.

In this chapter, a novel method to enhance the THz image resolution beyond the diffraction limit is described, attaining a lateral (xy) resolution increase and a depth (z) accuracy increase. The concept is demonstrated for a Frequency Modulated Continuous Wave (FMCW) THz scanning system operating at $514 - 640$GHz, but can certainly be adapted to other THz imaging techniques. This method combines the depth accuracy with the lateral resolution enhancement to achieve a jointly improved spatial resolution and accuracy in both, xy- and z-direction.

As illustrated in Fig. 3.1, this approach concentrates on the optimization based enhancement, the quality of THz image, the THz data processing, and the physical model (3.3) within the overall context of the optimization-based enhancement concept (Fig. 1.1b). The details of the pre-processing and the curve fitting procedure are described in Sec. 3.2.1 and Sec. 3.2.2 respectively, while Sec. 3.2.3 provides details about the initialization of the curve fitting. Sec. 3.2.8 describes the 2D deconvolution procedure for the reconstructed THz intensity image. In Sec. 3.3, the evaluation of the computational result of the method is depicted.

3.2 METHODOLOGY

In the THz 3D imaging system mentioned in [Ding+ 13], the signal was assumed to have an ideal flat target with perfect orthogonal alignment to the THz sensor. However, perfect planarity and orthogonality require high precision in the manufacturing procedure and calibration of the acquisition setup. To study more realistic THz imaging scenarios, *non-planar* targets (see Figs. 2.20 and 2.21), which are not perfectly orthogonally aligned to the sensor, are examined such that the distance between sensor to a lateral pixel in xy-direction is an unknown variable.

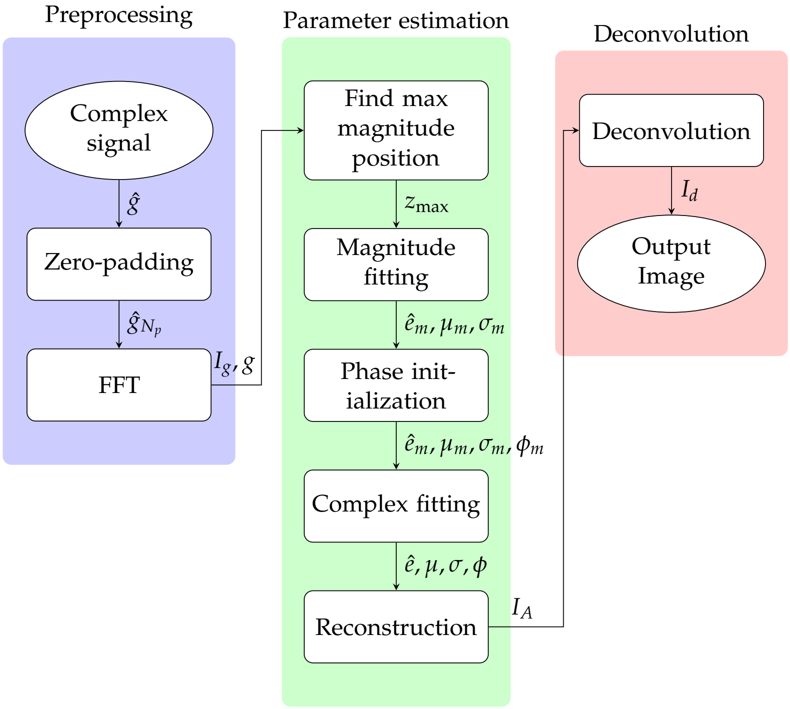


Figure 3.2: Block diagram of the computational enhancement method (see intensity images in Fig. 3.6 and Fig. 3.8).

Fig. 3.2 depicts the computational enhancement method, which comprises three major components: pre-processing, parameter extraction, and deconvolution. In the pre-processing part, the measured complex signal is interpolated by zero-padding to obtain more sampling points in z-direction (see Chapter 2 Sec. 2.2). In the parameter extraction part, a complex model is fitted to the in-phase and quadrature components of the signal in the z-direction for each lateral position. From this fitting, the corrected reflectance complex field signal and depth information is deduced. In the deconvolution part, the reconstructed 2D image is processed by the deconvolution algorithm to form a high resolution image in (x, y)-domain.

3.2.1 *Preprocessing*

As described in the fundamentals (Chapter 2 Sec. 2.2), the measured complex frequency domain signal is zero-padded and converted to the complex spatial domain signal by Fourier Transform. With the initial input data to the computational procedure, a complex THz signal is acquired in a per-pixel manner with frequency ramping [MV 89]. In this chapter, $\hat{g}(x, y, t)$ is denoted as the measured complex signal at lateral position (x, y) with length N_t.

In order to achieve sub-wavelength geometric correction, more sampling points on the z-axis are required for robust curve fitting.

Based on the current acquisition system (see Chapter 2 Sec. 2.5.1), an intuitive method is to interpolate the signal in the spatial domain, but the time domain signal provides another option. Instead of spatial interpolation, the discrete signal is extended by a factor of N_p using zero-padding of the complex electric field signal $\hat{g}(x,y,t)$ in the time domain:

$$\hat{g}_{N_p}(x,y,t) = \begin{cases} \hat{g}(x,y,t), & \text{if } t < N_t \\ 0, & \text{otherwise} \end{cases}, \tag{3.1}$$

where N_p is the zero-padding factor, and the length of $\hat{g}_{N_p}$ is $N_z = N_p \cdot N_t$. In this chapter, $N_p = 9$ is adopted empirically.

After zero-padding, the signal is transformed into the spatial domain by applying a deramp-FFT [MV 89].

$$g(x,y,z) = \mathcal{F}\{\hat{g}_{N_p}(x,y,t)\} \tag{3.2}$$

The resulting 3D image g can be expressed as a 3D matrix in the spatial xyz-domain, representing per-pixel (x,y) the complex reflectivity of THz energy in z-direction represented by the complex samples $g(x,y,z_0) \to g(x,y,z_{N_z-1})$.

3.2.2 *Per-pixel Model Parameter Estimation*

In this part, the per-pixel parameter estimation is applied in the z-direction in order to represent the measured complex signal by a complex reflection model. As each pixel is treated independently, notations are simplified by dropping the pixel-location using, e.g. $g(z_i)$ for $g(x,y,z_i)$.

As described in Chapter 2, the system is calibrated by amplitude normalization with respect to an ideal metallic reflector. In this case, an ideal rectangular frequency amplitude signal response is achieved, which, after being Fourier transformed, results in an ideal sinc function $A(z)$ as continuous spatial signal amplitude. In this model shown in (2.16), the single-path reflection is assumed, and the extension to multi-path reflection will be discussed in Chapter 6. By the single-path reflection assumption, the complex signal model $A(z)$ is modeled as a modulated sinc function:

$$\begin{aligned} & A(\hat{e},\sigma,\mu,\phi|z) = \hat{e} \cdot \operatorname{sinc}\left(\sigma(z-\mu)\right) \cdot \exp\left(-\boldsymbol{i}(\omega z - \phi)\right) \\ & \text{where, } \operatorname{sinc}(t) = \begin{cases} \dfrac{\sin(\pi t)}{\pi t} & t \neq 0 \\ 1 & t = 0 \end{cases} \end{aligned} \tag{3.3}$$

where $\hat{e}$ is the electric field amplitude, μ and σ are the mean (i.e. the depth) and the width of the sinc function, respectively, ω is the center frequency of the sinusoidal carrier and ϕ is the depth-related phase shift. Thus, the parameter extraction is formulated as a complex curve fitting process for optimizing and minimizing the energy function $\mathcal{L}$:

$$\begin{aligned}\mathcal{L} = \arg \min_{\hat{e},\sigma,\mu,\phi} \sum_{z \in Z_f} \Big[& (g_{re}(z) - A_{re}(\hat{e},\sigma,\mu,\phi|z))^2 \\ & + (g_{im}(z) - A_{im}(\hat{e},\sigma,\mu,\phi|z))^2 \Big]\end{aligned} \tag{3.4}$$

where the subscripts *re* and *im* denote the real and the imaginary part of a complex number, respectively, and Z_f is the fitting window (see Sec. 3.2.4).

3.2.3 *Initialization for Non-convex Optimization*

Because of the highly non-linear optimization involved in the curve fitting process, a direct application of (3.4) to a non-linear solver potentially results in local minima and does not lead to robust results. Therefore, the following optimization steps are applied to achieve a robust initialization of the complex curve fitting:

1. Estimate the signal's maximum magnitude z-position z_m in order to localize the *curve fitting window* (see Sec. 3.2.4).
2. Apply a *curve fitting to the magnitude signal* leading to initial values for $\hat{e}_m, \mu_m, \sigma_m$ (see Sec. 3.2.5).
3. Estimate the initial phase value ϕ_m using a *phase matching* with respect to the angle of complex signal $\angle g$ (see Sec. 3.2.6).
4. Based on the initial values $\hat{e}_m, \mu_m, \sigma_m, \phi_m$ the optimization is performed by minimizing the energy $\mathcal{L}$ (see (3.4)) using the Trust Region Algorithm [CL 96].
5. Reconstruct an intensity image I_A and an depth image D_A based on the curve fitting result (sec Sec. 3.2.7).

This curve fitting approach significantly reduces the per-pixel intensity inhomogeneity (see Sec. 3.3.3). The subsequent lateral deconvolution algorithm discussed in Sec. 3.2.8 involves the numerical solution of an ill-posed inverse problem of finding the blur kernel and enhancing the image's sharpness at the same time, which is very sensitive to noise. Therefore, correcting the intensity yields two advantages, i.e., it stabilizes the numerical deconvolution process and prevents wrong interpretations of intensity variation as structural or material transitions. The comparison of lateral resolution will be provided in Fig. 3.8 and discuss the lateral resolution enhancement with and without the per-pixel parameter extraction in Sec. 3.3.4.

3.2.4 *Curve Fitting Window*

As this chapter focuses on the primary reflection signal, assuming that the geometric reflection energy concentrates on the first air-material interface, we locate the z-position that exhibits the maximum magnitude within a fitting window

$$Z_f \in [z_{\max} - \tau_f, z_{\max} + \tau_f] \tag{3.5}$$

with center

$$z_{\max} = \arg\max_{z_i} |g[z_i]|\,, \tag{3.6}$$

i.e., z_m is the maximum magnitude z-position in the complex spatial domain data. τ_f is the half-width of the fitting window. The choice of τ_f is discussed in Sec. 3.3.1.

3.2.5 *Magnitude Curve Fitting*

Since the complex model $A(z)$ in (3.3) is non-linear and the optimization for $\mathcal{L}$ in (3.4) is non-convex, the estimation of the initial parameters is critical to avoid local minima. A reliable initial estimate of the complex curve fitting parameters $\hat{e}_m, \mu_m, \sigma_m$ is deduced from a magnitude curve fitting.

The magnitude signal model $A_m(z)$ is derived from (3.3) and is expressed as

$$A_m(\hat{e}_m, \mu_m, \sigma_m | z) = \hat{e}_m \cdot |\text{sinc}\,(\sigma_m(z - \mu_m))| \tag{3.7}$$

where $\hat{e}_m$ is the electric field amplitude based on signal magnitude, μ_m is the center of sinc function, and σ_m is the width. The magnitude curve fitting minimizes the energy function $\mathcal{L}_m$ by the Trust-Region Algorithm [CL 96]

$$\mathcal{L}_m = \arg \min_{\hat{e}_m, \mu_m, \sigma_m} \sum_{z \in Z_f} \left(|g(z)| - A_m(\hat{e}_m, \mu_m, \sigma_m | z)\right)^2 \tag{3.8}$$

After the magnitude curve fitting, $\hat{e}_m, \mu_m, \sigma_m$ are the initial values for $\hat{e}, \mu, \sigma$ with respect to the complex signal model in (3.3). However, an estimate for the phase angle ϕ_m is still required.

3.2.6 *Estimating the Initial Phase Value ϕ_m*

We assume that within the fitting window Z_f, the phase angle $\omega z - \phi$ in the model (3.3) matches the phase angle $\angle g$ in the spatial domain data $g(z)$. The corresponding optimization energy functional measures the linear deviation between these phase angles as follows

$$\begin{aligned}\mathcal{L}_\phi = \arg\min_{\phi_m} \frac{1}{2} \sum_{z \in Z_f} \Big[& (\cos(\phi_m - \omega z) - \cos(\angle g))^2 \\ & + (\sin(\phi_m - \omega z) - \sin(\angle g))^2 \Big]\end{aligned} \tag{3.9}$$

Setting the gradient of $\mathcal{L}_\phi$ to zero and applying trigonometric identities, we obtain

$$\sin\phi_m \sum_{z \in Z_f} \cos(\omega z + \angle g) = \cos\phi_m \sum_{z \in Z_f} \sin(\omega z + \angle g) \tag{3.10}$$

In (3.10), the initial phase angle ϕ_m is independent from the data angle $\angle g$. Therefore, the minimum to (3.9) is found by solving for ϕ_m in (3.10), yielding

$$\phi_m = \tan^{-1} \frac{\sum_{z \in Z_f} \sin(\omega z + \angle g)}{\sum_{z \in Z_f} \cos(\omega z + \angle g)} \tag{3.11}$$

After this phase initialization, $\hat{e}_m, \mu_m, \sigma_m$ and ϕ_m are given as initial values for $\hat{e}, \mu, \sigma$ and ϕ in the model in (3.3), respectively, and the model is fitted according to the energy function $\mathcal{L}$ in (3.4) using the Trust Region Algorithm [CL 96]. Because of scattering and multi-layer reflection, error exists if we fit in an ideal sinc-function. Therefore, σ is extracted as a varying parameter to indicate the error. The depth parameter μ is evaluated in Sec. 3.3.2. The amplitude parameter $\hat{e}$ is further processed in a 2D approach. The processing method on all other parameters (μ, σ, ϕ) will be investigated in future research.

3.2.7 *THz Image Reconstruction*

In order to extract the per-pixel intensities using the THz model, the reference intensity image I_g is defined based on the input data g as the intensity of the z-slice with the maximum average magnitude:

$$\begin{aligned} I_g(x,y) &= g[z_{mean}] \cdot g[z_{mean}]^* \\ \text{where, } z_{mean} &= \arg\max_{z_i} \frac{\sum_{x,y} |g(x,y)[z_i]|}{N_x N_y} \end{aligned} \tag{3.12}$$

where g^* is the complex conjugate of g and N_x, N_y are the size of the matrix in the x-axis and y-axis, respectively. The *reconstructed intensity* is deduced from the curve fitted data model and is defined as the intensity of the THz signal (3.3) at the center position μ:

$$\begin{aligned} I_A(x,y) &= A(x,y,\mu) \cdot A(x,y,\mu)^* \\ &= \hat{e}^2(x,y) \cdot sinc^2(0) \\ &= \hat{e}^2(x,y) \end{aligned} \tag{3.13}$$

3.2.8 *Deconvolution*

After the curve fitting, the reconstructed intensity I_A is a 2D image with real and positive values. Now, it is possible to apply a state-of-art 2D deconvolution algorithm to enhance the xy-domain resolution. In contrast to prior work, a total variation (TV) blind deconvolution algorithm is applied to improve the spatial resolution [XJ 10; XZJ 13].

In deconvolution, the reconstructed intensity image I_A is expressed as a blurred observation of a sharp image I_d

$$I_A = I_d \circledast h + \varepsilon \tag{3.14}$$

where h is the *spatially invariant* point spread function (PSF, also known as *blur kernel*), ε denotes the noise and $\circledast$ denotes convolution. Blind deconvolution methods allow for estimating the blur kernel directly from the data, which is, however, an ill-posed inverse problem that requires prior knowledge in order to deduce a robust result [Levi+ 11]. In this chapter, the sparse nature of intensity gradients [PF 14] is utilized and a state-of-art TV blind deconvolution algorithm is chosen that minimizes

$$(I_d, h) = \arg\min_{I_d, h} \|I_d \circledast h - I_A\|_1 + \lambda_r \|\nabla I_d\|_1. \tag{3.15}$$

Here, $\|I_d \circledast h - I_A\|_1$ is commonly referred as a *data term*, λ_r is a *regularization parameter* and $\|\nabla I_d\|_1$ is the *TV-regularization* (or *prior*) that enforces the gradient of the resulting deblurred image I_d to be sparse. As a by-product, the blind deconvolution yields the estimated PSF h. The final results are obtained by the implementation from Xu et al. [XJ 10; XZJ 13].

3.3 EXPERIMENTAL RESULT

In this section, the experimental results of the computational image enhancement method are discussed in detail, where the window size (Sec. 3.3.1), the depth accuracy (Sec. 3.3.2), the intensity reconstruction (Sec. 3.3.3), the lateral resolution (Sec. 3.3.4) and the embedded structures (Sec. 3.3.5) are evaluated.

To evaluate the computational image enhancement method, the **MetalPCB** and **StepChart** datasets are used (see Chatper 2 Sec. 2.5.2).

The following images, reconstruction and image enhancement methods are applied and compared, using the naming template as {*image*}_{*method*}–{*kernel*}, where *image* is the intensity image applying deconvolution methods, *method* is the method applied to the intensity image, and *kernel* is the deblur kernel adopted in the deconvolution method.

1. **Refer**: The reference intensity image I_g (3.12); no reconstruction using curve fitting and no image enhancement applied.
2. **Reconst**: The reconstructed intensity image I_A using curve fitting; no image enhancement applied.
3. **Refer_Xu**: image enhancement using Xu et al. [XJ 10; XZJ 13] applied to reference intensity **Refer**.
4. **Reconst_Xu**: image enhancement using Xu et al. [XJ 10; XZJ 13] applied to reconstructed intensity **Reconst**. This is the method described in Sec. 3.2.
5. **Refer_LR-G**: using Lucy-Richardson [Lucy 74; Rich 72] with gaussian kernel applied to reference intensity **Refer**.
6. **Reconst_LR-G**: using Lucy-Richardson [Lucy 74; Rich 72] with gaussian kernel applied to reconstructed intensity **Reconst**.
7. **Refer_LR-Xu**: using Lucy-Richardson [Lucy 74; Rich 72] with sparse kernel extracted from Xu et al. [XJ 10; XZJ 13] applied to reference intensity **Refer**.
8. **Reconst_LR-Xu**: using Lucy-Richardson [Lucy 74; Rich 72] with sparse kernel extracted from Xu et al. [XJ 10; XZJ 13] applied to reconstructed intensity **Reconst**.

In the following, the optimal window size will be firstly deduced for the quality control of the fitting using the **MetalPCB** dataset (Sec. 3.3.1). In Sec. 3.3.2, the depth accuracy will be discussed using the **StepChart** dataset. In Sec. 3.3.4, the lateral resolution will be evaluated on the **MetalPCB** dataset. All intensity images are normalized to a perfect metal reflection and displayed using MATLAB® 's perceptional uniform colormap *parula* [Math 16].

3.3.1 *Window Size Optimization*

In Fig. 3.3, the average intensity of g by the z-axis in the PCB region of the **MetalPCB** dataset is shown. By comparison to the symmetric model in (3.3), the z-axis signal has a lower main lobe to side lobe ratio in the PCB region. This might be due to the superposition of signal reflection from the front and back PCB surfaces (see Chapter 2 Fig. 2.20). This indicates that a too-large fitting window size τ_f in (3.5)

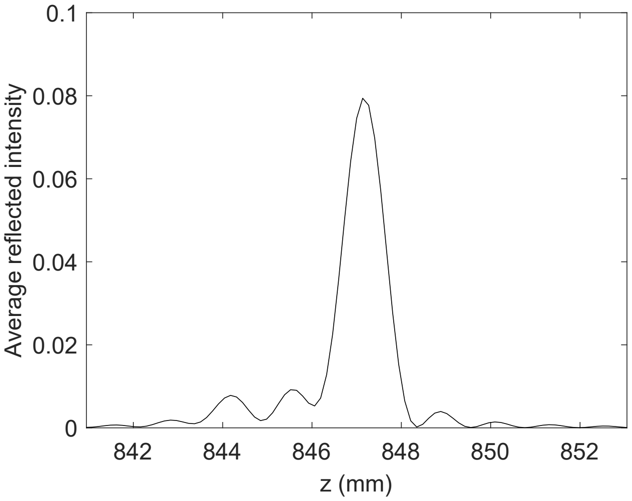


Figure 3.3: Average reflected intensity of g in PCB region.

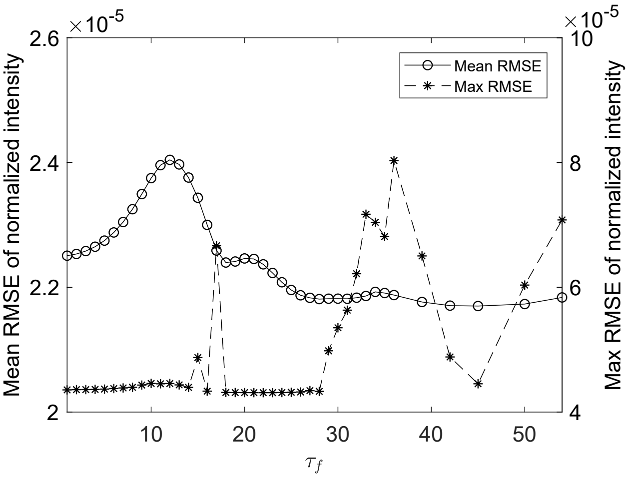


Figure 3.4: Mean RMSE and maximum RMSE by fitting window width τ_f.

would corrupt any quantitative evaluation and should be avoided. On the other hand, a too-small fitting window size is also not feasible, as a sufficient number of sampling points is needed to get a robust fitting result and to avoid over-fitting.

To obtain a reliable numeric measurement, the fitting error is evaluated for varying fitting window half-widths τ_f using the Root-Mean-Square-Error (RMSE) between the fitted model A (3.3) and the measured data g in z-direction:

$$\text{RMSE}(x, y) = \sqrt{\frac{\sum_{z \in Z_f} |g(x, y)[z] - v(x, y, z)|^2}{2\tau_f + 1}} \tag{3.16}$$

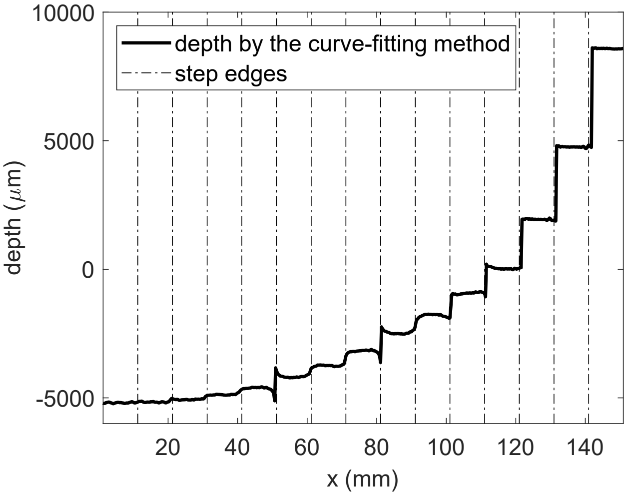


Figure 3.5: Cross section depth of **StepChart** datasets by the curve-fitting method and the step edges.

As a measure for the full THz image, the mean and the maximum RMSE over all pixels are calculated by:

$$\begin{aligned}\text{Mean RMSE} &= \frac{\sum_{x,y} \text{RMSE}(x,y)}{N_x N_y} \\ \text{Max RMSE} &= \max_{x,y}(\text{RMSE}(x,y))\end{aligned} \tag{3.17}$$

In Fig 3.4, the mean and the maximum RMSE of the curve fitting with a different fitting window τ_f are shown. It can be seen that the mean RMSE and maximum RMSE are both increasing when $\tau_f \leq 13$, which is expected due to over-fitting. When the fitting window is increased to a larger value, it can be observed that the mean RMSE is decreased steadily until $\tau_f = 45$. The maximum RMSE, however, has no clear tendency and strongly varies beyond $\tau_f = 28$. After considering that the mean and maximum RMSE are both considerably low when the window size is 45, $\tau_f = 45$ is chosen as the reference and optimal window size, which is used throughout the rest of this chapter.

3.3.2 *Depth Accuracy*

In this part, the depth accuracy is evaluated using the **StepChart** dataset. In comparison to the depth of the curve-fitting method depth$_\mu$, the depth depth$_{\text{max}}$ is obtained using the maximum magnitude position

Depth Difference					Relative Error	
depth_{gt}	$\text{depth}_{\max}$		depth_{μ}		$\text{Error}_{\max}$	Error_{μ}
	Mean	SD	Mean	SD		
(μm)	(μm)	(μm)	(μm)	(μm)	(%)	(%)
4009.0	3898.5	7.2	3831.3	12.4	2.76	4.43
2987.0	2797.2	54.0	2810.8	13.0	6.35	5.90
2006.0	1908.7	53.6	1926.8	19.4	4.85	3.95
1004.0	941.1	0.0	958.6	23.4	6.26	4.53
903.0	806.7	0.0	815.3	17.5	10.67	9.71
803.0	792.8	40.9	742.0	14.0	1.27	7.60
703.0	633.8	77.3	665.5	20.5	9.84	5.33
600.0	590.0	65.6	561.9	19.5	1.66	6.35
472.0	403.3	0.0	468.8	13.9	14.55	0.68
410.0	403.3	0.0	391.0	13.7	1.63	4.64
298.0	268.9	0.0	287.6	14.7	9.77	3.48
208.0	268.9	0.0	192.2	15.1	-29.27	7.60
91.0	17.7	45.5	89.3	17.4	80.58	1.88
42.0	102.2	61.8	34.9	21.9	-143.28	16.84

Table 3.1: Depth difference, depth difference standard deviation (SD) and error comparison between maximum magnitude and the curve-fitting method.

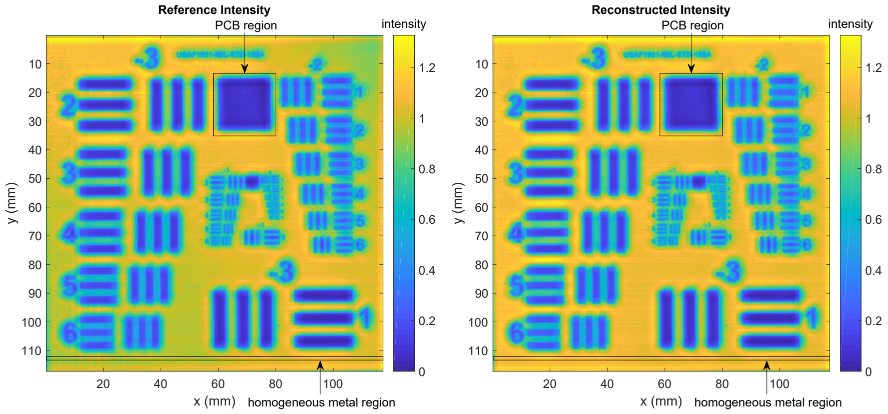
$z_{\max}$ in (3.6) of g. The z-positions are both calibrated to μ by the reference zero z-position z_0:

$$\begin{aligned} \text{depth}_{\mu} &= \frac{\mu - z_0}{N_p} \cdot \Delta d \\ \text{depth}_{\max} &= \frac{z_{\max} - z_0}{N_p} \cdot \Delta d \end{aligned} \tag{3.18}$$

where N_p is the zero-padding factor in (3.1), $\Delta d = 1210\mu m$ is the physical depth per sample, i.e. the system depth resolution in (2.35).

In Fig. 3.5, the cross-section depth of **StepChart** dataset is plotted with an expected position of the edges. It can be observed an interference effect due to signal superposition at several edges, most notably at $x = 50mm$ and $x = 81mm$. Even though blind deconvolution can hardly resolve strong interference effects, a spatially varying point-spread-function would be required in order to cope with this kind of effect in the deconvolution stage; see also Hunsche et al. [Huns+ 98].

To circumvent interference effects, depth values are extracted and averaged from the center 350 xy-samples for each step. Then, the depth differences between adjacent steps are calculated and compared to the ground truth values, which are obtained by mechanical measurement.



(a) (b)

Figure 3.6: Comparison between (a) Reference intensity I_g (**Refer**) (b) Reconstructed intensity I_A (**Reconst**). The homogeneous metal regions for Sec. 3.3.3 and the PCB region for Sec. 3.3.5 are indicated.

Table 3.1 depicts the depth differences and the corresponding standard deviation (SD) of the ground truth depth$_{gt}$, the maximum magnitude method depth$_{\text{max}}$ and the proposed curve fitting depth$_{\mu}$ (3.18). In order to compare the depth accuracy, the relative error is calculated as:

$$\begin{aligned} \text{error}_{\mu} &= \frac{\text{depth}_{\mu} - \text{depth}_{gt}}{\text{depth}_{gt}} \\ \text{error}_{\text{max}} &= \frac{\text{depth}_{\text{max}} - \text{depth}_{gt}}{\text{depth}_{gt}} \end{aligned} \tag{3.19}$$

In this chapter, a depth difference is considered resolvable when the relative error is below 10% with a reasonably low deviation. Thus, the method can still resolve the $91\mu m$ depth difference, while the maximum magnitude method can only resolve the depth difference up to $298\mu m$. As a result, the curve fitting method enhances the system depth accuracy to $91\mu m$.

3.3.3 *Intensity Reconstruction*

In this part, the reconstructed intensity image that is directly deduced from the enhanced depth accuracy according to (3.13) is evaluated by the large homogeneous copper regions in the **MetalPCB** dataset. However, the **MetalPCB** target is not perfectly flat and/or not perfectly aligned orthogonally.

Fig. 3.6 depicts the intensity images for the reference intensity I_g (Fig. 3.6a) based on (3.12) and the reconstructed intensity I_A (Fig. 3.6b) based on (3.13). It can be observed that the copper regions do not

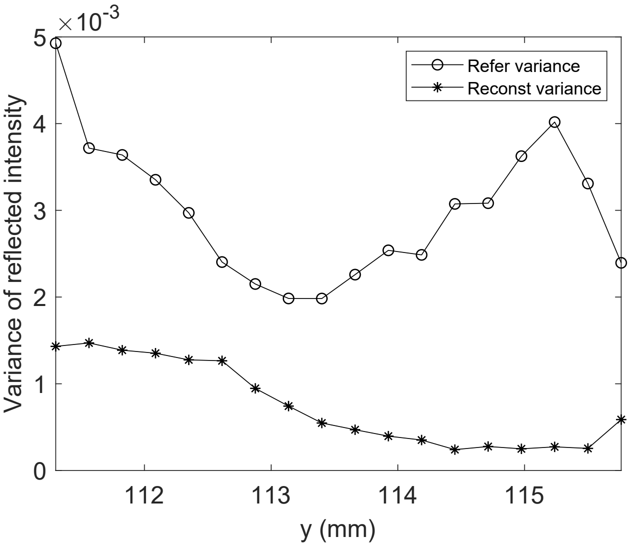


Figure 3.7: Evaluation of the variance of the reflected intensities along the homogeneous metal regions in Fig. 3.6a and Fig. 3.6b in y-direction: For each pixel-line (parameterized over the y-position in mm) we compute the variance of the reference intensity I_g (**Refer**) and the reconstructed intensity I_A (**Reconst**) is given, showing the improvement resulting from the curve fitting in Sec. 3.2.2.

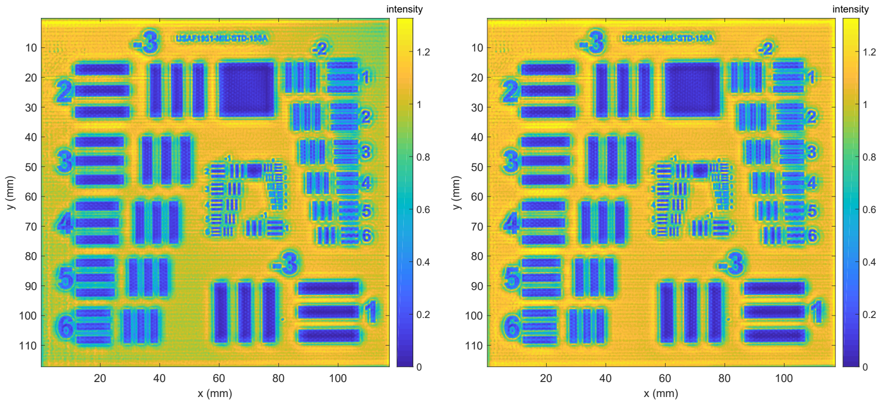
appear to be fully homogeneous in the reference intensity image I_g. After applying the high accuracy depth reconstruction, this intensity inhomogeneity is significantly reduced in the reconstructed image I_A.

Fig. 3.7 further analyzes the intensity homogeneity using a horizontal metal region (see Fig. 3.6, lower image part). The horizontal metal region consists of 18 pixel rows, while the intensity variance of each y-position is computed in the reference intensity image I_g and the reconstructed intensity image I_A. Apparently, the reconstructed intensity I_g has a significantly reduced intensity variance within all rows of copper, as the reconstructed intensity is focused on an accurate depth position for each lateral position.

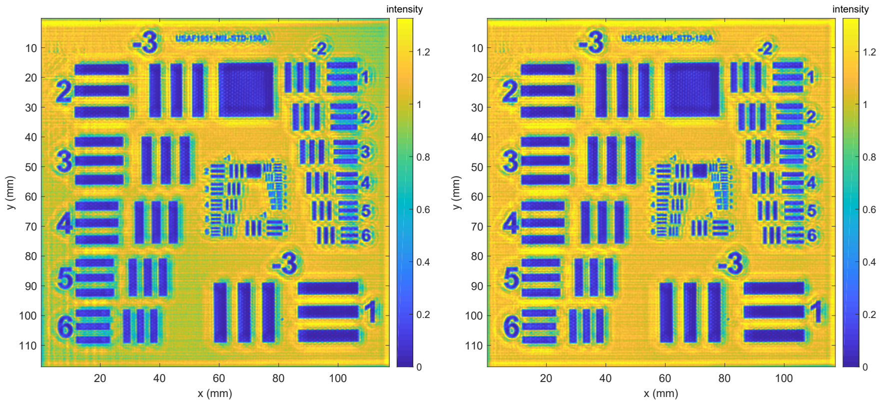
Both, the visual and the numerical results in Fig. 3.6 and Fig. 3.7, respectively, demonstrate that the intensity reconstruction method achieves significantly improved homogeneous intensities on homogeneous material regions without introducing additional distortions or noise.

3.3.4 *Lateral Resolution and Analysis*

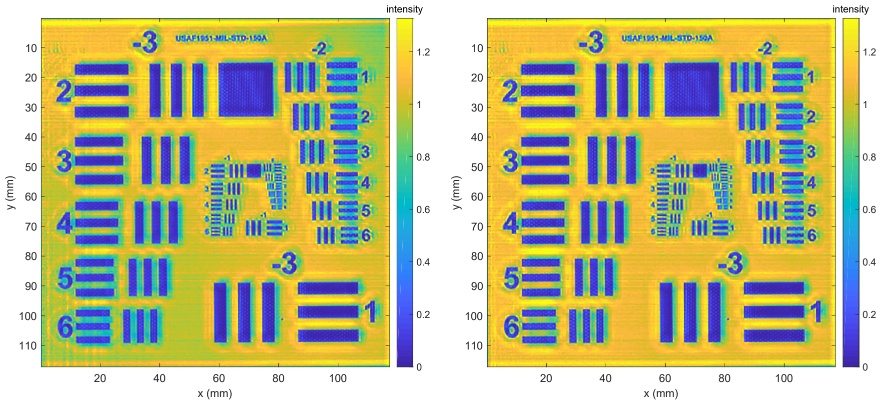
In this section, the lateral domain enhancement by the enhancement method is analyzed using the **MetalPCB** dataset. In Sec. 3.3.4, the enhancement method is compared to Lucy-Richardson (LR) deconvolution method [Lucy 74; Rich 72] on behalf of the lateral resolution and in Sec. 3.3.5 the silk structure embedded in the PCB region is visually analyzed.



(a) **Refer_LR-G** (b) **Reconst_LR-G**

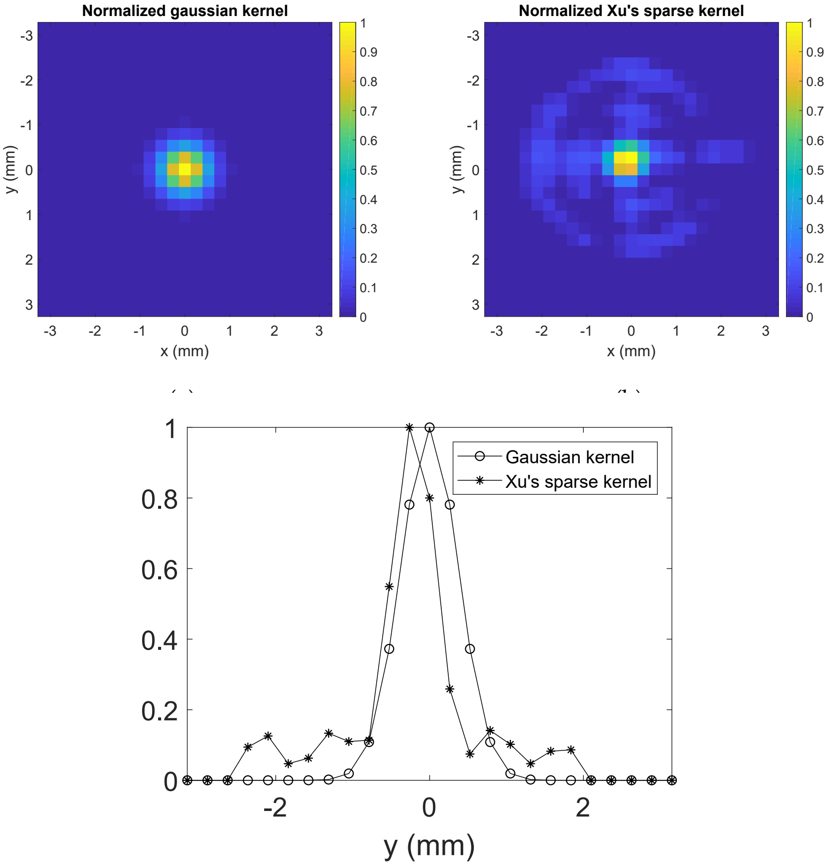


(c) **Refer_LR-Xu** (d) **Reconst_LR-Xu**



(e) **Refer_Xu** (f) **Reconst_Xu** (the enhancement method)

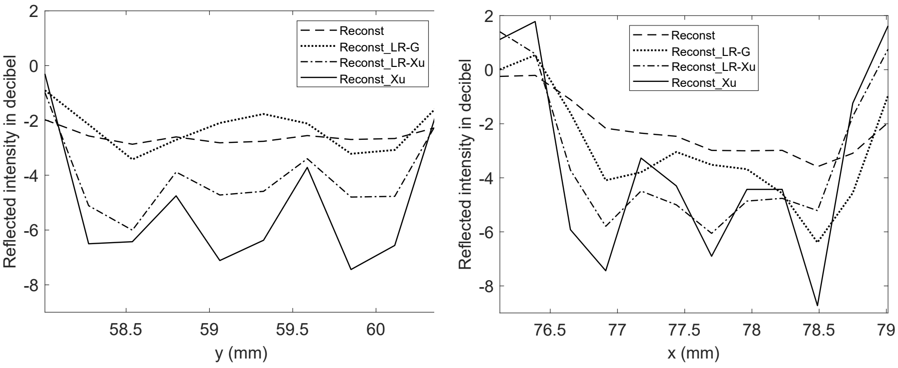
Figure 3.8: Comparison between the three deconvolution methods ((a)(b) Lucy-Richardson (LR) [Lucy 74; Rich 72] with gaussian kernel, top row ; (c)(d) Lucy-Richardson with Xu's kernel, middle row ; (e)(f) Xu et al. (Xu) [XJ 10; XZJ 13], bottom row) applied to the reference intensity image I_g (left column) and the reconstructed intensity image I_A (right column). (f) depicts the Xu deconvolution method on the reconstructed intensity image using sparse kernel (the enhancement method).



(c)

Figure 3.9: Comparison between (a) gaussian kernel (b) Xu's sparse kernel (c) cross-section of Gaussian and Xu's sparse kernel

In Fig. 3.8, different output intensity images are shown with respect to the input intensity image, the deconvolution method, and the blur kernel. The left column shows the deconvolution results based on the reference intensity image I_g, whereas the right column depicts the results using the reconstructed intensity image I_A. For deconvolution, different deconvolution methods are applied: Xu et al.'s method [XJ 10; XZJ 13] (**Refer_Xu**, **Reconst_Xu**, top row) and Lucy-Richardson (LR) deconvolution method [Lucy 74; Rich 72] using a gaussian kernel based on the theoretical numeric aperture (Fig. 3.9a) (**Refer_LR-G**, **Reconst_LR-G**, center row). The LR deconvolution method with a Gaussian kernel based on the system numerical aperture is a commonly applied THz deblurring method [XFL 14] [Li+ 08] [Ding+ 10]. Furthermore, the sparse kernels estimated by Xu et al.'s method (Fig. 3.9b) are extracted and are plugged into the LR deconvolution method (**Refer_LR-Xu**, **Reconst_LR-Xu**, bottom row). Note that **Reconst_Xu** (Fig. 3.8f), is equal to I_d, is the intensity image by the enhancement method described in this chapter.



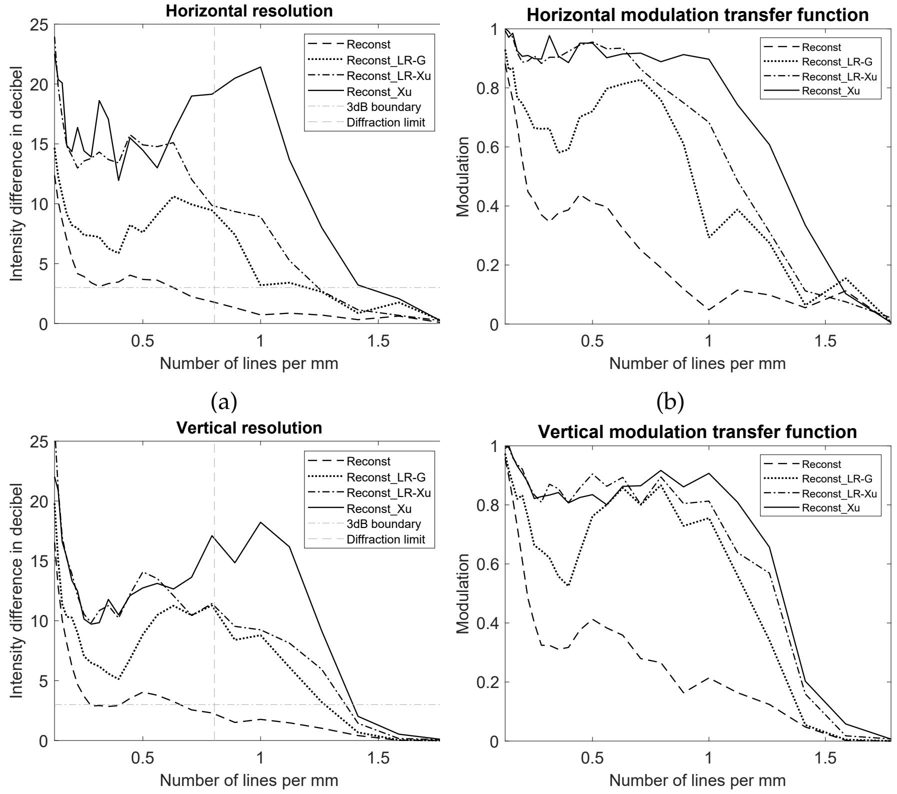
(a) (b)

Figure 3.10: Cross section intensity comparison at Group 0 Element 4 (see Fig. 2.20) for different deconvolution methods. This element represents a $353.6\mu m$ line distance in (a) vertical direction (b) horizontal direction

In Fig. 3.9a and Fig. 3.9b, the Gaussian kernel and Xu's sparse kernel computed by blind deconvolution are shown respectively. Obviously, the kernel estimated by Xu's method models further effects related to the imaging system and/or the observed target that are not covered by the Gaussian kernel. By observation of the kernel cross-section in Fig. 3.9c, Xu's sparse kernel is not centered in the y-axis since the blind deconvolution does not assume any symmetric and centralized property during kernel optimization, but instead, the kernel weights are optimized as fully independent parameters throughout the process.

By visual comparison, it is clear that **Refer_LR-G**, **Reconst_LR-G** (Lucy-Richardson with the Gaussian kernel), which has previously been used for THz resolution enhancement, yields inferior results in terms of sharpness and local contrast. Xu's method that estimates the blur kernel from the given intensity image yields much sharper images with improved local contrast (**Refer_Xu**, **Reconst_Xu**). Using Xu's kernel instead of the standard Gaussian kernel clearly improves the results obtained by the LR method (**Refer_LR-Xu**, **Reconst_LR-Xu**). On the downside, Xu's method and the LR method with Xu's kernel increase noise and overshooting effects. Xu's method result is, however, less affected by these artifacts. Apparently, all three methods benefit from the enhanced intensity image using the curve-fitting reconstruction method.

Moreover, Fig. 3.10 depicts the resolution enhancement using a cross-section of group 0 element 4 of the USAF test target (see Fig. 2.20), representing a line distance of $353.6\mu m$. The cross section intensities before deconvolution (**Reconst**) and after deconvolution (**Reconst_LR-G**, **Reconst_Xu**, **Reconst_LR-Xu**) are shown respectively (refer to Fig. 3.6b, 3.8b, 3.8f and 3.8d). Only the enhancement method **Reconst_Xu** can resolve this resolution according to the 3 dB criterion. Note that there are different definitions for resolution as discussed



(a) (b)

(c) (d)

Figure 3.11: Comparison of intensity difference and modulation transfer function (MTF) by methods to dimensions (a) horizontal intensity difference (b) horizontal MTF (c) vertical intensity difference (d) vertical MTF

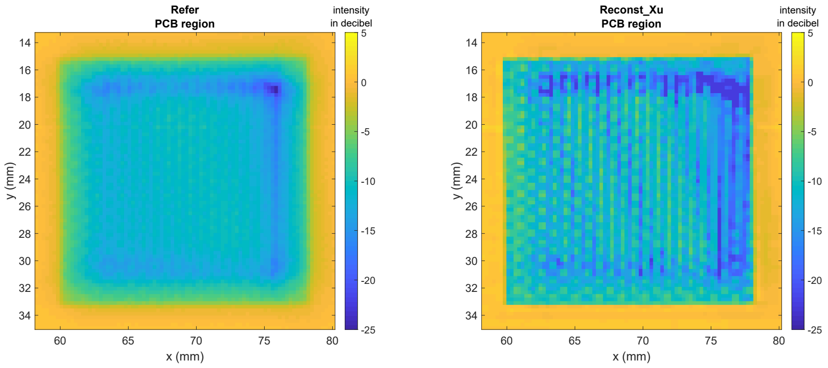
in [Fors+ 83]. However, the center position of **Reconst_LR-Xu**, **Reconst_Xu** are both shifted, because the sparse kernel Fig. 3.9b is not centered in the middle. Although blind deconvolution does not retain the absolute intensity position, the relative intensity positions are constant because a spatial invariant blur kernel is assumed in this chapter.

In order to quantify the lateral resolution, the intensity contrasts are evaluated at each of the horizontal and vertical resolution patterns of the **MetalPCB** dataset (USAF target). In the case of vertical stripes, the minimal and maximal intensity values are determined for each row crossing the pattern's edges given its known geometric structure. After removing 10% of the cross sections in the boundary area, the mean value is computed as the intensity difference (in dB). Analogously, the intensity differences are computed for the horizontal stripes. Fig. 3.11 shows the intensity difference by the number of lines per mm and the modulation transfer function (MTF) [Bore 01; Smit 66] for all methods for vertical and horizontal resolutions. Commonly, to avoid simple enhancement by linear multiplication, a logarithmic measurement over 3 dB intensity difference is considered as a resolution boundary.

Method	Figure	Horizontal resolution (μm)	Horizontal improvement	Vertical resolution (μm)	Vertical improvement
Reconst (I_A)	Fig. 3.6b	794.3	–	762.0	–
Reconst_LR-G	Fig. 3.8b	419.4	1.89	393.3	1.94
Reconst_LR-Xu	Fig. 3.8d	402.3	1.97	368.6	2.07
Reconst_Xu (I_d)[1]	Fig. 3.8f	**346.2**	**2.29**	**359.6**	**2.12**

[1] **Reconst_Xu** is the intensity image I_d by the enhancement method

Table 3.2: Lateral resolution enhancement based on the 3 db criterion using the USAF test target. The best results are highlighted.



(a)

(b)

Figure 3.12: Comparison between (a) PCB region of reference intensity I_g (in decibel) (b) PCB region of the deconvoluted intensity I_d (in decibel).

Table 3.2 depicts the highest resolution at which each method delivers $\geq 3\ dB$. Notice, that there is a resolution improvement from the non-deconvoluted image I_A (**Reconst**), via the original LR method (**Reconst_LR-G**) and the one using Xu's kernel (**Reconst_LR-Xu**), finally, to the enhancement method (**Reconst_Xu**). Taking the $3\ dB$ limit as the boundary, horizontal improvement factors (in terms of resolution) of 2.29 is found for the curve-fitting method **Reconst_Xu** and of 1.89 and 1.97 for the LR method **Reconst_LR-G** and the improved LR method **Reconst_LR-Xu**, respectively.

For the vertical resolution, the enhancement method has a vertical improvement factor of 2.12, while the LR method and the improved LR method have a vertical improvement factor of 2.07, respectively. For the MTF, Fig. 3.11b and Fig. 3.11d show a significantly higher contrast for the enhancement method (**Reconst_Xu**) compared to both, the LR method (**Reconst_LR-G**) and improved LR method (**Reconst_LR-Xu**).

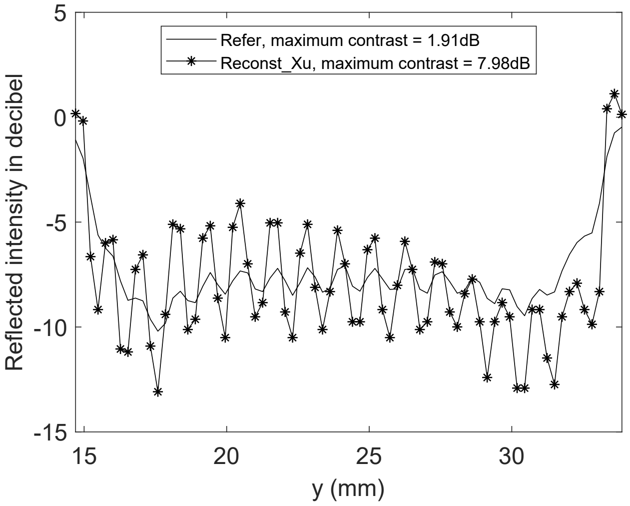


Figure 3.13: Comparison of cross-section intensity in PCB region.

3.3.5 *Embedded Structures*

In this part, the PCB region on **MetalPCB** dataset is analyzed more closely to investigate the effect of the enhanced lateral resolution on the embedded silk structures.

In Fig. 3.12, the PCB region of the reference intensity I_g **Refer** (see Fig. 3.6a) and the deconvoluted intensity I_d **Reconst_Xu** (see Fig. 3.8f) are shown. It can be observed that a periodic intensity pattern is visible by the enhanced lateral resolution that is caused by woven silk material embedded in the PCB region. The silk fibers introduce a second energy reflection to the imaging system.

In Fig. 3.13, the cross-section intensity of the PCB regions in the decibel scale is shown. Extracting the maximum contrast within the PCB region we find an enhancement from $1.91dB$ to $7.98dB$. Even though the enhancement method nicely emphasizes the periodic silk structure underneath the polymer surface material, the depth of this embedded structure cannot be extracted by this approach, as the handling of multi-reflection effects is beyond the scope of this chapter.

3.4 SUMMARY

In this chapter, a THz computational image enhancement method is described to enhance the lateral resolution and depth accuracy beyond the diffraction limit. This enhancement method comprises a parameter extraction by the complex signal fitting model in the z-direction for each pixel which allows for the acquisition of *non-planar* targets. This parameter extraction incorporates an accurate estimation of the per-pixel distance to the object surface, and a proper reconstruction of the reflection intensity as a measure of the object's material properties. Based on the accurately reconstructed reflection intensity, this approach allows applying state-of-art 2D blind deconvolution techniques to improve the spatial xy-resolution beyond what is achievable with traditional (e.g. gaussian kernel) deconvolution procedures.

The experiment results show that the curve-fitting method enhances the depth accuracy to $91\mu m$. Because of this enhanced depth accuracy, the experiment also shows that the curve-fitting reconstruction method achieves improved intensities on homogeneous, non-planar material regions without introducing additional distortions or noise.

Based on the reconstructed intensities, several lateral blind deconvolution methods are applied. Evidently, all three examined THz deconvolution methods benefit from the enhanced intensity image using the curve-fitting method. In comparison to the classical Lucy-Richardson deconvolution algorithm, the experiments show that this enhancement method achieves the best horizontal resolution enhancement from $794.3\mu m$ to $346.2\mu m$, yielding an improvement factor of 2.29. In terms of intensity contrast, this enhancement method clearly outperforms other approaches. Moreover, the experiments show that this enhancement method can emphasize fine silk texture embedded within a polymer material.

4

MODEL-BASED AUTOENCODER

4.1 MOTIVATION

In the THz imaging system, the physically interpretable quantities relevant to the physical structure (i.e. the imaged object) cannot always be measured directly. Instead, as shown in Chapter 2 and 3, each pixel contains implicit information about such quantities, making the *inverse problem* of inferring these physical quantities a challenging problem with high practical relevance and interest in a wide variety of applications.

By reformulating the model equation in Chapter 3 (3.3), in this chapter $\vec{x}$ denotes the lateral x, y position. At each pixel location $\vec{x}$, the relation between the desired (i.e. *unknown*) model parameters u, i.e. the electric field amplitude $\hat{e}$, the position of the surface μ, the width of the reflected pulse σ, and the phase ϕ, and the actual measurements $g(\vec{x}) \in \mathbb{C}^{n_z}$ is modelled via the equation:

$$\begin{aligned} g(\vec{x}, z) &= (A_u(z_i))_{i \in \{1,\dots,n_z\}} + \varepsilon(\vec{x}) \\ A_u(z) &= \hat{e} \cdot \operatorname{sinc}(\sigma(z-\mu)) \cdot \exp(-i(\omega z - \phi)) \\ \text{where, } \operatorname{sinc}(t) &= \begin{cases} \dfrac{\sin(\pi t)}{\pi t} & t \neq 0 \\ 1 & t = 0 \end{cases}, \\ \text{and, } u(\vec{x}) &= (\hat{e}(\vec{x}), \sigma(\vec{x}), \mu(\vec{x}), \phi(\vec{x})) \in \mathbb{R}^4 \end{aligned} \tag{4.1}$$

and ε denotes the noise and $z_{\text{grid}} = (z_i)_{i \in \{1,\dots,n_z\}}$ is the device-dependent sampling grid in range direction. Thus, the crucial step in THz imaging is to find the solution to the optimization problem in Chapter 3 (3.4) of the form:

$$\min_u \quad \mathcal{L}(A_u(z_{\text{grid}}), g(\vec{x})) \tag{4.2}$$

at each pixel $\vec{x}$, where $\mathcal{L}$ denotes the loss function, possibly along with additional regularizers on the unknown parameters.

To solve these inverse problems, the microwave and THz communities have studied for the past two decades, with the widespread use of classical, mainly gradient-based optimization approaches for the material parameters estimation [DBM 01; BFD 05; Requ+ 06]. Other recent works [Bose+ 19; Burg+ 19; BZR 18; Clar+ 18] have proposed to use second-order gradient-based optimization methods. However, as the classical optimization methods are discussed in Sec. 3.2, even with simple choices of the loss function $\mathcal{L}$ such as an ℓ^2-squared loss, the resulting fitting problem is highly non-convex and global solutions become rather expensive. Considering that the number $(N_x \cdot N_y)$ of pixels, i.e. of optimization problem (4.2) to be solved, typically is in the order of hundred thousand to millions, even local first order or quasi-Newton methods become quite costly: For example, running the build-in Trust Region optimizer [CL 96] of MATLAB® to reconstruct a 446×446 THz image takes over 170 minutes.

Due to the revolutionary success (convolutional) neural networks have had on computer vision problems over the last decade, researchers have extended the fields of applications of neural networks significantly. A particularly interesting concept is to learn the solution to complex, possibly non-convex, optimization problems. Different lines of approaches have been considered: directly learning the optimizer itself [Andr+ 16; Kobl+ 17], including optimization problems in the network architecture [AK 17], combining optimizers with networks [Chan+ 17; Mein+ 17]. Möller et al. [MMC 19] have trained a network to predict descent directions to a given energy function to give provable convergence results on the learned optimizer. Objectives that are similar to the one arising in the training of the model-based autoencoders are considered, for instance, for solving inverse problems with deep image priors [UVL 18a] or deep decoders [HH 19]. However, these works use the fixed random noise as the input to the networks and solve an optimization problem for the network's weights for each inverse problem, such that these methods are regularization-by-parametrization approaches rather than learned optimizers.

Another related work is the 3D face reconstruction network from Tewari et al. [Tewa+ 17], which aimed at finding a semantic code vector from a given facial image such that feeding this code vector into a rending engine yields an image similar to the input image itself. While this problem had been addressed using optimization algorithms a long time ago [BV 99] (also known under the name of analysis-by-synthesis approaches), the approach by Tewari et al. [Tewa+ 17] replaced the optimizer with a neural network and kept the original cost function to train the network in an unsupervised way. The resulting structure resembles an autoencoder, where the decoder is fixed to the forward model and was therefore coined model-based AE. Hence, the idea of model-based autoencoders generalizes far beyond 3D face reconstruction and can be used to boost the THz parameter identification problem significantly.

In this chapter, an approach to train a neural network as an optimizer to solve the per-pixel optimization problem (4.2) directly is discussed. The training of the neural network is formulated as a model-based autoencoder (AE), which allows the training of the corresponding network with realistic data in an unsupervised way, i.e. without ground truth.

As depicted in Fig. 4.1, this approach closely relates to the network-based optimizer itself, including the quality, the robustness, and the

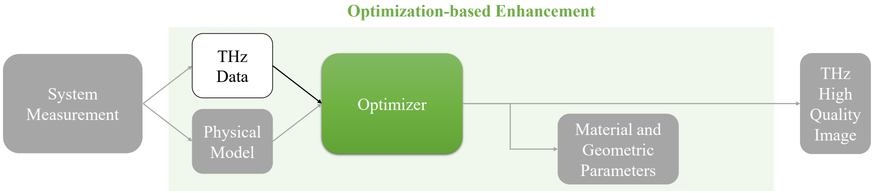


Figure 4.1: The model-based autoencoder approach focuses on the network-based optimizer itself, including the quality, the robustness, and the speed regarding its ability to find the global minima.

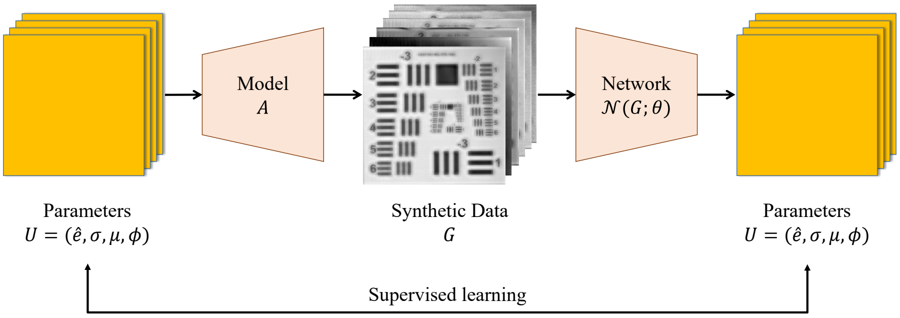


Figure 4.2: Classical supervised learning strategy with simulated data: the forward model A (e.g. from (4.1)) is used to simulate synthetic data G, which can subsequently be fed into a network $\mathcal{N}$ to be trained to reproduce the simulation parameters in a supervised way.

speed regarding its ability to find the global minima within the overall context of the optimization-based enhancement concept (Fig. 1.1b). Sec. 4.2 describes the model-based autoencoder in contrast to classical supervised learning approaches in detail, before Sec. 4.3 summarizes the implementation details. Sec. 4.4 compares the model-based autoencoder approach to classical optimization-based reconstruction techniques in terms of speed and accuracy.

4.2 METHODOLOGY

By considering the real and imaginary parts of THz data $g \in \mathbb{C}^{N_x \times N_y \times N_z}$ in (4.1) as two separate channels, the THz data is denoted by a 4D real data tensor $G \in \mathbb{R}^{N_x \times N_y \times N_z \times 2}$. It indicates that the four unknown parameters $u = (\hat{e}, \sigma, \mu, \phi)$ are $\mathbb{R}^{N_x \times N_y}$ matrices, allowing each parameter to change at each pixel.

Under slight abuse of notation, all operations in (4.1) can be interpreted to be pointwise and again identify complex values with two real values in order to have $A_u(z_{\text{grid}}) \in \mathbb{R}^{N_x \times N_y \times N_z \times 2}$. Concatenating all four matrix valued parameters into a single parameter tensor $U \in \mathbb{R}^{N_x \times N_y \times 4}$, the objective can be formalized as finding U such that $A_U(z_{\text{grid}}) \approx G$.

In Fig. 4.2, a classical supervised machine learning approach to problems with known forward operator is illustrated for the example

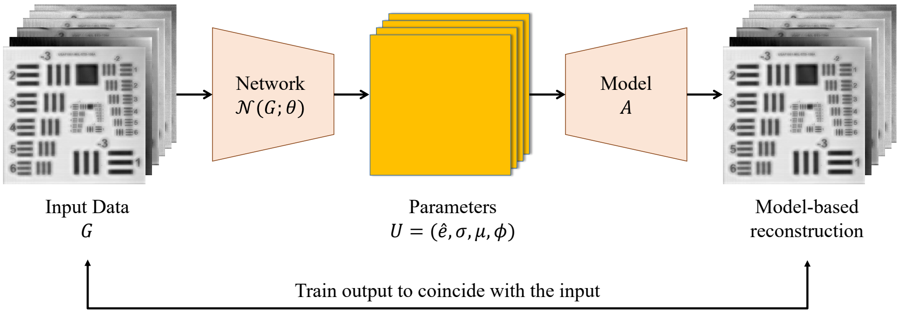


Figure 4.3: A model-based autoencoder (AE) for THz image reconstruction: the input data G is fed into a neural network $\mathcal{N}$ whose parameters θ are trained in such a way that feeding the neural network's prediction $\mathcal{N}(G;\theta)$ into a model function A again reproduces the input data G. Such an architecture resembles an AE with a learnable encoder and a model-based decoder and allows unsupervised training on measurement data.

of THz image reconstruction: the explicit forward model A is used to simulate a large set of images G from known parameters U, which can subsequently be used as training data for predicting U via a neural network $\mathcal{N}(G;\theta)$ depending on weights θ.

Such supervised approaches with simulated training data are frequently used in other image reconstruction areas, e.g. super resolution [Dong+ 14; KKM 16], or image deblurring [NHM 17; Schu+ 16]. The accuracy of networks trained on simulated data, however, crucially relies on precise knowledge of the forward model and the simulated noise. Slight deviations thereof can significantly degrade a network performance as demonstrated in [PR 17], where deep denoising networks trained on Gaussian noise were outperformed by the image denoising algorithm of block-matching and 3D filtering such as BM3D [Dabo+ 07] when applied to realistic sensor noise.

Instead of pursuing the supervised learning approach described above, the desired model parameters $u = (\hat{e}, \sigma, \mu, \phi)$ are replaced in the optimization approach (4.2) by a suitable network $\mathcal{N}(G;\theta)$ that depends on the input data G and learnable parameters θ, such that the neural network $\mathcal{N}$ can be trained in an *unsupervised* way *on realistic measurement data*.

Therefore, assuming multiple examples G^b of THz data are given based as training batch b, and choosing the loss function in (4.2) as an ℓ^2-squared loss, gives rise to the unsupervised training problem

$$\min_{\theta} \sum_{b} \|A\left(\mathcal{N}(G^b;\theta)|z_{\text{grid}}\right) - G^b\|_2^2. \tag{4.3}$$

As illustrated in Fig. 4.3, this training resembles an autoencoder architecture: the input to the neural network is data G^k which gets mapped to model parameters U that – when fed into the model function A – ought to reproduce G^k again.

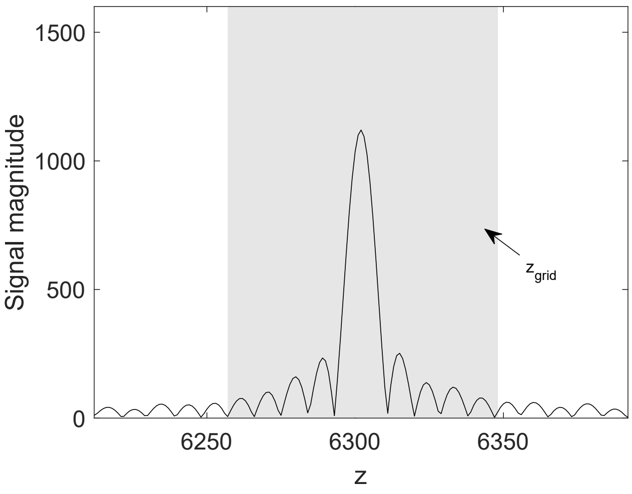


Figure 4.4: Magnitude of a sample point of measured THz signal. The main lobe and major side lobes are included in the grid window, which is colored gray.

Opposed to the straightforward supervised learning approach, the model-based AE approach (4.3) has two significant advantages:

- It allows to train the neural network in an *unsupervised* way, i.e. on realistic measurement data, and therefore learn to deal with measurement-specific distortions.
- The cost function in (4.3) implicitly handles the scaling of different parameters, and therefore circumvents the problem of defining meaningful cost functions on the parameter space: simple parameter discrepancies such as $||U_1 - U_2||_2^2$ for two different parameters sets U_1 and U_2 largely depend on the scaling of the individual parameters and might even be meaningless, i.e. for cyclic parameters such as the phase offset ϕ.

4.3 IMPLEMENTATION

In this section, the implementation of the model-based autoencoder is described in detail. Sec. 4.3.1 includes the data processing for rearranging the THz data into a 4D real tensor. Sec. 4.3.2 describes the encoder neural network architecture and the training procedure.

4.3.1 *Data Preprocessing*

As illustrated in the plot of the magnitude of an exemplary measured THz signal shown in Fig. 4.4, the THz energy is mainly focused in the main lobe and first side-lobes of the sinc function. Because the physical model remains valid in close proximity to the main lobe only, the THz data is pre-processed to reduce the large range of 12600 samples (due

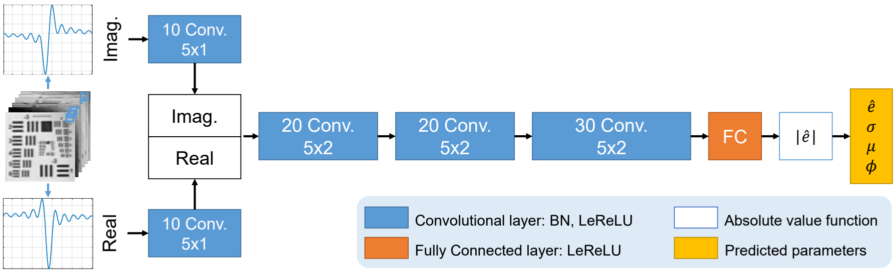


Figure 4.5: Architecture of encoder CNN network $\mathcal{N}(G;\theta)$ that predicts the parameters: At each pixel, the real and imaginary part is extracted, convolved, concatenated and processed via three convolutional layers (blue box, conv.) and one fully connected layer (orange box, FC). To obtain physically meaningful (non-negative) amplitudes, an absolute value function is applied to the first component.

to zero-padding) per pixel (see Chapter 2 Sec. 2.5.2). Therefore, 91 samples (covering 4 side-lobes) per-pixel centered around the main lobe are cropped out, where the position is related to the object distance and the parameter μ (see Chapter 3 Sec. 3.3.1 for details of the cropping window). Hence, the THz data is represented by a 4D real tensor $G \in \mathbb{R}^{N_x \times N_y \times N_z \times 2}$, where $N_x = N_y = 446$, and N_z is the size of the cropping window, i.e. $N_z = 91$ in this case.

4.3.2 *Encoder Architecture and Training*

For the encoder network $\mathcal{N}(G;\theta)$, a spatially decoupled Convolutional Neural Network (CNN) architecture is selected using 1×1 convolutions on G only, leading to a signal-by-signal reconstruction mechanism that allows a high level of parallelism and therefore maximizes the reconstruction speed on a GPU. The specific CNN architecture (illustrated in Fig. 4.5) applies a first set of convolutional filters on the real and imaginary parts separately, before concatenating the activations and applying three further convolutional filters on the concatenated structure. Batch Normalization (BN) [IS 15] is applied after each convolution and Leaky Rectified Linear Units (LeReLU) [GBB 11] are used as activation functions. Finally, a fully connected (FC) layer reduces the dimension to the desired size of four output parameters per pixel. To ensure that the amplitude is physically meaningful, i.e. non-negative, an absolute value function is applied on the first parameter i.e. $\hat{e}$. Interestingly, this choice compared favorably to a plain rectified linear unit when the neural network is trained.

The network optimizing (4.3) is trained by the Adam optimizer (Adaptive Moment estimation, [KB 14]) on 80% of the 446×446 pixels from a measured THz image for 1200 epochs. The remaining 20% of the pixels serve as a validation set. The batch size is set to 4096. The initial learning rate is set to 0.005, and is reduced by a factor of 0.99 every 20 epochs. Fig. 4.6 illustrates the decay of the training and

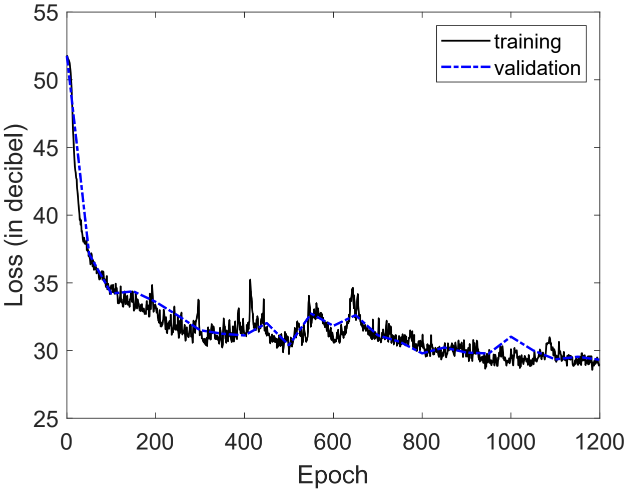


Figure 4.6: The average losses of the training and validation sets over 1200 epochs on a decibel scale illustrate that there is almost no generalization gap between training and validation.

validation losses over 1200 epochs. The result shows that the validation loss resembles the training loss with almost no generalization gap.

4.4 EXPERIMENTAL RESULT

In this section, the experimental results are discussed in detail. The optimal loss and timing (Sec. 4.4.1), the intensity image quality, and cross-section (Sec. 4.4.2) achieved by the mode-based AE are compared to classical optimizers.

The model-based AE is trained on the **MetalPCB** datasets only, while the parameter inference is made for both the **MetalPCB** and **StepChart** datasets. This cross-referencing between two datasets can verify whether the model-based AE method is modeling the physical behavior of the system without overfitting to a specific dataset or recorded material.

To compare with the classical optimization methods discussed in Chapter 3 Sec. 3.2, similarly, the model parameters are also estimated using the Trust-Region Algorithm (TRA) [CL 96], which is implemented in MATLAB® . The TRA optimization requires a proper definition of the parameter ranges. Furthermore, it is very sensitive to the initial parameter set. Therefore, it is required to carefully select the initial parameters by sequentially estimating them from the source data (see Chapter 3 Sec. 3.2.3 for more details). Still, the optimization may result in a parameter set with significant loss values (see Sec. 4.4.2).

Regarding the parameter initialization of the network as an optimizer, the trained encoder network is independent of any initialization scheme as it tries to directly predict optimal parameters from the input

Dataset (Region)	Measurement	TRA	AE	AE+TRA
MetalPCB (All)	Average Loss	693.9	886.3	**442.2**
MetalPCB (PCB)	Average Loss	**589.0**	872.6	**589.0**
MetalPCB (Metal)	Average Loss	519.6	446.1	**115.7**
StepChart (All)	Average Loss	3815.1	5148.3	**3675.3**
StepChart (Edges)	Average Loss	4860.4	6309.1	**2015.7**
StepChart (Steps)	Average Loss	1152.5	2015.7	**1150.3**
MetalPCB	Training time (sec.)	**none**	9312.8	9312.8
MetalPCB	Runtime (sec.)	10391.2	†**73.5**	*4854.7
StepChart	Runtime (sec.)	3463.9	†**22.8**	*1712.4

† Inference time

* Runtime is the sum of AE inference and TRA optimization time

Table 4.1: Loss and timing enhancement based on the model-based autoencoder

data. While the network alone gives remarkably good results with significantly lower runtimes than the optimization method, there is no guarantee that the network predictions are critical points of the energy to be minimized. This motivates the use of the encoder network as an initialization scheme to TRA, specifically because the TRA guarantees the monotonic decrease of objective function such that using TRA on top of the network can only improve the results. This approach is abbreviated to *AE+TRA* for the rest of this chapter.

To fairly compare all three approaches, the optimization time of TRA and the inference time of AE are both recorded by an Intel® i7-8700K CPU computation, while AE is trained on a NVIDIA® GTX 1080 GPU. The PyTorch source code is available at `https://github.com/tak-wong/THz-AutoEncoder`.

4.4.1 *Loss and Timing*

In order to evaluate the optimization quality on different materials and structures, the **MetalPCB** dataset is evaluated in three regions: The *PCB region* is a local region that contains PCB material only, the *Metal region* is a local region that contains copper material only, and the *All region* is the entire image area. Similarly, the **StepChart** dataset is evaluated in three regions: The *Edge region* is the region that contains physical edges, the *Steps region* is the center planar region of each step, and the *All region* is the entire image area. This segmentation is done, because the THz measurements of the highly specular aluminum target result in strong multi-path interference artifacts at the edges that should be investigated separately.

In Table 4.1, the average loss regarding (4.3) and the timing are shown for the Trust-Region Algorithm (TRA), the model-based Autoencoder (AE) and the combined AE+TRA approaches, respectively.

The result shows that the model-based AE achieves a lower average loss than the TRA method in the metal region of the **MetalPCB** datasets, but it yields higher average losses than the TRA on both datasets. It is encouraging to see that although the AE was trained on the **MetalPCB** datasets, the relative performance in comparison to the TRA does not decay too significantly when changing to an entirely unseen data set with different material and different range characteristics, with the AE loss being 21.7% and 25.9% higher than the TRA loss on the **MetalPCB** and **StepChart** datasets, respectively.

If such a sacrifice in accuracy is acceptable, the speed-up in runtime is very significant with the AE being over 140 times faster than the TRA (for both methods being evaluated on a CPU). Note that even the sum of training and inference time is smaller for the AE than the runtime of the TRA on the **MetalPCB** datasets.

Interestingly, the combined AE+TRA approach of initializing the TRA with the encoder network's prediction leads to better losses than the TRA alone in all regions. Additionally, the AE-initialized TRA converged more than 2 times faster due to the stopping criterion being reached earlier.

However, the result also shows that the average losses of all approaches are significantly higher for the **StepChart** datasets than they are for the **MetalPCB** datasets. This can be because the aluminum **StepChart** object (Fig. 2.21) has a more complex physical structure than the **MetalPCB** object (Fig. 2.20), which can result in a mixture of scattered THz pulses involving multi-path interference effects in all object regions. Incorporating such effects in the reflection model of (4.1) could therefore be an interesting aspect of future research for improving the explainability of the measured data with the physical model.

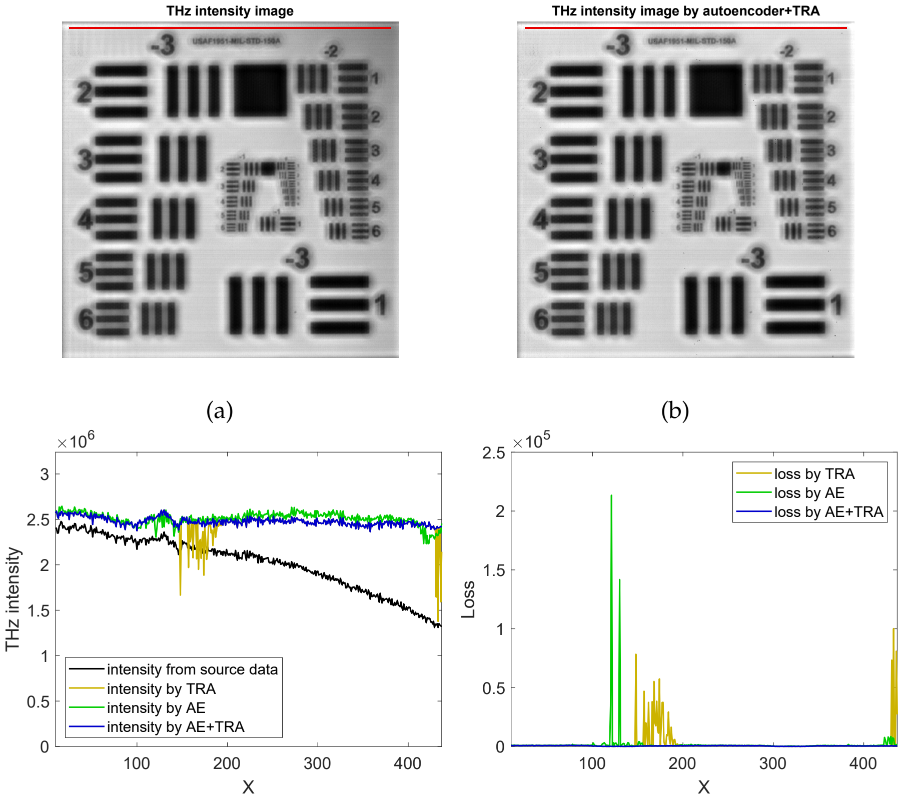
4.4.2 *Quality Assessment*

As stated in Chapter 3 Sec. 3.2.7, in THz imaging the *intensity* image I, which is equal to the squared amplitude i.e. $I = \hat{e}^2$, is one of the most important physical quantity for quality assessment. Note that the intensity could be inferred directly from the data by considering that (4.1) yields

$$A_u(\mu) \cdot A_u^*(\mu) = \hat{e}^2 \cdot \operatorname{sinc}^2(0) = \hat{e}^2 = I \tag{4.4}$$

where A^* is the complex conjugate of A, and the intensity level should be calibrated to around 1 at the metal region.

As illustrated in Fig. 4.7, the model-based AE approach is not only capable of extracting all relevant parameters (i.e. $\hat{e}$, μ, σ and ϕ) but compared to values directly extracted from the source data, the resulting intensity I is more consistent in homogeneous material



(a) (b)

(c) (d)

Figure 4.7: Comparison of the THz intensity for the **MetalPCB** datasets: (a) intensity image extracted from the source data without any model-based processing (in red: the pixel line for plots (c) and (d)); (b) image extracted by the AE+TRA approach (in red: the pixel line for plots (c) and (d)); (c) plot of the intensity extracted along the horizontal line in the copper region; (d) plot of the per-pixel loss by TRA, AE, and AE+TRA approaches along the horizontal line in the copper region.

regions. The homogeneity of the directly extracted intensity results from the very low depth of field of THz imaging systems in general, combined with the slight non-planarity of the **MetalPCB** target. As depicted in Figure 4.7c, the intensity variations along the selected line in the homogeneous copper region are reduced using the three model-based methods, i.e. TRA, AE, and AE+TRA.

However, due to the crucial selection of the initial parameters (see discussion at the beginning of Sec. 4.4), the TRA optimization results exhibit significant amplitude fluctuations and loss values (Fig. 4.7d) in the two horizontal sub-regions $x \in [150, 200]$ and $x > 430$. In contrast, the AE and AE+TRA methods deliver superior results with respect to the main quality measure applied in THz imaging, i.e. to the intensity homogeneity and the loss in model fitting. Still, the AE approach shows very few extreme loss values, while the AE+TRA method's loss values are consistently low along the selected line in the homogeneous copper region.

4.5 SUMMARY

In this chapter, a model-based autoencoder approach is introduced for THz image reconstruction. The training of the neural network is formulated as a model-based autoencoder (AE), which allows the training of the corresponding network with real data in an unsupervised way, i.e. without ground truth. The experiment demonstrates that the resulting optimization networks yield model parameters $(\hat{e}, \sigma, \mu, \phi)$ that result in only slightly higher losses than actually running an optimization algorithm, despite the advantage of being more than 140 times faster (if only the inference time is considered). The neural network can serve as an excellent initialization scheme for classical optimizers, by using the prediction of the neural network as a starting point for a gradient-based optimizer. Lower losses are obtained and converge more than 2 times faster than classical optimization approaches while benefiting from all theoretical guarantees of the respective minimization algorithm.

5

DEEP OPTIMIZATION PRIOR

5.1 MOTIVATION

From a modern THz imaging perspective, deriving physically interpretable material quantities from THz data is generally considered to be reliable only acquired by highly specialized THz spectroscopic instrumentation operating in well-controlled experimental laboratory environments. As shown in the foregoing chapters, the estimation of material-related information using 3D THz data attained with widely used THz imaging components is achievable. However, the estimation of the associated physical quantities according to the known FMCW THz data formation model comprises a sophisticated, non-linear optimization process, even if only part of all model parameters (e.g. the electric field intensity) are being estimated (see Chapters 3 and 4). Due to the low signal strength of the widely used THz sources, it takes up to hours to acquire high Signal-to-Noise Ratio (SNR) THz image data for robust parameter estimations, and the parameter estimation for high SNR data already requires significant optimization efforts and fine-tuned parameter initialization. Therefore, there is a high practical interest not only to estimate all model parameters from the acquired 3D THz data but also to improve the robustness of the parameter estimation process for lower SNR THz data.

Often image analysis and reconstruction problems, such as the THz problem stated above, are modeled such that a quantity of interest $u_{x,y}$ is extracted from measurements $g_{x,y}$ at every pixel (x, y) to match a given non-linear data formation process A, i.e. $g_{x,y} = A(u_{x,y})$, yielding optimization problems of the form

$$\min_{u} \sum_{x,y} \mathcal{L}(A(u_{x,y}), g_{x,y}) + R(u), \tag{5.1}$$

where $\mathcal{L}$ and R are a suitable discrepancy measure (e.g. loss) and an optional regularization respectively. Since the regularization term can include a dependency of the neighboring pixels, optimization problems in (5.1) go beyond the pixel-wise optimization problem as mentioned in the foregoing chapters. As A is commonly non-linear and the problem (5.1) is highly non-convex, (5.1) is often solved locally with first-order descent methods.

In this chapter, the concept of *deep optimization prior* is introduced, which is a novel unsupervised method to solve highly non-linear optimization problems. This name comes from the pioneering work deep image prior [UVL 18b], which is applied to problems similar

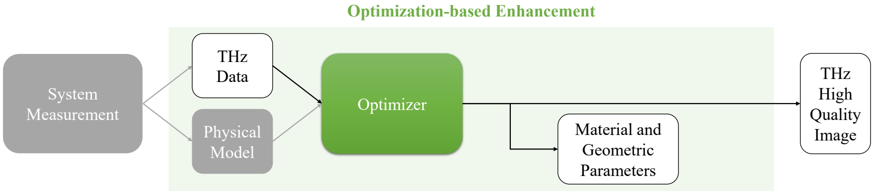


Figure 5.1: The deep optimization prior approach focuses on network-based optimization, the quality of THz images, and the accuracy of model parameters.

to (5.1), but for linear inverse imaging tasks yielding *convex* optimization problems with unique global minimizers, by reparameterizing the original (image) variable u as the output of a CNN network $\mathcal{N}(g;\theta)$:

$$\min_{\theta} \sum_{xy} \mathcal{L}(A(\mathcal{N}(g;\theta)_{xy}), g_{xy}) + R(\mathcal{N}(g;\theta)). \tag{5.2}$$

Subsequently, by omitting the regularizer R and stopping the iteration early [UVL 18b], the reconstruction obtained is of higher quality. The *deep optimization prior* concept extends deep image prior to *non-convex* optimization problems and shows that not only the quality of the solution increases, but also the ability to find *lower energy minima*: by reparameterizing the originally *spatially uncoupled* variables u as the output of a U-net [RFB 15] acting on the data, a gradient descent method can avoid undesirable local minima by taking the neighboring pixels into account. Most strikingly, the quality of a classical approach (5.1) has a severe dependency on a good and robust initialization based on extensive physical knowledge, while the common *random initialization* of network weights seems to be sufficient for consistently finding good local minima for (5.2) [UVL 18b].

As illustrated in Fig. 5.1, this approach focuses on network-based optimization, the quality of THz images, and the accuracy of model parameters within the overall context of the optimization-based enhancement concept (Fig. 1.1b). Sec. 5.2 describes the methodology, Sec. 5.3 gives a theoretical aspect of reparameterizations, and Sec. 5.4 discusses all details regarding the implementation. In Sec. 5.5, the experiment results provide an evaluation of the quantitative and qualitative aspects.

5.2 METHODOLOGY

THz model parameter estimation aims to extract the parameters tensor $u(\vec{x}) = (\hat{e}(\vec{x}), \sigma(\vec{x}), \mu(\vec{x}), \phi(\vec{x})) \in \mathbb{R}^4$ of the THz model (2.16) (see Chapter 2 Sec. 2.2):

$$A(u) = \hat{e} \cdot \operatorname{sinc}(\sigma(z-\mu)) \cdot \exp(-i(\omega z - \phi)) \tag{5.3}$$

where $u = (\hat{e}, \sigma, \mu, \phi)$ denotes the model parameters, and z denotes the device-dependent sampling grid (for simplicity, this variable is omitted in this chapter), at each pixel location $(\vec{x})$ such that it corresponds to the given THz data tensor $G(\vec{x}) \in \mathbb{R}^{N_z \times 2}$, i.e. $\min_u \mathcal{L}(A(u), G(\vec{x}))$. Even with simple choices of the loss function $\mathcal{L}$ such as an ℓ^2-squared loss, the resulting fitting problem is highly nonconvex. Approaches in Chapters 3 and 4 use a simple ℓ^2-squared loss

$$\min_{u} \sum_{x,y} \|A(u_{x,y}) - G_{x,y}\|_2^2, \tag{5.4}$$

to not further exacerbate the goal of a robust THz parameter estimation. Moreover, applying local first-order or quasi-Newton methods to $10^5 - 10^6$ pixel optimizations (5.4) is already quite costly. To the best of the author's knowledge, no regularization approaches have been applied to this kind of THz model parameter estimation so far.

1D PER-PIXEL MODEL-BASED AUTOENCODER As the deep optimization prior approach extends the 1D per-pixel approach of the model-based autoencoder (AE, see Chapter 4), the major difference between the per-pixel model-based AE approach and the deep optimizer prior approach is discussed here in more detail. The model based AE directly estimates $U = (\hat{e}, \sigma, \mu, \phi)$ from the given data tensor $G_{x,y} \in \mathbb{R}^{N_z \times 2}$ in the spatial domain utilizing an one-dimensinoal CNN network $\mathcal{N}^{1D}(\cdot; \theta)$ and the loss function:

$$\min_{\theta} \sum_{\text{training batch } b} \|A(\mathcal{N}^{1D}(G^b_{x,y}; \theta)) - G^b_{x,y}\|_2^2. \tag{5.5}$$

This 1D per-pixel model-based AE is trained by the classical *training-then-prediction* approach: using 80% of the pixels for training in (5.5), and tests if the resulting network can directly predict the desired parameters on the remaining 20% of the THz image pixels.

5.2.1 *Deep Optimization Prior*

Recalling the overall approach depicted in Sec. 5.1, the main idea is to reparameterize the unknown (i.e. image) variable $u_{x,y}$ in nonconvex optimization problems of the form (5.1) by the prediction of a neural network $\mathcal{N}$ via $u_{x,y} = \mathcal{N}(G_{x,y}; \theta)$ for network parameters, yielding a reformulated optimization problem of the form (5.2).

Fig. 5.2 illustrates the overall *deep optimization prior* network architecture comprising the combination of the reparameterization network and the model-based autoencoder. The deep optimization prior approach is minimizing the loss function $\mathcal{L}$ as an optimizer during the unsupervised *training* procedure, which is different from the unsupervised *training-then-prediction* approach mentioned in Chapter 4. The

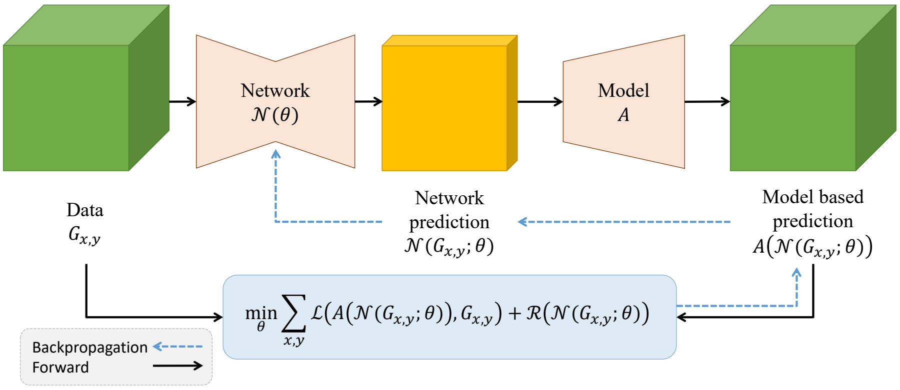


Figure 5.2: The deep optimization prior approach: reparametrizating $u_{x,y}$ by a network $\mathcal{N}$ in combination with the model-based autoencoder.

network-based reparameterization for THz model parameter estimation by combining (5.2) and (5.4) is formulated as:

$$\min_\theta \sum_{x,y} \|A(\mathcal{N}(G_{x,y};\theta)_{x,y}) - G_{x,y}\|_2^2. \tag{5.6}$$

Besides the data term loss function (5.6), a regularization term for THz model parameter estimation can be applied. As the result will be shown in Sec. 5.5, the regularization improves the THz parameter estimation in the case of individual pixel failure, i.e. shot noise. The regularizing term is added to (5.6) as follows:

$$\min_\theta \sum_{x,y} \|A(\mathcal{N}(G_{x,y};\theta)_{x,y}) - G_{x,y}\|_2^2 + \lambda\|\nabla\mathcal{N}(G;\theta)_{x,y}\|_1, \tag{5.7}$$

where ∇ is the gradient operator on neighborhood pixels applied to the model parameter predicted by network $\mathcal{N}$, weighted by the regularization coefficients λ.

5.2.2 *3D Model-based Autoencoder*

As shown in Chapter 4, the 1D per-pixel model-based autoencoder allows unsupervised learning of measurement data by resembling an autoencoder with a learnable network-based encoder and a physical model-based decoder, and therefore able to deal with measurement specific distortions. However, during the learning phase in the per-pixel learning (i.e. z-dimension) approach, the lateral neighborhood information is not considered. Note that in contrast to the 1D per-pixel model-based autoencoder, this network-based reparameterization (Sec. 5.2.1) allows spatial coupling even though the THz model (5.3) is independent in the lateral (x, y) spatial domain.

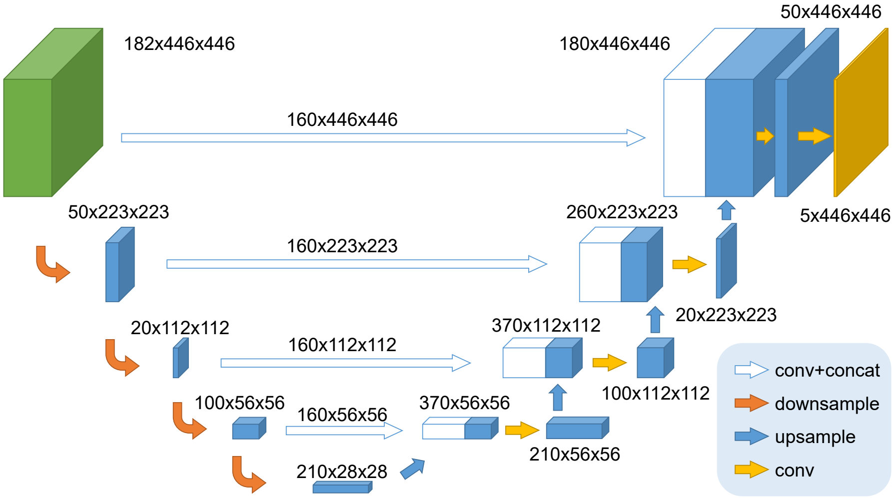


Figure 5.3: The modified U-net architecture of network $\mathcal{N}$ (example for 182 channels with 446×446 pixels) start from the data tensor $G_{x,y}$ (green box) to the desired parameter $u_{x,y} = \mathcal{N}(G_{x,y}; \theta)$ (yellow box). Blue boxes represent feature maps.

NEURAL NETWORK ARCHITECTURE As the THz measurement data is high-dimension data in the form of a 4-D tensor, using typical CNN architectures similar to Chapter 4 is no longer possible because of the high memory and computational requirement of the fully connected layers. Hence, a U-net-type network similar to [RFB 15] is utilized as illustrated in Fig. 5.3. The U-net structure is fully convolutional networks (FCN) [LSD 15], i.e. it only consists of convolutional layers, and is, therefore, computational extremely efficient, while it couples pixels in large lateral spatial regions, which is an important feature in the given application.

The exact structure of the U-net architecture is optimized based on an ablation study on the THz imaging dataset **MetalPCB**. Two modifications to the original U-net architecture [RFB 15] are made that significantly improve the results for optimization purposes: First, the number of channels is not doubled in the encoder part, but an intermediate bottleneck (i.e. the third layer of the encoder, blue box $20 \times 112 \times 112$ in Fig. 5.3) is inserted instead. Second, the number of channels that are skipping from the encoder to the decoder features at the same scale via a concatenation is increased from the commonly used 4 channels to 160 channels (see Fig. 5.3). The effects of these changes are shown in the ablation study Sec. 5.5.1.

In the 1D per-pixel model-based autoencoder, the network directly estimates the phase angle $\phi \in [0, 2\pi]$ linearly, raising the problem of the gradient computation across the $0 - 2\pi$ phase wrap. In the 3D model-based autoencoder, this problem is solved by predicting two real-valued phase components $\phi_c = \cos\phi$ and $\phi_s = \sin\phi$, which restricts to $\phi_c, \phi_s \in [-1, 1]$, and reconstructing the phase afterward

$\phi = \text{atan2}(\phi_s, \phi_c)$. This phase wrapping operation implies that the modified U-net architecture predicts 5 parameters $\{\hat{e}, \sigma, \mu, \phi_c, \phi_s\}$ (see Fig. 5.3) instead of 4 desired parameters. In order to have a fair comparison, the 1D per-pixel model-based autoencoder is adopted by incorporating the phase unwrapping part for the evaluation in Sec. 5.5.

5.3 THEORETICAL ASPECTS OF REPARAMETRIZATIONS

This section provides a theoretical analysis of the parameterization using neural networks and shows that it implicitly corresponds to a variable metric optimization strategy applicable to optimization problems such as (5.1).

Neglecting the regularizer, the optimization problem (5.1) is in itself not coupled on a pixel level. For the sake of simplicity, consider for now the general uncoupled problem

$$\min_{u \in \Omega} \sum_i h_i(u_i), \tag{5.8}$$

where $\Omega = \Omega_1 \times \cdots \times \Omega_n$ is the product space of the pixel-wise domains and h_i are (non-convex) cost functions at pixel i. Clearly, minimizing (5.8) reduces to minimizing problem h_i for each pixel i as the sum of the cost functions decouples on a pixel level. Therefore, gradient descent on problem (5.8) corresponds to gradient descent on each of the sub-problems h_i. Considering a reparameterization of the problem by a continuous function $\mathcal{N} \colon \Theta \to \Omega$ that maps from the function space Θ to the product space Ω, yields

$$\min_{\theta \in \Theta} (H \circ \mathcal{N})(\theta) \tag{5.9}$$

for $H(u) := \sum_i h_i(u_i)$, and thus generalizes (5.6). Although the problems at the pixel level can share a common structure, reformulation (5.9) alone without knowledge of this structure is not advantageous in general due to the preservation of local geometries, as stated in the following remark of *preservation of local minima*: Let $\hat{u}$ be a local minimizer of H in the range of $\mathcal{N}$, then each $\hat{\theta} \in \mathcal{N}^{-1}(\hat{u})$ is also a local minimizer of $H \circ \mathcal{N}$. This remark means that the reparametrizations are not preventing or moving out from critical points, and the local minima of problem (5.9) are still obtained similar to problem (5.8).

Furthermore, assuming differentiability of $\mathcal{N}$, consider a continuous interpretation of gradient descent, the *gradient flow* with respect to H, i.e. a $\theta(t)$ such that $\theta'(t) = -\nabla(H \circ \mathcal{N})(\theta(t))$. Then, for $u(t) := \mathcal{N}(\theta(t))$, it holds

$$\begin{aligned} u'(t) &= \nabla\mathcal{N}(\theta(t))^T\theta'(t) \\ &= -\nabla\mathcal{N}(\theta(t))^T\nabla(H \circ \mathcal{N})(\theta(t)) \\ &= -\nabla\mathcal{N}(\theta(t))^T\nabla\mathcal{N}(\theta(t))\nabla H(u(t)) \end{aligned} \tag{5.10}$$

For $\Omega \subset \mathbb{R}^d$ the matrix $M(t) := \nabla\mathcal{N}(\theta(t))^T\nabla\mathcal{N}(\theta(t)) \in \mathbb{R}^{d\times d}$ is positive semi-definite and hence $-M(t)\nabla H(u(t))$ is a descent direction. Hence, it is hypothesized that for certain problem classes h_i as studied in the numerical experiments, the temporally changing implicit gradient preconditioning with $M(t)$ is advantageous in terms of training dynamics. In particular, networks with a large receptive field such as a U-net typically yield dense matrices $\nabla\mathcal{N}(\theta(t))^T\nabla\mathcal{N}(\theta(t))$ and thus induce changes in predictions $u_{x,y}$ even if $\frac{\partial H}{\partial u_{xy}} = 0$.

Therefore, this theoretical analysis shows that the reparametrizations are not eliminating critical points of the original cost function, but they only yield a different gradient descent path even if part of the pixels' gradients with respect to H is zero.

5.4 IMPLEMENTATION

In this chapter, all computation times are recorded by NVIDIA® GTX 1080Ti GPU, using PyTorch 1.9.0 version. The source code is available at https://github.com/tak-wong/Deep-Optimization-Prior.

The optimization performance of classical approaches and the deep optimization prior approach is evaluated based on synthetically simulated datasets **SynthUSAF** and **SynthObj** (in addition to AWGN and shot-noise respectively) and measured datasets **MetalPCB**. More details of datasets can be found in Chapter 2 Sec. 2.5.2.

5.4.1 *Training Procedure*

In contrast to the classical *training-then-prediction* approach by 1D per-pixel model-based autoencoder, *deep optimization prior* utilize a training approach which *trains a neural network as an optimizer* for a single dataset. The pseudo-code of the unsupervised deep prior training procedure of the proposed method is shown in Algorithm 5.1, where the inputs are the measurement data tensor G, the physical forward model A, depth z-direction sampling vector z, the number of iterations (i.e. epochs) M. Note that, unlike the random sampled per-pixel approach in Chapter 4, the network $\mathcal{N}$ is trained, i.e. optimized, to predict parameters u at all lateral pixel (x, y) based on the entire 4D tensor G.

Algorithm 5.1 Training procedure of deep optimization prior

```
Input: Data tensor $G$, forward model $A$, z-direction sampling grid $z$, iteration $N_{it}$
function AUTOENCODER($G, A, z, N_{it}$)
    $\mathcal{N}$, $\theta$ = NN();                      ▷ initialize a neural network model
    for $i = 1$ to $N_{it}$ do
        $u = \mathcal{N}(G;\theta)$;                     ▷ network prediction
        Model = $A(u|z)$;                                ▷ physical model
        Loss = $\mathcal{L}(\text{Model}, G)$;           ▷ loss function
        $\theta = \mathcal{N}$.train(Loss);              ▷ backpropagation
    end for
return $\theta$, $u$
end function
```

5.4.2 *Choice of Optimizer*

Commonly used optimization methods for the THz inverse problem can be categorized as: Hessian based methods (second order gradient), which include Levenberg Marquardt [Marq 63], Trust Region Algorithm [CL 96], and LBFGS [LN 89]; Gradient descent methods (first order gradient), which include gradient descent, and steepest gradient descent.

The deep optimization prior loss functions (5.6) and (5.7) is optimized by the AdamW [LH 18] optimizer as implemented in PyTorch with GPU acceleration. To ensure a fair comparison, the classical optimization (5.1) is phrased as the minimization of a "network" that does not receive any input vector (i.e. without any input layer), but instead only outputs the learnable parameters u to avoid any differences in implementation. As a second baseline, the LBFGS [LN 89] optimizer is additionally evaluated as one of the classical approaches to exclude a systematic advantage of the specific AdamW method for optimization problems with a deeply nested structure. All formulations and optimizers are run for 1200 iterations (i.e. *full-batch epochs* in machine learning terminology). Moreover, these optimizers are compared to the 1D per-pixel model-based autoencoder (see Chapter 4). Again, for a fair comparison, the optimization algorithm of the 1D per-pixel model-based autoencoder is changed from Adam to AdamW.

5.4.3 *Projection*

To project parameters onto the non-negative orthant, the network predicted parameters $\mathcal{N}(G;\theta)$ are projected to $[u_{min}, u_{max}]$ using the sigmoid function. Similarly, for LBFGS and AdamW optimizers, parameters u are projected to the non-negative orthant, except that the linear bounded function is adopted instead of the sigmoid function:

$$u = \min\left(\max(u_{min}, u_o), u_{max}\right) \tag{5.11}$$

where u_o denotes the ordinary parameters from classical optimizers. Based on the empirical comparison, the linear bounded projection function (5.11) performs better than the sigmoid function in terms of the minimized loss value. However, because of the non-differentiable zero-point, the projection of LBFGS and AdamW optimizers is implemented after the gradient descent update for each iteration.

5.4.4 *Initialization*

Descent-based non-convex optimization methods depend on the selected parameter initialization. In the numerical experiments, two types of initialization schemes are evaluated for the classical approach of minimizing loss function $\mathcal{L}$ directly: *random* and *physics-based* initialization.

The *random* initialization is to choose every parameter u at every pixel from a uniform random distribution over $[u_{\text{min}}, u_{\text{max}}]$ where u_{min} and u_{max} are estimates of the reasonable minimum and maximum values these parameters should attain. As we will see in Sec. 5.5.2, such initialization is too crude for classical optimization to yield reasonable results.

Second, the *physics-based* initialization tries to exploit physical knowledge about each application in order to provide reasonably accurate initial guesses for the parameters at each pixel. In this chapter, the initialization method discussed in Chapter 3 Sec. 3.2.3 is adopted as *physics-based* initialization, and a *random* initialization is tested for comparison. As *random* initialization yields very bad results for classical optimization methods (see Sec. 5.5.2), the *physics-based* initialization is also applied to AdamW and LBFGS.

Since classical approaches greatly benefit from a good initialization, it is worth a try to benefit from good initial guesses for the network-based reparameterizations: by adding the initial parameters to the network prediction. However, this approach did not improve the numerical results in comparison to the usual random initialization of network parameters which is why this approach was discarded in this chapter.

For the random network initialization, the method from [He+ 15] is adopted for the 1D per-pixel model-based autoencoder and the deep optimization prior. To verify the robustness of random parameters and network initialization, each setting that is related to the random initialization of the model parameters or random initialization of the network parameters is run 5 times. Note that this sanity check is also run for a 1D per-pixel model-based autoencoder to verify its robustness.

5.4.5 *Hyperparameter Optimization*

In order to respect the physical meaning of the THz model parameters, the original data scale is retained for training and optimization. However, the large variance of numeric ranges of these parameters leads to a diverging optimal hyperparameter for network training and optimizer. Therefore, all hyperparameters are optimized by a grid search for 4 learning rates from 10^{-3} to 10^{0} for all approaches individually using the **MetalPCB** dataset, synthetic datasets (**SynthUSAF** and **SynthObj**) at $0dB$ PSNR noise level respectively as the corresponding optimal learning rate (see Table 5.2, column LR).

For the regularization coefficients λ, the coefficients are maximized empirically but do not blur the parameter images based on visual inspection for all optimizers (LBFGS, AdamW, and the deep optimization prior) using the shot noise model.

5.5 EXPERIMENTAL RESULT

In this section, the deep optimization prior approach (abbreviated to DOP) is evaluated in comparison to 1D Per-Pixel Model-based Autoencoder (PPAE, see Chapter 4), LBFGS [LN 89] and AdamW [LH 18]. The evaluation is based on the ablation study on the neural network architecture (Section 5.5.1), the loss (Sec. 5.5.2), the parameter accuracy (Sec. 5.5.3) and the computational requirement (Sec. 5.5.4) respectively, in the numerical and qualitative perspectives.

5.5.1 *Ablation Study on the Network Architecture*

The introduction of concatenating skip connections in U-net that often exceed the number of channels as well as the additional intermediate bottleneck in the encoder appear unintuitive from a supervised learning perspective. Thus, their effect is studied by considering the standard U-net (*Unet*), a U-net with large skip-connections by a standard encoder (*Unet+Skip*), a U-net with standard skip-connection but the modified encoder (*Unet+Bottleneck*), and the modified U-net (*modified Unet*) architecture illustrated in Fig. 5.3. Table 5.1 shows their optimal performance on the measurement datasets **MetalPCB** and on the synthetic datasets **SynthUSAF** and **SynthObj** at various noise levels respectively. All network architectures are trained for 4 learning rates using the real measurement **MetalPCB** datasets respectively. This optimal learning rate is then applied to all datasets.

Noise Level (PSNR)	**Normalized Loss ($\times 10^{-6}$)**				
	Network	**Unet**	**Unet+Skip**	**Unet+Bottleneck**	**Modified Unet (Fig. 5.3)**
MetalPCB					
measured	**Optimal LR**	0.1	0.1	0.01	0.01
measured	**Min.**	58.87	**56.55**	60.16	56.78
	Mean	62.95	57.63	63.81	**57.56**
	Max.	69.33	61.20	67.04	**58.25**
MetalPCB+AWGN					
-20dB	**Min.**	32365.39	30851.79	32064.11	**30797.56**
	Mean	47598.25	35825.00	32121.31	**30871.59**
	Max.	58154.44	54943.34	32196.63	**30918.26**
-10dB	**Min.**	3303.06	**3264.36**	3311.45	3267.41
	Mean	3362.96	8401.39	3369.38	**3271.89**
	Max.	3465.76	28943.87	3431.25	**3278.23**
0dB	**Min.**	422.88	**395.64**	415.69	397.34
	Mean	5552.23	5523.62	423.50	**400.09**
	Max.	26033.73	26032.38	428.74	**403.63**
10dB	**Min.**	117.14	**109.12**	118.03	109.69
	Mean	5241.37	**109.93**	121.15	111.22
	Max.	25729.74	**110.91**	124.82	113.76
SynthUSAF+AWGN					
-20dB	**Min.**	31145.93	29799.38	30948.27	**29746.34**
	Mean	37539.14	44513.64	31022.22	**29802.03**
	Max.	62445.44	66699.85	31080.66	**29819.02**
-10dB	**Min.**	3216.02	**3020.85**	3196.11	3042.54
	Mean	10928.21	**3031.51**	10624.38	3058.49
	Max.	41705.13	**3050.10**	40263.39	3080.95
0dB	**Min.**	343.63	**313.34**	342.51	315.10
	Mean	345.74	517.48	347.60	**317.82**
	Max.	349.03	1324.67	354.28	**320.96**
10dB	**Min.**	43.01	**35.99**	54.54	38.18
	Mean	46.56	**38.94**	60.47	40.82
	Max.	48.75	**41.94**	73.15	45.26
SynthObj+AWGN					
-20dB	**Min.**	32059.66	29860.43	31813.17	**29668.97**
	Mean	38232.22	30048.30	31947.22	**29729.65**
	Max.	60389.10	30240.42	32187.62	**29823.52**
-10dB	**Min.**	3422.17	3095.52	3535.84	**3088.49**
	Mean	9481.55	10106.76	3633.48	**3276.31**
	Max.	33048.26	30638.85	3696.07	**3343.63**
0dB	**Min.**	406.96	323.21	387.25	**323.11**
	Mean	612.38	**381.05**	460.56	387.28
	Max.	885.45	**571.84**	673.18	588.90
10dB	**Min.**	75.92	**40.02**	90.64	48.85
	Mean	98.85	**101.67**	100.37	106.93
	Max.	**112.52**	317.85	121.63	289.59

Table 5.1: Comparison of normalized ℓ^2-squared loss by standard U-net architecture (*Unet*), standard encoder with large skip-connection (*Unet+Skip*), the modified encoder with standard skip-connection (*Unet+Bottleneck*) to the modified U-net architecture illustrated in Fig. 5.3. The best optimizers (lower is better) are highlighted.

Noise Level (PSNR)	Normalized Loss ($\times 10^{-6}$)						
	Optimizer	**PPAE**	**LBFGS**		**AdamW**		**DOP**
	Initialization	**Random**	**Physics**	**Random**	**Physics**	**Random**	**Random**
MetalPCB							
measured	**Optimal LR**	0.001	0.01	0.1	0.001	0.01	0.01
measured	**Min.**	56.89	218.02	15008.87	61.32	12665.03	**56.78**
	Mean	3372.63	218.02	15465.89	61.32	12677.52	**57.56**
	Max.	16615.59	218.02	15904.82	61.32	12688.49	**58.25**
MetalPCB+AWGN							
-20dB	**Min.**	30800.43	39766.81	73667.70	36100.08	49546.92	**30797.56**
	Mean	34927.94	39766.81	105352.48	36100.08	49608.79	**30871.59**
	Max.	51007.51	39766.81	126876.18	36100.08	49643.10	**30918.26**
-10dB	**Min.**	**3217.44**	10488.53	59122.06	7380.64	21559.17	3267.41
	Mean	**3232.18**	10488.53	85814.26	7380.64	21591.09	3271.89
	Max.	**3252.32**	10488.53	107167.15	7380.64	21636.72	3278.23
0dB	**Min.**	**395.70**	1967.64	60312.11	965.00	18182.68	397.34
	Mean	408.61	1967.64	63289.90	965.00	18226.71	**400.09**
	Max.	436.22	1967.64	66334.08	965.00	18247.48	**403.63**
10dB	**Min.**	**108.96**	240.86	22642.91	135.92	17422.87	109.69
	Mean	112.16	240.86	27453.51	135.92	17439.26	**111.22**
	Max.	114.37	240.86	32264.10	135.92	17464.49	**113.76**

Table 5.2: Comparison of ℓ^2-squared loss in (5.6) using **MetalPCB** and **MetalPCB+AWGN** datasets by optimizers PPAE, LBFGS and AdamW to the DOP approach. All optimizers are tested for 4 learning rate individually using *MetalPCB* dataset, and the corresponding optimal learning rates are shown in the first row. Then, this optimal learning rate is applied for different noise level. Note that the ℓ^2-squared loss is normalized by the signal power. The best optimizers (lower is better) are highlighted.

Based on the loss value, the large skip connection U-net (*Unet+Skip*) and the modified U-net architectures achieve the lowest loss for almost all datasets. This improvement by the large skips connection can be because the number of channels of the data tensor (i.e. 182 channels in this chapter) is much larger than the channels of typical 2D image tensors (i.e. 3 or 32 channels used in [UVL 18b]), while there is highly correlated information between the first 91 channels (i.e. real parts) and the second half of channels (i.e. imaginary parts). However, the variance of loss by *Unet+Skip* is significantly higher than the modified U-net for some datasets (e.g. **MetalPCB**, **SynthUSAF** at $0dB$, **SynthObj** at $-10dB$). It indicates that the modified U-net architecture is the most robust architecture among these 4 network structures.

5.5.2 *Loss*

MEASUREMENT DATASETS Table 5.2 shows the average ℓ^2-squared loss in (5.6) for the **MetalPCB** and **MetalPCB+AWGN** datasets for the optimizers PPAE, LBFGS and AdamW to DOP approach.

The DOP approach obtains the lowest loss for the measurement **MetalPCB** dataset. For the additional AWGN noise levels, the DOP approach is overall the best optimizer, except for a marginal gap to PPAE at $-10dB$ PSNR noise level. Moreover, the minimum and maximum of multiple runs are relatively stable for the DOP approach, while PPAE obtains a huge variation in the measurement dataset and at $-20dB$ noise level.

For the initialization, the classical optimizers LBFGS and AdamW obtain very high loss by using random initialization, while the proposed method achieves very good results, i.e., the proposed method is robust to the initialization.

As the physics-based initialization for LBFGS and AdamW is a better optimizer than random-based initialization, we use physics-based initialization for LBFGS and AdamW from here on for the remaining evaluation section.

SYNTHETIC DATASETS WITH AWGN Table 5.3 shows that the deep optimization prior (DOP) approach achieves the lowest average loss for both synthetic datasets and at all noise levels significantly.

DATASETS WITH SHOT NOISE MODEL Table 5.4 shows the average loss with regularization in (5.7) using **MetalPCB**, **SynthUSAF** and **SynthObj** datasets at $0dB$ AWGN and 10% shot noise respectively. To make a fair comparison, all optimizers use the same set of regularization coefficients λ. Note that only LBFGS, AdamW, and the DOP approach are included in this table as PPAE cannot optimize the loss function with total variation regularization.

Apparently, the DOP approach obtains the lowest loss among all optimizers. AdamW also achieves a marginally worse, second-best optimizer for synthetic datasets, while the DOP approach still achieves a significantly lower loss for measurement datasets **MetalPCB**.

DISCUSSION Given the fact that all these optimizers are using the same loss function (5.6) and (5.7) respectively, the result shows that, compared to the prior methods, the deep optimization prior approach finds lower energy minima by avoiding undesirable local minima.

5.5.3 *Parameter Accuracy*

In addition, classical optimizers LBFG and AdamW, the 1D Per-Pixel Model-based Autoencoder (PPAE), and the deep optimization prior (DOP) approach are evaluated according to the parameter accuracy during optimization in (5.6) and (5.7). To estimate the model parameter accuracy, the Root Mean Square Error (RMSE) of the estimated parameters is calculated regarding the ground truth parameters for the synthetic datasets **SynthUSAF** and **SynthObj**. Due to the varia-

Noise Level (PSNR)	**Normalized Loss ($\times 10^{-6}$)**				
	Optimizer	**PPAE**	**LBFGS**	**AdamW**	**DOP**
	Initialization	**Random**	**Physics**	**Physics**	**Random**
SynthUSAF+AWGN					
0dB	**Optimal LR**	0.001	0.01	0.01	0.01
-20dB	**Min.**	**29498.45**	38624.05	35838.50	29746.34
	Mean	38843.38	38624.05	35838.50	**29802.03**
	Max.	65652.00	38624.05	35838.50	**29819.02**
-10dB	**Min.**	**3020.54**	15620.10	9243.03	3042.54
	Mean	19326.69	15620.10	9243.03	**3058.49**
	Max.	35783.14	15620.10	9243.03	**3080.95**
0dB	**Min.**	**312.73**	3730.37	3698.46	315.10
	Mean	8107.89	3730.37	3698.46	**317.82**
	Max.	39273.80	3730.37	3698.46	**320.96**
10dB	**Min.**	45.22	1220.81	1190.69	**38.18**
	Mean	7750.62	1220.81	1190.69	**40.82**
	Max.	38507.01	1220.81	1190.69	**45.26**
SynthObj+AWGN					
0dB	**Optimal LR**	0.001	0.001	0.01	0.01
-20dB	**Min.**	**30073.29**	49725.60	36259.96	29668.97
	Mean	30163.70	49725.60	36259.96	**29729.65**
	Max.	30254.10	49725.60	36259.96	**29823.52**
-10dB	**Min.**	**3091.60**	13370.01	10202.86	3088.49
	Mean	13342.05	13370.01	10202.86	**3276.31**
	Max.	28511.46	13370.01	10202.86	**3343.63**
0dB	**Min.**	**340.81**	7027.83	4032.16	323.11
	Mean	5170.08	7027.83	4032.16	**387.28**
	Max.	19096.94	7027.83	4032.16	**588.90**
10dB	**Min.**	44.90	16733.44	1711.49	**48.85**
	Mean	2741.44	16733.44	1711.49	**106.93**
	Max.	8798.07	16733.44	1711.49	**289.59**

Table 5.3: Comparison of ℓ^2-squared loss in (5.6) using **SynthUSAF+AWGN** and **SynthObj+AWGN** datasets by optimizers PPAE, LBFGS and AdamW to the DOP approach. The optimal learning rate is selected by optimizing the loss by $0dB$ noise level.

tion in signal power of different noise levels, the RMSE of the model parameter $\hat{e}$ is normalized by the signal power given by the individual dataset.

5.5.3.1 *Synthetic Dataset with AWGN*

NUMERICAL COMPARISON In Fig. 5.4 and Fig. 5.5, the RMSE of model parameters $(\hat{e}, \mu, \sigma, \phi)$ estimated by PPAE, LBFGS, AdamW, and the DOP approach are plotted using **SynthUSAF+AWGN** and **SynthObj+AWGN** datasets respectively.

Noise Level (PSNR)	**Normalized Loss** $\times 10^{-6}$			
	Optimizer	**LBFGS**	**AdamW**	**DOP**
	Initialization	**Physics**	**Physics**	**Random**
MetalPCB+ShotNoise				
0dB	**Optimal LR**	1	0.1	0.01
	Min.	2034.9	1670.4	**941.6**
	Mean	2034.9	1670.4	**967.3**
	Max.	2034.9	1670.4	**980.8**
SynthUSAF+ShotNoise				
0dB	**LR**	1	0.1	0.01
	Min.	10953.1	5036.5	**4819.6**
	Mean	10953.1	5036.5	**4831.2**
	Max.	10953.1	5036.5	**4842.9**
SynthObj+ShotNoise				
0dB	**LR**	1	0.1	0.01
	Min.	7771.8	4329.7	**4253.8**
	Mean	7771.8	4329.7	**4271.7**
	Max	7771.8	4329.7	**4289.1**

Table 5.4: Comparison of loss with regularization in (5.7) using **MetalPCB+ShotNoise**, **SynthUSAF+ShotNoise** and **SynthObj+ShotNoise** datasets by optimizers LBFGS and AdamW to the DOP approach. The learning rate is selected based on optimal learning rate of **MetalPCB+ShotNoise** using $0dB$ PSNR noise level.

In general, the DOP approach has the most accurate (i.e. lowest RMSE) estimation for all model parameters. PPAE occasionally achieves more accurate parameters at low noise levels ($0dB$ and $10dB$), but the error bars show its instability. Given all optimizers are using the same loss function, the parameter accuracy gap between all three per-pixel optimizers (PPAE, LBFGS, AdamW) and the DOP approach shows that the deep optimization prior approach significantly improves the accuracy of model parameter estimation by overcoming the undesired local minima.

QUALITATIVE COMPARISON To check the quality of the estimated model parameters, the following figures depict the corresponding model parameter images $\hat{e}$ (top row), μ, σ and ϕ (bottom row) of ground truth (first column) and estimation by PPAE, AdamW and the DOP approach (last column): Fig. 5.6 and Fig. 5.7 show the model parameter images using the **SynthUSAF** and **SynthObj** datasets respectively, with AWGN noise model at $0dB$ noise level. The images from the **SynthUSAF+AWGN** (Fig. 5.6) datasets visualize the median RMSE (*median quality run*) among 5 runs. The images from **SynthObj+AWGN** (Fig. 5.7) are depicting the maximum RMSE (*worst quality run*) among 5 runs.

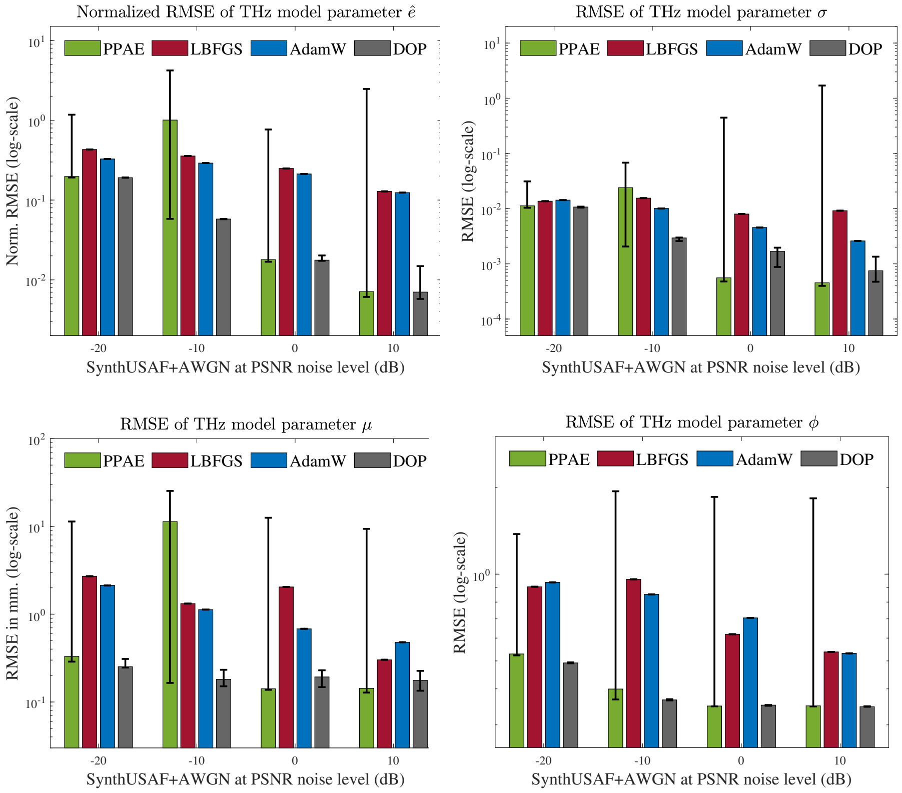


Figure 5.4: Comparison of RMSE of THz model parameters by optimizers PPAE, LBFGS and AdamW to the DOP approach using dataset **SynthUSAF+AWGN** at noise level from -20 to 10dB AWGN. Bars and whiskers indicate minimum, median, and maximum RMSE among 5 runs for each optimizer respectively.

From the μ images for the **SynthUSAF+AWGN** dataset, the PPAE, and the DOP approach both estimate generally accurate parameters, while the DOP approach has slightly more accurate μ values than PPAE in the middle of the structure. However, when comparing the μ images of **SynthObj+AWGN** obtained by the *worst quality run*, the DOP approach performs significantly better than PPAE and AdamW. This performance difference shows that PPAE is sensitive to network initialization, while the DOP approach can overcome undesired local minima given the fact that both autoencoders utilize the same initialization scheme [He+ 15].

5.5.3.2 *Synthetic Datasets with Shot Noise*

NUMERICAL COMPARISON Table 5.5 depicts the RMSE to the ground truth of each model parameter using **SynthUSAF** and **SynthObj** datasets with a shot noise model. In general, the DOP approach obtains the most accurate parameters with a slight margin to AdamW for the **SynthUSAF+ShotNoise** datasets, while AdamW

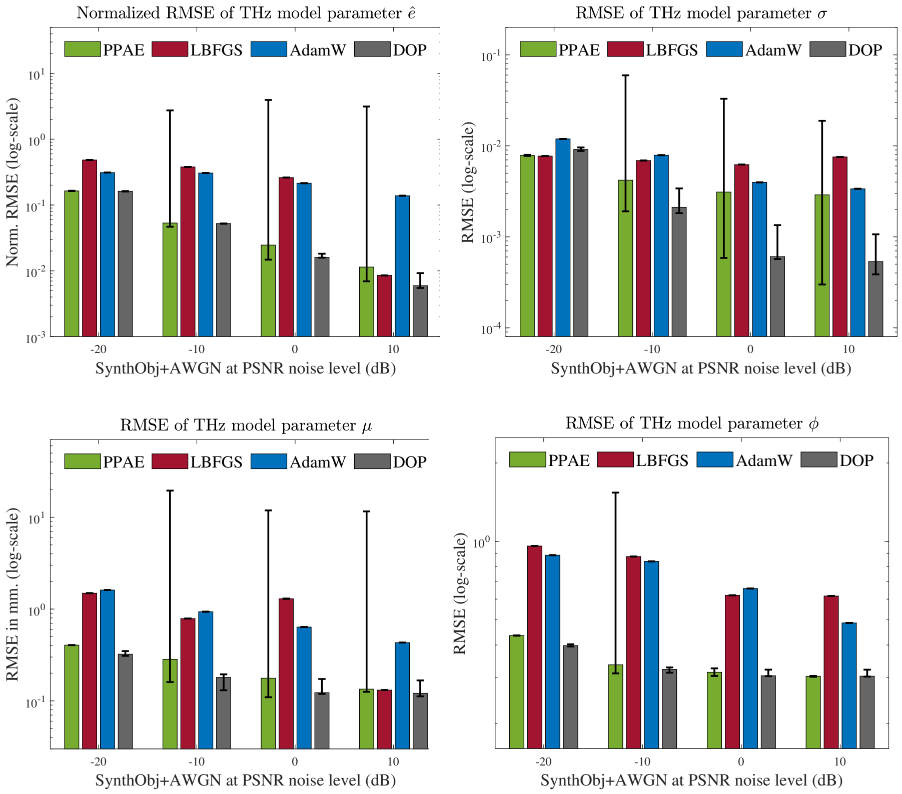


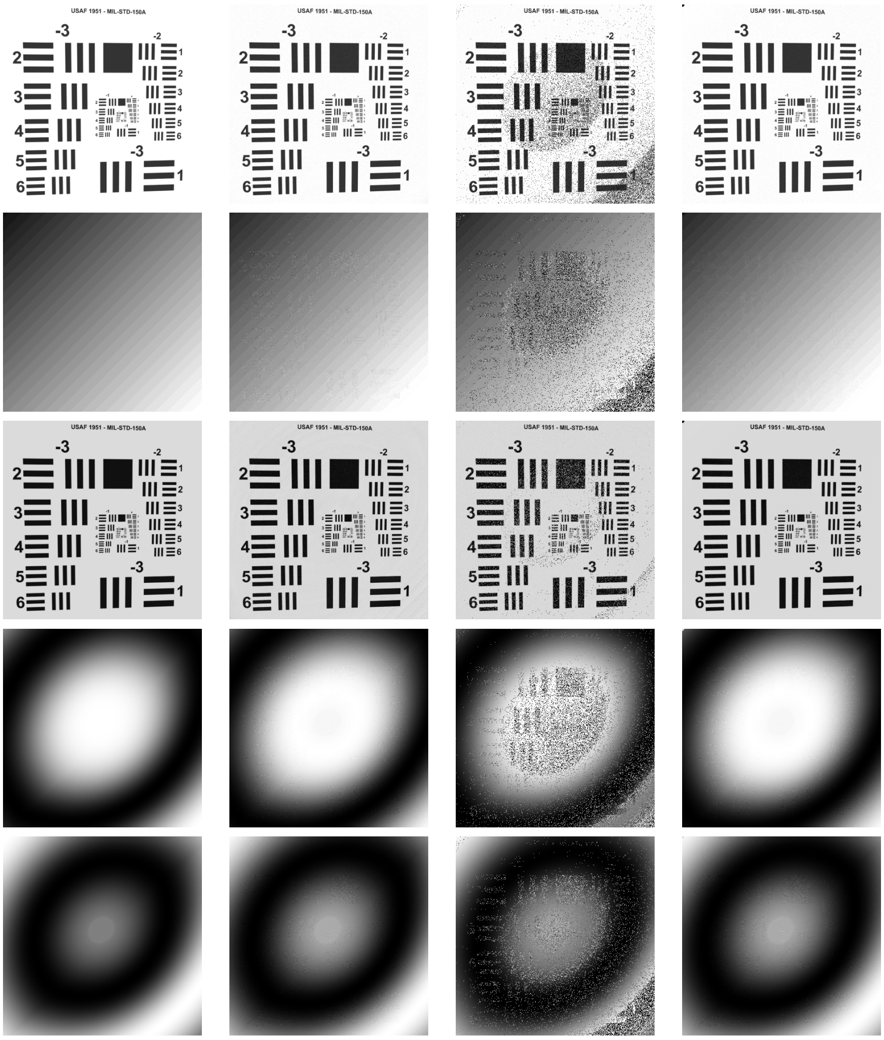
Figure 5.5: Comparison of RMSE of THz model parameters by optimizers PPAE, LBFGS, and AdamW to the DOP approach using datasets **SynthObj+AWGN** at noise level from -20 to $10dB$ PSNR AWGN. Bars and whiskers indicate minimum, median, and maximum RMSE among 5 runs for each optimizer respectively.

achieves better accuracy for the model parameter μ of the **SynthObj+ShotNoise** dataset.

QUALITATIVE COMPARISON The following figures depict the corresponding model parameter images $\hat{e}$ (top row), μ, σ and ϕ (bottom row) of ground truth (first column) and estimation by LBFGS, AdamW, and the DOP approach (last column): Fig. 5.8 and Fig. 5.9 show the model parameter images using **SynthUSAF** and **SynthObj** datasets respectively with shot noise model at 0dB noise level. The images display the median RMSE (*median quality run*) among 5 runs.

By visual comparison of the **SynthUSAF** μ images, it can be seen that the DOP approach removes shot noise significantly, while the AdamW optimizer still retains more shot noise. The visual comparison of the **SynthObj** μ images also shows that the DOP obtains less shot noise than the AdamW optimizer.

DISCUSSION The evaluation of the quality of the parameter images shows that the quality of the optimization solution is significantly



Ground truth PPAE AdamW DOP

Figure 5.6: Comparison of model parameters $\hat{e}$ (first row), μ (second row), σ (third row), $\cos(\phi)$ (fourth row) and , $\sin(\phi)$ (last row) by ground truth, PPAE, AdamW and the DOP approach using the **SynthUSAF+AWGN** dataset at $0dB$ PSNR. All images shows the median RMSE (*median quality run)* among 5 runs.

improved by the deep optimization prior approach However, notice that in rare cases the deep optimization prior approach generates artifacts in the corner of the single parameter image, such as in the top-left corner of the $\hat{e}$ image for the **SynthUSAF+AWGN** datasets (Fig. 5.6).

5.5.4 *Timing and Memory*

Table 5.6 compares the optimization time for PPAE, LBFGS, AdamW, and the DOP approach. As PPAE is trained by the classical *training-then-prediction* approach, the training and prediction times are stated

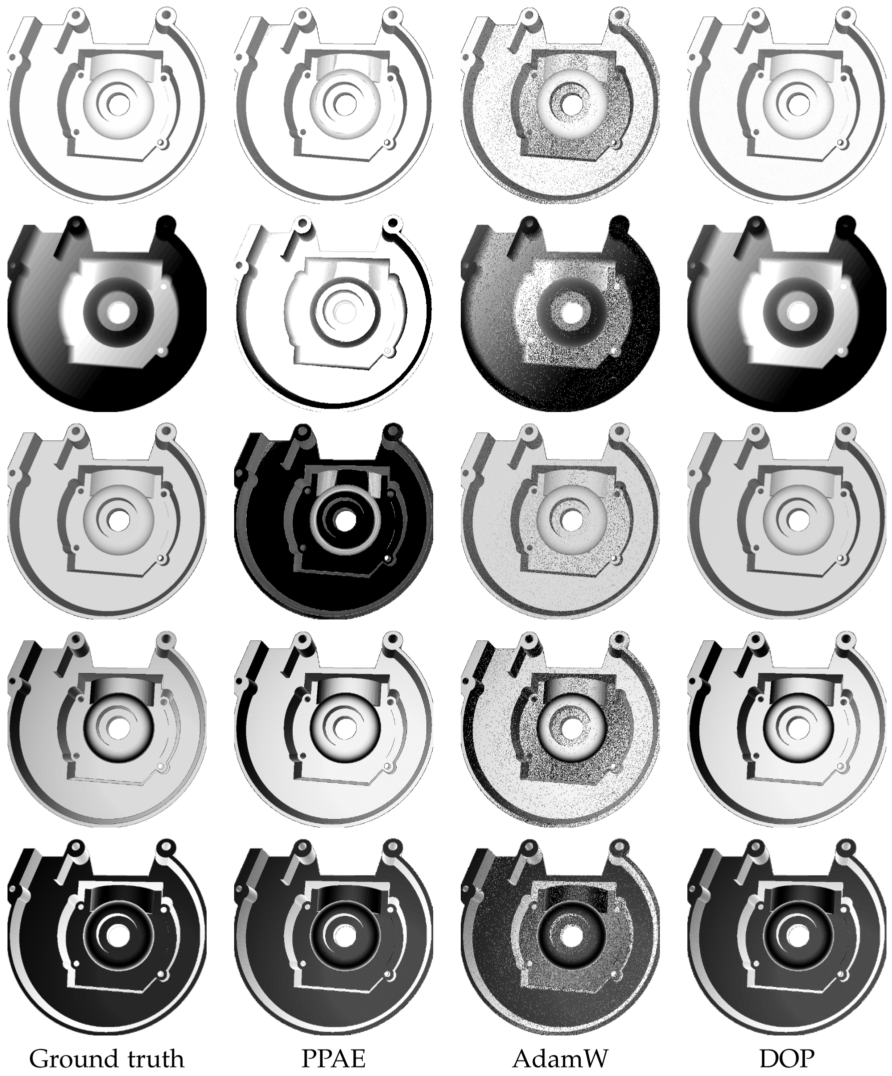


Figure 5.7: Comparison of model parameters $\hat{e}$ (first row), μ (second row), σ (third row), $\cos(\phi)$ (fourth row) and , $\sin(\phi)$ (last row) by ground truth, PPAE, AdamW and the DOP approach using **SynthObj+AWGN** datasets at $0dB$ PSNR. All images show the maximum RMSE among 5 runs (*worst quality run*).

individually. Note that datasets with the shot noise model are optimized by an additional regularization term using (5.7), which includes spatial dependence.

The DOP approach has a drastically improved training time compared to PPAE, from 1.4 hours to 3.7 minutes, i.e. a factor of 22.2. However, when compared to AdamW, AdamW is still the fastest optimizer in general. It is quicker than the DOP approach by a factor of 3.1, when the optimization problem is pixel-wise operation *without* regularization, while it is out-performing the DOP approach only by a factor of 1.6 for the optimization problem with regularization.

To compare the memory requirement for PPAE and the DOP approach, the graphics memory requirement of the neural network is

RMSE (Median of 5 runs)			
Optimizer	**LBFGS**	**AdamW**	**DOP**
SynthUSAF+ShotNoise at $0dB$ PSNR			
LR	1	0.1	0.01
$\hat{e}$	0.2125	0.1346	**0.1274**
μ	0.3606	0.1642	**0.1393**
σ	0.00877	0.00584	**0.00372**
ϕ	0.7067	0.4872	**0.4572**
SynthObj+ShotNoise at $0dB$ PSNR			
LR	1	0.1	0.01
$\hat{e}$	0.2233	0.1098	**0.1079**
μ	0.3083	**0.2344**	0.3129
σ	0.00495	0.00508	**0.00435**
ϕ	0.602	0.4029	**0.387**

Table 5.5: Comparison of model parameters RMSE using **SynthUSAF** and **SynthObj** datasets with shot noise model.

Average time in seconds				
Optimizer	**PPAE**	**LBFGS**	**AdamW**	**DOP**
Optimizer without regularization				
MetalPCB	†5012.0+3.0	1650.3	**71.9**	225.2
MetalPCB+AWGN	†4339.2+3.1	1835.8	**71.6**	223.8
Optimizer with regularization				
MetalPCB+ShotNoise	*N/A	5503.5	**223.2**	367.4
SynthUSAF+ShotNoise	*N/A	5829.8	**225.0**	371.3
SynthObj+ShotNoise	*N/A	4853.8	**225.1**	371.4

† This is average training time and prediction time for PPAE

* Not available: PPAE is not available for regularization.

Table 5.6: Runtime comparison of PPAE, LBFGS, AdamW and the DOP approach.

recorded by PyTorch Profiler during training. For PPAE, the graphics memory consumption for a 446 × 446 batch size is 10.53GB, while the DOP approach only requires 2.20GB. This shows that the DOP approach is more efficient in terms of computation time and memory than PPAE.

5.6 deblurring

In this section, the resolution enhancement is evaluated when combining the deep optimization prior (DOP) approach and the method mentioned in Chapter 3. More precisely, the blind deconvolution

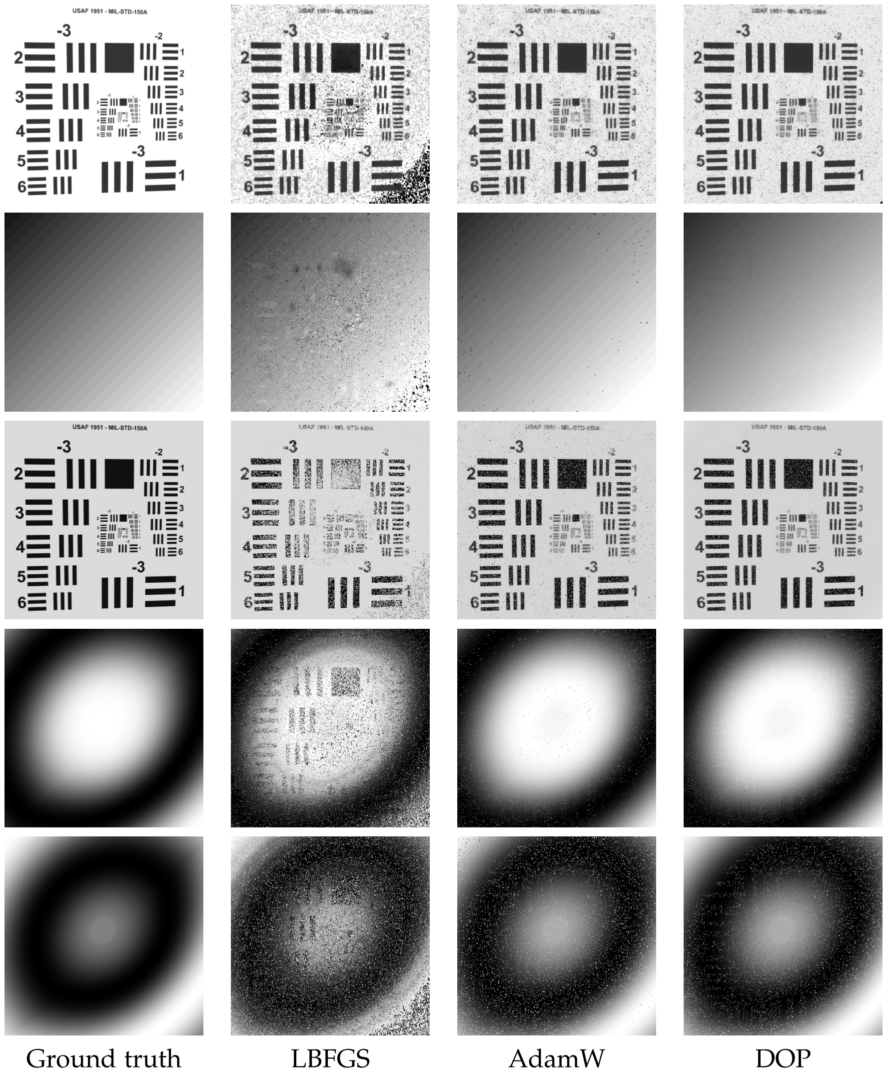


Figure 5.8: Comparison of model parameters $\hat{e}$ (first row), μ (second row), σ (third row), $\cos(\phi)$ (fourth row) and , $\sin(\phi)$ (last row) by ground truth, LBFGS, AdamW and the DOP approach using **Synth-USAF+ShotNoise** datasets at $0dB$ PSNR. All images show the median RMSE (*median quality run)* among 5 runs.

method, such as Xu et al. [XJ 10; XZJ 13], is applied to the resulting intensity images achieved by the DOP approach.

In order to evaluate the impact of different noise levels, the **MetalPCB+AWGN** datasets are synthetically added with an AWGN at PSNR noise level from $-20dB$ to $10dB$ by simulation respectively.

For the comparison of the per-pixel curve fitting optimizer, the AdamW optimizer is chosen instead of the Trust-Region Algorithm (TRA) used in Chapter 3, to compare properly and fairly with the DOP approach in this chapter by taking the same optimizer and hyperparameters.

In this section, firstly the intensity images are reconstructed for each optimizer according to the method mentioned in Chapter 3

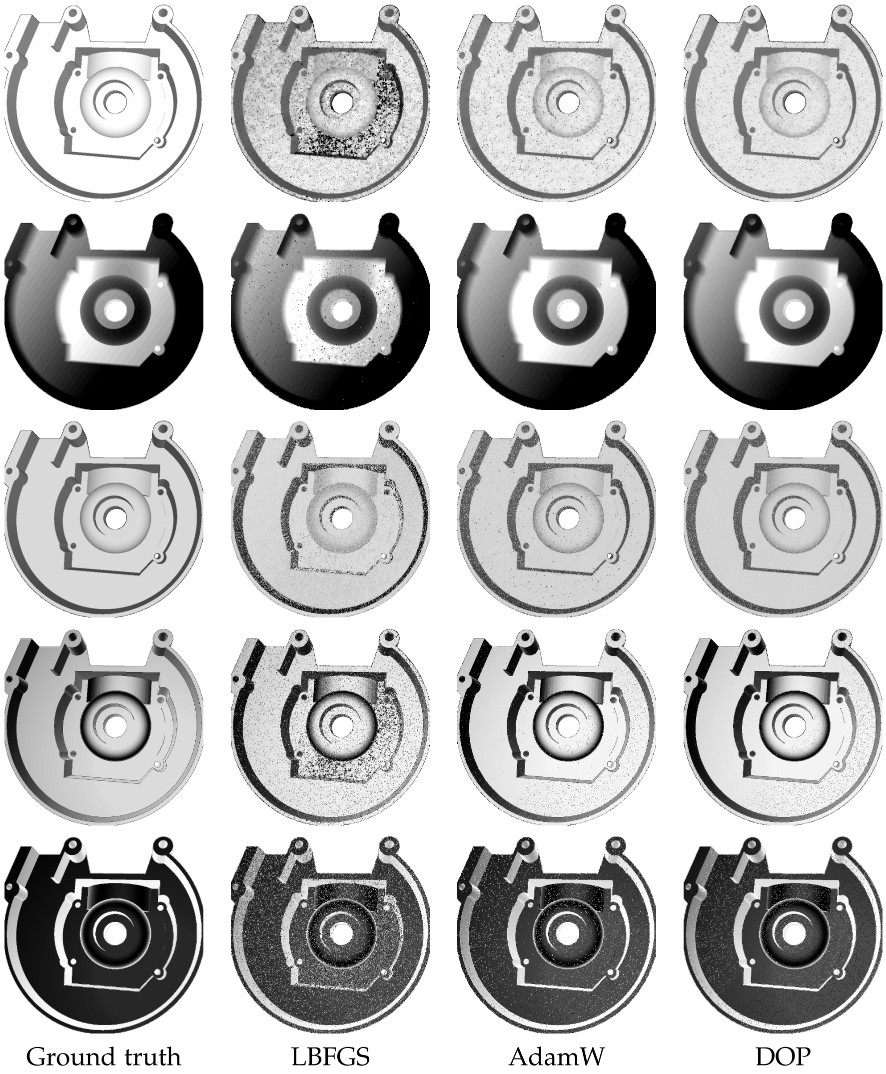


Figure 5.9: Comparison of model parameters $\hat{e}$ (first row), μ (second row), σ (third row), $\cos(\phi)$ (fourth row) and , $\sin(\phi)$ (last row) by ground truth, LBFGS, AdamW and the DOP approach using **SynthObj+ShotNoise** datasets at $0dB$ PSNR. All images shows the median RMSE (*median quality run)* among 5 runs.

Sec. 3.2.7, while the lateral resolution is defined as the finest dimension that can resolve a target with $3dB$ intensity difference. The vertical and horizontal intensity differences are calculated for each resolution group of the resolution target (from $4000\mu m$ to $280.6\mu m$). The vertical and horizontal resolutions are then determined respectively as the first minimum dimension that obtained a $3dB$ crossing of intensity difference. By repeating this procedure, the resolutions for each noise level and the lateral resolution improvement are compared for each optimization approach using different AWGN noise levels (see Table 5.7). Note that in practice the intensity difference can decrease non-homogeneously (see example in Fig. 5.11). Therefore, the range of uncertainty is also determined (see Table 5.7), by indicating the

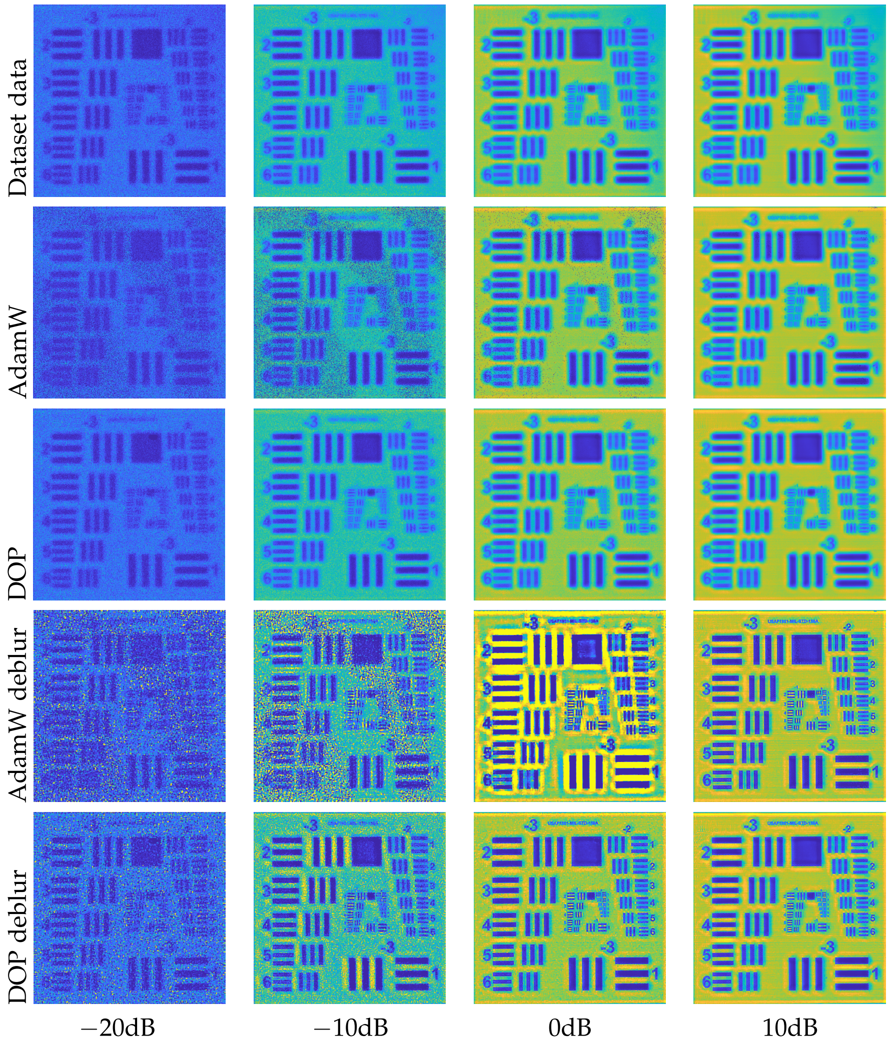


Figure 5.10: Comparison of THz intensity images by original dataset data (first row), the per-pixel AdamW optimizer (second row) and the DOP approach (third row) using **MetalPCB+AWGN** datasets at noise level from $-20dB$ to $10dB$. Images by AdamW and DOP (last two rows) approaches are deblurred by blind deconvolution method from Xu [XJ 10; XZJ 13]. All images shows the lowest RMSE (*best quality run)* among 5 runs.

difference between the first and last noise levels obtaining a $3dB$ crossing intensity difference.

RESULT AND DISCUSSION Fig. 5.10 shows the intensity images by the original datasets (first row), AdamW (second row), the DOP approach (third row), and respective deblurring images (last two rows) using **MetalPCB** datasets at AWGN noise level from $-20dB$ to $10dB$. In this section, all results show the lowest RMSE (*best quality run*) among 5 runs. Note, that the original datasets intensity image is extracted by the same methodology in Chapter 3 Sec. 3.2.7, which is the signal intensity of the **MetalPCB+AWGN** datasets data at the center of the sampling window.

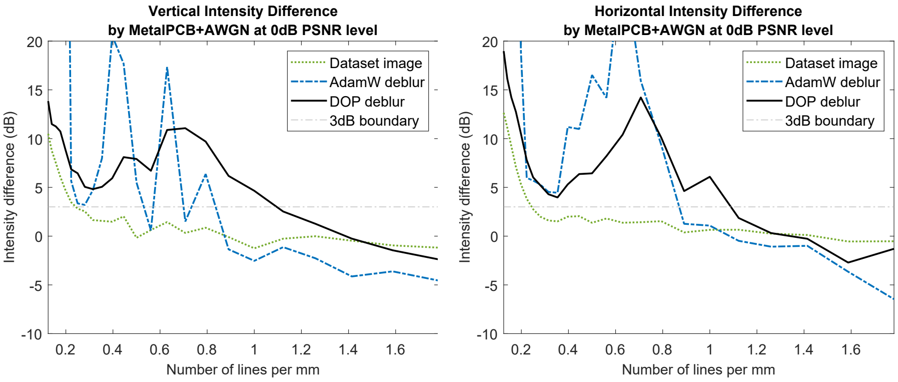


Figure 5.11: Comparison of the vertical and horizontal intensity difference using **MetalPCB+AWGN** datasets at $0dB$ PSNR level for each resolution target group from $4000\mu m$ to $280.6\mu m$. All results show the lowest RMSE (*best quality run)* among 5 runs.

By visual comparison of the intensity images of AdamW and DOP approaches, the DOP approach obtains fewer outliers for very low PSNR levels, i.e. $-20dB$ and $-10dB$. These outliers are more significantly observable after the deblurring procedure, which makes the deblurred DOP intensity image have a better quality than the deblurred AdamW intensity image because of the robustness of the DOP estimation.

Fig. 5.11 plots the vertical and horizontal intensity difference for each resolution target group from $4000\mu m$ to $280.6\mu m$, using the **MetalPCB+AWGN** datasets at $0dB$ PSNR level as an example. As a higher intensity difference represents a better ability to resolve a resolution target, both deblurred AdamW and DOP intensity images outperform the original dataset image, while the deblurred DOP intensity image obtains a more stable intensity difference than the deblurred AdamW image. By comparing the location of the $3dB$ crossing for both deblurred images, the DOP approach enhances the performance of the deblurring method, because of the robustness of model parameter estimation.

Table 5.7 compares the vertical and horizontal resolution for the AWGN PSNR level from $-20dB$ to $10dB$ respectively, and the range of uncertainty is shown in parentheses. Generally, the DOP approach is improving the deblurring method with respect to its resolution enhancement ability. The improvement of the DOP approach over AdamW is mainly due to the improved robustness of model parameters estimation, which obtains a shorter range of uncertainty except for the vertical resolution at $10dB$ and the horizontal resolution at $-20dB$ PSNR level.

Vertical Resolution in μm (range of uncertainty)

SNR	**Datasets image**	**AdamW deblur**	**DOP deblur**
-20dB	2282.62 (-0.0)	2808.23 (-2027.3)	**2223.05 (-1454.1)**
-10dB	2088.29 (-0.0)	2318.77 (-1712.9)	**585.87 (-0.0)**
0dB	2066.73 (-0.0)	944.45 (-344.4)	**457.60 (-0.0)**
10dB	2022.34 (-0.0)	**535.28 (-0.0)**	568.10 (-99.6)

Horizontal Resolution in μm (range of uncertainty)

SNR	**Datasets image**	**AdamW deblur**	**DOP deblur**
-20dB	**2208.92 (-0.0)**	2633.74 (-0.0)	**2288.36 (-1539.3)**
-10dB	2099.83 (-0.0)	1808.98 (-1076.7)	**1048.52 (-447.1)**
0dB	2054.55 (-0.0)	575.93 (-0.0)	**460.14 (-0.0)**
10dB	2055.73 (-0.0)	573.10 (-80.8)	**444.31 (-0.0)**

Table 5.7: Comparison of vertical and horizontal resolution using **MetalPCB+AWGN** datasets at PSNR level from $-20dB$ to $10dB$. The range of uncertainty is shown in parentheses. The best (lower is better) optimizers are highlighted. All images shows the lowest RMSE (*best quality run)* among 5 runs.

5.7 SUMMARY

In this chapter, the deep optimization prior concept with the application to THz model parameters estimation is introduced, i.e. the reparameterization of a pixel-wise non-convex THz model parameter estimation problem via a spatially coupled 3D neural network. Theoretically, surjective reparameterizations can never eliminate critical points of the original cost function but merely yield a different gradient descent path. By numerical and visual evaluation, the experimental results demonstrate that the deep optimization prior approach robustly reconstructs THz model parameters in low SNR and shot noise situations. It finds significantly better local minima compared to classical first-order (AdamW), and second-order (LBFGS) optimizers and the 1D per-pixel model-based autoencoder method, and it is very memory efficient. The modified U-net encoder network architecture results in a neural network that is computationally and memory-wise highly efficient compared to the state-of-art 1D convolutional neural network structure.

Moreover, by combining with the modern blind deconvolution method, the deep optimization prior approach improves the lateral resolution enhancement robustly because of the robust reconstruction of model parameters under low SNR noise conditions. Experiments demonstrate that the deep optimization prior approach improves the lateral resolution enhancement because of the robust reconstruction of model parameters in low SNR noise levels.

6

JOINT MATERIAL AND GEOMETRIC PARAMETER ESTIMATION FOR THZ MULTI-PATH REFLECTION MODEL

6.1 MOTIVATION

According to the single-path reflection model mentioned in previous chapters, the model parameters $(\hat{e}, \sigma, \mu, \phi)$ represent physical quantities in an *implicit* way, i.e. they mathematically denote the signal amplitude, the pulse width, the depth position and the phase angle respectively. However, each parameter should obtain certain *explicit* information regarding its physical property, e.g., it should be possible to denote the depth position by the time-of-travel in seconds or the material refractive index. Hence, there is a high interest for researchers in THz imaging to obtain parameters out of THz models in a more *physical* perspective, for both the single-path and the multi-path reflection.

By representing parameters in a more physical-based perspective, solving the inverse problem of parameter estimation also implies that joint estimation of material parameters (e.g. the refractive index, the dielectric coefficient) and geometric parameters (e.g. the depth position, the material thickness) is possible. While FMCW THz imaging systems commonly estimate the geometric parameters (e.g. thickness [Schr+ 19]), THz researchers often utilize time-domain spectroscopy systems for material parameters (e.g. the refractive index, the dielectric coefficient) individually [BFD 05; Requ+ 06; BZR 18] and both material and geometric parameters simultaneoulsy [DBM 01]. As the practical aspect of THz imaging, it is interesting to study the possibility of this joint material and geometric parameters estimation for FMCW THz imaging systems.

Moreover, regarding the direct, single bounce reflection model used in previous chapters, the discrete received signal $a[n]$ in the frequency domain is transformed (i.e. deramped) to the discrete-time domain signal $A[k]$ by the discrete-time Fourier transform. This transformed time signal is then interpreted as the spatial domain signal $A[z]$, because each time domain sample represents a certain time-of-travel in the air and hence the spatial depth position (see Chapter 2 Sec. 2.2).

However, this interpretation can be misleading and incomplete when the THz radiation is no longer traveling in one medium (i.e. air). Indeed, given that the FMCW THz system transmits a linear frequency modulated waveform, i.e. $f_{\mathrm{T}}(t) = f_c + K_r t$ in Chapter 2 (2.2), the transmitted and received THz signal can be interpreted in both

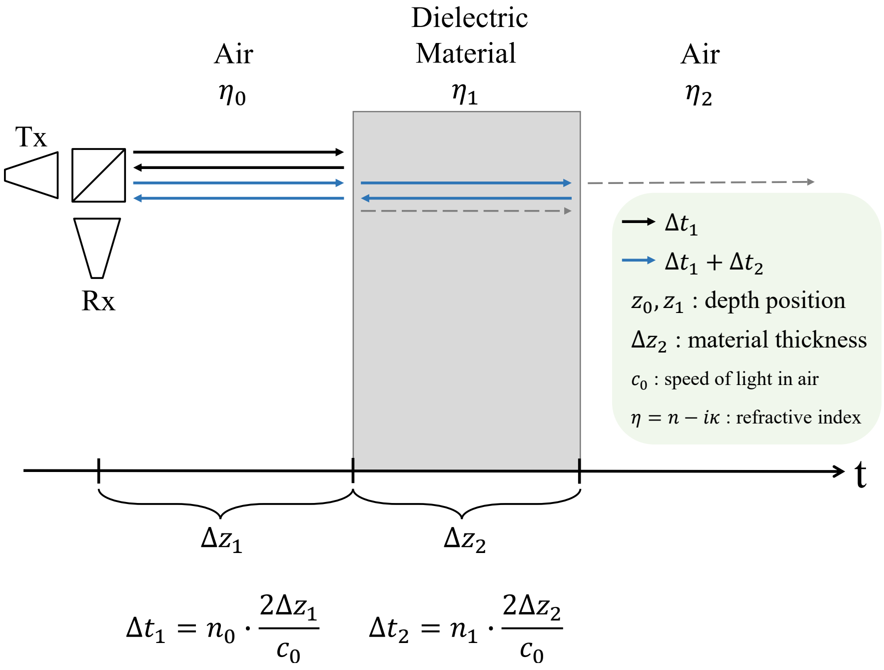


Figure 6.1: The multi-path reflection model is simplified to two reflection paths.

the time or the frequency domain signal correctly, in which the latter one can be deramped to time domain signal.

Hence, in order to have a clearer picture of the THz signal in the multi-path reflection model, in this chapter the received signal is interpreted as the *frequency domain signal*, and this frequency signal is transformed (i.e. deramped) to the *time domain* by the discrete time *Inverse Fourier Transform*.

According to this interpretation, as illustrated in Fig. 6.1, the time domain signal contains multiple responses for each transition position of a dielectric material, and hence the time difference between adjacently received impulses (i.e. Δt_2) represents the thickness of the material. In theory, this reflection model can be extended to more than two paths, but it would drastically increase the complexity of computation. Hence, in this chapter, only the first two paths are assumed to be detectable and the full paths are neglected (depicted as dotted arrows in Fig. 6.1) for simplicity.

Therefore, the joint material and geometric parameter estimation of the dielectric material is formulated as the estimation of material parameters (i.e. the refractive index n_1 and the dielectric coefficient κ_1, where the complex refractive index $\eta_1 = n_1 - i\kappa_1$) and geometric parameters (i.e. the depth position of material Δt_1 and the material thickness Δt_2), from the measured multi-path THz time domain signal.

In this chapter, the joint material and geometric parameter estimation for the THz multi-path reflection model is described. This approach expresses the inverse problem of parameter estimation as

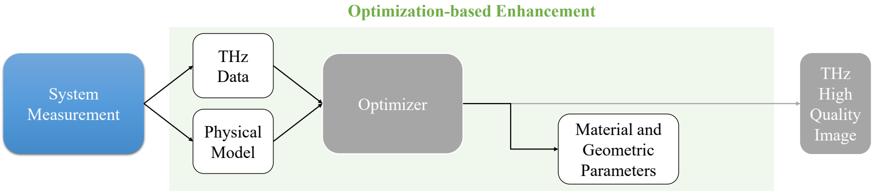


Figure 6.2: The joint material and geometric parameter estimation approach provides a first insight into the THz physical model from the system measurement, the quality of the THz data, and the estimated material and geometric parameters.

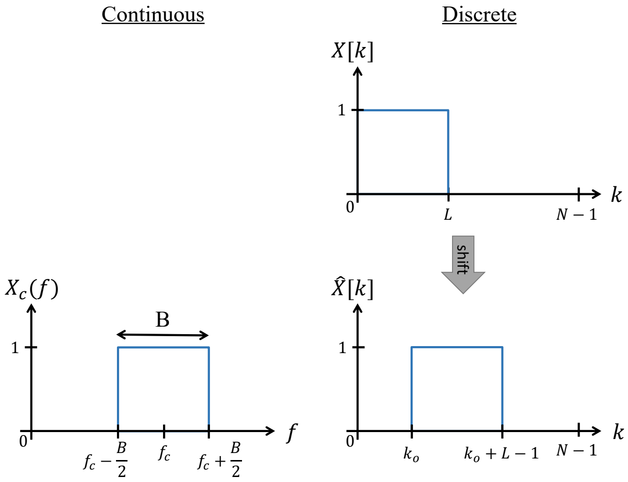


Figure 6.3: The start-from-zero discrete frequency signal is not correctly modeling the THz system, which is operating at the center frequency f_c with the bandwidth B. Instead, a discrete frequency shift k_o is introduced to the measured THz data before the deramping inverse Fourier Transform.

an optimization problem, which is based on the THz multi-path reflection model, the objective loss function, and the optimizer. By solving this optimization problem, both material-based and structural-related physical quantities are jointly estimated from the measured THz data, and hence enable per-pixel classical optimizers or modern neural network-based optimizers mentioned in previous chapters to be utilized.

As illustrated in Fig. 6.2, this approach provides a first insight into the THz physical model, the quality of the THz data, and the material and geometric parameters within the overall context of the optimization-based enhancement concept (Fig. 1.1b). Sec. 6.2 describes the approach in terms of the physical model and the mathematical optimization problem in detail. Sec. 6.3 evaluates the approach by the verification of the forward THz multi-path reflection model, the theoretical analysis of loss functions, and the material and geometric parameter accuracy.

6.2 METHODOLOGY

In this section, the THz multi-path reflection model is extensively studied and formulated by physically interpretable constants and variables. This model is derived by the Institute of High Frequency and Quantum Electronics (HQE). First, it is important to revisit the continuous frequency signal (see Chapter 2 Sec. 2.1.2) operated by the FMCW THz system $X_c(f)$ (as illustrated in Fig. 6.3, bottom-left), which is operating at the center frequency f_c with the bandwidth B, and the corresponding continuous-time signal $x_c(t)$ after the Inverse Fourier Transform yields:

$$\begin{aligned} X_c(f) &= \mathrm{rect}\left(\frac{f-f_c}{B}\right) \\ x_c(t) &= \mathcal{F}^{\text{-}1}\left\{X_c(f)\right\} = \int_{-\infty}^{\infty} X_c(f) e^{i2\pi f t} df \\ &= \frac{1}{B}\int_{f_c-B/2}^{f_c+B/2} e^{i2\pi f t} df = \frac{1}{B}\left[\frac{e^{i2\pi f t}}{i2\pi t}\right]_{f_c-B/2}^{f_c+B/2} \\ &= \frac{1}{B}\cdot\frac{e^{i2\pi f_c t}}{\pi t}\left(\frac{e^{i\pi B t}-e^{-i\pi B t}}{2i}\right) \\ &= \frac{1}{B}\cdot e^{i2\pi f_c t}\cdot B\cdot\frac{\sin(\pi B t)}{\pi B t} \\ &= \mathrm{sinc}(Bt)\cdot e^{i2\pi f_c t} \quad \text{, where } \mathrm{sinc}(t) = \frac{\sin(\pi t)}{\pi t} \end{aligned} \tag{6.1}$$

According to the basic principle of signal processing, as the measured THz data (i.e. the discrete frequency signal) represents the continuous frequency signal, the discrete signal transformation corresponding to the DTFT in Chapter 2 (2.12) is also reviewed in this section. Hence, similar to the DTFT, the Discrete Time Inverse Fourier Transform (DTIFT) is applied to a square discrete frequency signal. Note that DTFT and DTIFT only differ in the exponential term $e^{-i2\pi f_c t}$ and $e^{i2\pi f_c t}$, which should not cause a fundamental difference except the sign of the exponential part.

DISCRETE TIME INVERSE FOURIER TRANSFORM (DTIFT) Given an ideal square discrete frequency domain signal $X[k]$ starting from zero with the signal length L (Fig. 6.3, top-right):

$$X[k] = \begin{cases} 1, & 0 \le k \le L-1 \\ 0, & \text{otherwise} \end{cases}$$
$$\text{for } 0 \le k \le N-1$$

where $N = L \cdot N_p$ denotes the length of the signal after zero-padding by a factor of N_p. After DTIFT, the discrete-time domain signal is:

$$x[n] = \mathcal{F}^{\text{-}1}\left\{X[k]\right\} = \frac{1}{N} \cdot \frac{\sin[2\pi nL/(2N)]}{\sin[2\pi n/(2N)]} \cdot \exp\left(i2\pi\frac{n}{N}\left(\frac{L-1}{2}\right)\right) \tag{6.2}$$

However, given that the continuous signal model (Fig. 6.3, bottom-left) is centered in carrier frequency f_c with a bandwidth B, the start-from-zero discrete signal $X[k]$ (Fig. 6.3, top-right) does not match the continuous signal model, which makes this discrete signal $X[k]$ inappropriate to show how the THz system actually captures the signal.

In fact, in this chapter, the discrete signal from the THz system is considered as a *right shifted* discrete signal $\hat{X}[k]$, where the discrete signal is an ideal square signal starting at the discrete starting frequency k_o (Fig. 6.3, bottom-right), and it yields:

$$\hat{X}[k] = X[k - k_o]$$

Hence, the discrete time signal $\hat{x}[n]$, which properly models the system, is expressed as the deramped signal of $\hat{X}[k]$ by Inverse Fast Fourier Transform (IFFT):

$$\begin{aligned}\hat{x}[n] &= \mathcal{F}^{\text{-}1}\left\{\hat{X}[k]\right\} = \mathcal{F}^{\text{-}1}\left\{X[k-k_o]\right\} = \mathcal{F}^{\text{-}1}\left\{X[k]\right\} \cdot \exp\left(i2\pi\frac{n}{N}\cdot k_o\right)\\ &= x[n] \cdot \exp\left(i2\pi\frac{n}{N}\cdot k_o\right)\\ &= \frac{1}{N} \cdot \frac{\sin[2\pi nL/(2N)]}{\sin[2\pi n/(2N)]} \cdot \exp\left(i2\pi\frac{n}{N}\left(\frac{L-1}{2} + k_o\right)\right)\end{aligned} \tag{6.3}$$

Therefore, (6.3) indicates that unlike the data preprocessing in the previous chapters, *the modified deramping* is taken instead for the measured THz signal: given the measured THz discrete frequency signal $G_f[k]$, the measured discrete time signal $g_t[n]$ is properly modeled by an *IFFT* and *a frequency shift term*:

$$g_t[n] = \mathcal{F}^{\text{-}1}\left\{G_f[k-k_o]\right\} = \mathcal{F}^{\text{-}1}\left\{G_f[k]\right\} \cdot \exp\left(i2\pi\frac{n}{N}\cdot k_o\right) \tag{6.4}$$

with a discrete frequency shift $k_o = \frac{f_{\min}}{B/L}$. In this chapter, g_t denotes the THz discrete time signal after transformation from the measured THz discrete frequency data. As $\operatorname{sinc}(t) \approx \frac{\sin(\pi t)}{L\sin(\pi t/L)}$ and $\frac{L-1}{L} \approx 1$ when L is large, and taking the time parameter $t = \dfrac{n}{N_p} \cdot \dfrac{1}{B}$, (6.3) yields:

$$\begin{aligned}
\hat{x}[n] &= \frac{L}{N} \cdot \operatorname{sinc}(\frac{n}{N_p}) \cdot \exp\left(i2\pi \frac{n}{N_p} \cdot \frac{1}{L}\left(\frac{L-1}{2} + k_o\right)\right) \\
\hat{x}[t] &= \frac{L}{N} \cdot \operatorname{sinc}(Bt) \cdot \exp\left(i2\pi t \cdot \frac{B}{L}\left(\frac{L-1}{2} + k_o\right)\right) \\
&= \frac{L}{N} \cdot \operatorname{sinc}(Bt) \cdot \exp\left(i2\pi t\left(\frac{B}{2} + f_{\min}\right)\right) \\
&= \frac{L}{N} \cdot \operatorname{sinc}(Bt) \cdot \exp\left(i2\pi f_c t\right)
\end{aligned} \tag{6.5}$$

Hence, by considering the time-shift property and the linearity property of DTIFT as shown in Chapter 2 (2.14), in this chapter the THz single-path reflection model A_{sp} is expressed as:

$$A_{\text{sp}}(t) = A_0 \cdot \operatorname{sinc}\left(B(t - \Delta t_1)\right) \cdot \exp\left(i\omega_c(t - \Delta t_1)\right) \cdot r_{01} \tag{6.6}$$

where $A_0 = \frac{L}{N}$ denotes the signal amplitude at the center position, Δt_1 denotes the time-of-travel of the first reflection path, $\omega_c = 2\pi f_c$ denotes the angular center frequency and r_{01} denotes the reflection coefficient from first material to second material (see Fig. 6.1, black arrow). By assuming a perpendicular angle of incidence and using the Fresnel equations, the reflection coefficient r_{ab} from material a to material b is determined by their complex refractive indices $\eta = n - i\kappa$:

$$r_{ab} = \frac{\eta_b - \eta_a}{\eta_b + \eta_a} \tag{6.7}$$

6.2.1 *THz Multi-path Reflection Model*

As illustrated in Fig. 6.1, the multi-path reflection model is expressed as the super-position of first-path reflection (i.e. single-path reflection model from air to the material) and the second-path reflection which scatters the THz beams in the dielectric material. By extending the single-path reflection model in (6.6), the multi-path reflection model is expressed as:

$$\begin{aligned}
A_{\text{mp}}(t) = &A_0 \cdot \operatorname{sinc}\left(B(t - \Delta t_1)\right) \cdot e^{i\omega_c(t-\Delta t_1)} \cdot r_{01} + \\
&A_0 \cdot \operatorname{sinc}\left(B(t - (\Delta t_1 + \Delta t_2))\right) \cdot e^{i\omega_c(t-(\Delta t_1+\Delta t_2))} \\
&\cdot t_{01} \cdot r_{12} \cdot t_{10} \cdot e^{-\alpha_1 \Delta z_2}
\end{aligned} \tag{6.8}$$

where Δt_2 denotes the time-of-travel in the dielectric material, and the transmission coefficient t_{ab} from material a to material b, the propagation loss α_1, the depth position Δz_1 and the material thickness Δz_2 are expressed as:

$$\begin{aligned} t_{ab} &= \frac{2\eta_b}{\eta_b + \eta_a} \\ \alpha_1 &= \frac{2 \cdot \omega_c}{c_0} \cdot \kappa_1 \\ \Delta z_1 &= \frac{c_0 \cdot \Delta t_1}{2} \\ \Delta z_2 &= \frac{c_0 \cdot \Delta t_2}{2n_1} \end{aligned}$$

Note that the coefficient term of the second-path $t_{01} \cdot r_{12} \cdot t_{10}$ depicts the sequence of reflection-transmission (as illustrated in Fig. 6.1, blue arrow): the transmission from the air to the material, then the reflection from material to the air and finally the transmission from material to air. The energy decay of the THz radiation while traveling within the dielectric material is modeled by the propagation loss α_1 term, given that the dielectric coefficient $\kappa_1 \geq 0$.

After expanding (6.8) in a symbolic way and considering the complex refractive index of air $\eta_0 = \eta_2 = 1 + 0i$, the overall THz multi-path reflection model yields:

$$\begin{aligned} A_{\text{mp}}(t; \Delta t_1, \Delta t_2, n_1, \kappa_1) = {} & A_0 \operatorname{sinc}(B(t - \Delta t_1)) \cdot e^{i\omega_c(t-\Delta t_1)} \cdot \frac{n_1 - 1 - i\kappa_1}{n_1 + 1 - i\kappa_1} \\ & + A_0 \operatorname{sinc}(B(t - (\Delta t_1 + \Delta t_2))) \cdot e^{i\omega_c(t-(\Delta t_1+\Delta t_2))} \\ & \cdot \frac{4(n_1 - i\kappa_1)(1 - n_1 + i\kappa_1)}{(n_1 - i\kappa_1 + 1)^3} \cdot \exp(-\frac{\omega_c \kappa_1 \Delta t_2}{n_1}) \end{aligned} \tag{6.9}$$

In this expression, the multi-path reflection model consists of only four unknown variables, i.e., the material parameters n_1, κ_1 and the geometric parameters $\Delta t_1, \Delta t_2$, and the independent variable time t. Therefore, the parameter estimation is formulated as finding the parameters $(\Delta t_1, \Delta t_2, n_1, \kappa_1)$ that minimize the discrepancy between the multi-path model A_{mp} and the measured THz time domain data g_t shown in (6.4).

In order to verify the data processing and the THz reflection models A_{sp} and A_{mp}, Sec. 6.3.1 will evaluate the discrepancy between the measured THz data and the multi-path reflection model.

6.2.2 *Loss Functions and Hamming Window*

The solution of the parameter estimation problem is formulated as the optimal parameters u_c^* of the optimization problem based on the loss function $\mathcal{L}_{\mathrm{c}}$, namely the *magnitude of complex difference*, which is also the ℓ^2-square loss in previous chapters.

$$\begin{aligned} u_c^* &= \arg\min_u \mathcal{L}_{\mathrm{c}}(g_t; u) \\ \mathcal{L}_{\mathrm{c}}(g_t; u) &= \frac{1}{N}\sum_t |A_{\mathrm{mp}}(t; \Delta t_1, \Delta t_2, n_1, \kappa_1) - g_t|^2 \end{aligned} \tag{6.10}$$

where $|\cdot|$ denotes the magnitude of complex number and $u = (\Delta t_1, \Delta t_2, n_1, \kappa_1)$.

However, even if a simple ℓ^2-square loss is chosen, apparently it is difficult for the optimizer to find the global minima of this highly non-convex optimization problem. In order to investigate other possible simplification of the optimization problem, another loss function $\mathcal{L}_{\mathrm{m}}$, i.e. the *difference of magnitude*, is also studied for comparison:

$$\begin{aligned} u_m^* &= \arg\min_u \mathcal{L}_{\mathrm{m}}(g_t; u) \\ \mathcal{L}_{\mathrm{m}}(g_t; u) &= \frac{1}{N}\sum_t \left(|A_{\mathrm{mp}}(t; \Delta t_1, \Delta t_2, n_1, \kappa_1)| - |g_t|\right)^2 \end{aligned} \tag{6.11}$$

However, as it will be shown in Sec. 6.3.2, the loss landscapes of both $\mathcal{L}_{\mathrm{c}}$ and $\mathcal{L}_{\mathrm{m}}$ show numerous local minima, which are mainly caused by the side-lobe of the time domain signal. One intuitive solution is to reduce the signal side-lobe amplitude by an additional operation of the Hamming window. Therefore, the THz multi-path model and also the measured THz data are both multiplied with the Hamming window in the frequency domain, where the Hamming window $H[k]$ in the discrete frequency domain is:

$$H[k] = 0.54 - 0.46\cos\left(2\pi \cdot \frac{k}{L}\right) \quad \text{, where } 0 \le k \le L-1 \tag{6.12}$$

Considering the measured discrete frequency signal $G_f[k]$ from (6.4), the discrete frequency signal multiplied with the Hamming window $\widetilde{G}_f[k]$ yields:

$$\widetilde{G}_f[k] = G_f[k-k_o] \cdot H[k-k_o] \tag{6.13}$$

After the IFFT, the discrete time signal with Hamming window $\widetilde{g}_t[n]$ is expressed as:

$$\begin{aligned} \widetilde{g}_t[n] &= \left(\mathcal{F}^{-1}\left\{G_f[k]\right\} \cdot e^{i2\pi \frac{n}{N} \cdot k_o}\right) \circledast \left(\mathcal{F}^{-1}\left\{H[k]\right\} \cdot e^{i2\pi \frac{n}{N} \cdot k_o}\right) && (6.14) \\ &= g_t[n] \circledast h[n] && (6.15) \end{aligned}$$

where $\circledast$ denotes the convolution operator and the Hamming window in the discrete-time domain is defined as $h[n] = \mathcal{F}^{-1}\left\{H[k-k_o]\right\}$.

Algorithm 6.1 Peak-finding procedure of the multi-path initialization method

```
Input: time domain data $g_t$, time domain sampling grid $t$
function PEAKFINDING($g_t, t$)
    $\nabla g_t$ = Differentiate $|g_t|$;
    $|g_t|^p$ = Find all maxima of $|g_t|$ by zero-crossing $\nabla g_t$;    ▷ peaks
    $\hat{t}_1$ = time position of largest maxima $|g_t|^p_1$;
    $\hat{t}_2$ = time position of second largest maxima $|g_t|^p_2$;
    Compare position of $\hat{t}_1$ and $\hat{t}_2$ and ratio of $|g_t|^p_1$ and $|g_t|^p_2$;
    if Two peaks are found then
        $\hat{t}_2$ = time position of largest maxima after $\hat{t}_1$;    ▷ $\hat{t}_2 > \hat{t}_1$
    else
        Subtract first reflection signal from data $g_t$;
        $\hat{t}_2$ = largest maxima position of residual signal magnitude after $\hat{t}_1$;
    end if
    $t_1, P_1$ = quadratic interpolation near $\hat{t}_1$ on $|g_t|$;    ▷ $P_1$ First peak magnitude
    $t_2, P_2$ = quadratic interpolation near $\hat{t}_2$ on $|g_t|$;    ▷ $P_2$ Second peak magnitude
    Initialize $\Delta t_1 = t_1$ and $\Delta t_2 = t_2 - t_1$
return peak magnitude $P_1$, $P_2$ and time $\Delta t_1$, $\Delta t_2$
end function
```

Correspondingly, the multi-path reflection model is also extended by the Hamming window convolution, which yields the windowed THz multi-path reflection model $\widetilde{A}_{\mathrm{mp}}$:

$$\begin{aligned}\widetilde{A}_{\mathrm{mp}}(t; \Delta t_1, \Delta t_2, n_1, \kappa_1) &= A_{\mathrm{mp}}(t; \Delta t_1, \Delta t_2, n_1, \kappa_1) \circledast h(t) \\ \text{where} \quad h(t) &= \mathcal{F}^{\text{-}1}\left\{H[k]\right\} \cdot \exp\left(i2\pi(f_c - \frac{B}{2})t\right)\end{aligned} \tag{6.16}$$

Note that the Hamming window $h(t)$ is independent to the desired parameters $u = (\Delta t_1, \Delta t_2, n_1, \kappa_1)$, which only changes the loss functions of optimization problems in (6.10) and (6.11) to:

$$\widetilde{\mathcal{L}}_{\mathrm{c}}(\widetilde{g}_t; u) = \frac{1}{N}\sum_t |\widetilde{A}_{\mathrm{mp}}(t; \Delta t_1, \Delta t_2, n_1, \kappa_1) - \widetilde{g}_t|^2 \tag{6.17}$$

$$\widetilde{\mathcal{L}}_{\mathrm{m}}(\widetilde{g}_t; u) = \frac{1}{N}\sum_t \left(|\widetilde{A}_{\mathrm{mp}}(t; \Delta t_1, \Delta t_2, n_1, \kappa_1)| - |\widetilde{g}_t|\right)^2 \tag{6.18}$$

6.2.3 *Optimization*

In this part, the implementation details of the optimization process are described.

INITIALIZATION In this section, the initialization method for the THz multi-path reflection model A_{mp} is described, which starts with a peak-finding algorithm for $\Delta t_1, \Delta t_2$ depicted in Algorithm 6.1. The peak-finding algorithm returns the magnitude of two peaks P_1 and P_2, and the initialized parameters $\Delta t_1'$ and $\Delta t_2'$.

Next, according to the A_{mp} model from (6.8), n_1 can be roughly estimated by the magnitude of first peak P_1 and the reflection coefficient r_{01}, given the prior-knowledge that the refractive index n_1 is generally much larger than the dielectric coefficient κ_1, i.e. $n_1 \gg \kappa_1$ and $\eta_1 \approx n_1$. The initialized parameter n_1' yields:

$$\begin{aligned} P_1 &\approx A_0 \cdot r_{01} \\ r_{01} &= \frac{\eta_1 - 1}{\eta_1 + 1} \approx \frac{n_1 - 1}{n_1 + 1} \\ \Rightarrow \quad n_1' &\approx -\frac{\frac{P_1}{A_0} + 1}{\frac{P_1}{A_0} - 1} = -\frac{P_1 + A_0}{P_1 - A_0} \end{aligned} \tag{6.19}$$

Finally, the estimation of κ_1 depends on the ratio of P_1 and P_2 (also assuming $\eta_1 \approx n_1$), by compariing coefficients of the first reflection C_1 and the second reflection C_2:

$$\begin{aligned} C_1 &= \frac{\eta_1 - 1}{\eta_1 + 1} \approx \frac{n_1 - 1}{n_1 + 1} \\ C_2 &= \frac{4\eta_1(\eta_1 - 1)}{(\eta_1 + 1)^3} \approx \frac{|4n_1(n_1 - 1)|}{|n_1 + 1|^3} \\ \frac{P_2}{P_1} &\approx \frac{C_2}{C_1} \cdot \exp\left(-\frac{\omega_c \kappa_1 \Delta t_2}{n_1}\right) \\ \Rightarrow \quad \kappa_1' &\approx \left| \frac{-n_1}{\omega_c \Delta t_2} \cdot \ln\left(\frac{C_1}{C_2} \cdot \frac{P_2}{P_1}\right) \right| \end{aligned} \tag{6.20}$$

Therefore, using the peak-finding procedure shown in Algorithm 6.1, and equations (6.19) and (6.20), the multi-path initialization method obtains the coarse estimation of desired parameters $u' = (\Delta t_1', \Delta t_2', n_1', \kappa_1')$ based on the discrete time signal g_t, which this estimation will be served as an initial point for the optimization.

OPTIMIZERS To solve the optimization problems, firstly reasonable boundaries of each parameter, i.e. constraints in mathematical terminology, are determined based on some commonly known physical properties:

$1 \leq n_1 \leq 30$	commonly $n_1 \leq 8$ for semiconductors and plastics
$0 \leq \kappa_1 \leq 1$	where $\kappa_1 = 0.71$ for water at 577GHz [LHM 91]
$\Delta t_2 > 0$	assuming a positive material thickness

Three optimizers are studied for their ability to estimate the parameters:

- **OptimM**: optimization on signal magnitude by minimizing $\mathcal{L}_\mathrm{m}$ in (6.11).
- **OptimC**: optimization on signal complex parts by minimizing $\mathcal{L}_\mathrm{c}$ in (6.10).
- **OptimJ**: joint optimization on both magnitude and complex parts by:
    1. Optimization on the signal magnitude by minimizing $\mathcal{L}_\mathrm{m}$ in (6.11).
    2. Initialized by optimal values u' at the first step, optimization on signal complex parts by minimizing $\mathcal{L}_\mathrm{c}$ in (6.10), using the refined boundary $u' \pm 20\%$.

In order to study the effect of Hamming window, all these 3 optimizers (i.e. **OptimM**, **OptimC**, **OptimJ**) are also tested for $\widetilde{\mathcal{L}}_\mathrm{m}$, $\widetilde{\mathcal{L}}_\mathrm{c}$ and jointly $\widetilde{\mathcal{L}}_\mathrm{m} + \widetilde{\mathcal{L}}_\mathrm{c}$ respectively.

For the optimization algorithm, all optimizers are run by per-pixel Trust-Region Algorithm (TRA) implemented on MATLAB® , because TRA has shown superior performance in previous Chapters 4 and 5 for the per-pixel optimization problem. Note that other variants of the optimization method, namely the AdamW optimizer and Gaussian distributed weights of loss, are also tested. However, these variants do not show better performance, so they are not shown in the experimental result section.

6.3 EXPERIMENTAL RESULT

Sec. 6.3.1 compares the measured THz data to the data derived by the THz reflection models based on the measured reference value of materials to verify the THz physical models. Sec. 6.3.2 theoretically analyzes the loss function and the maximum boundary of initialization from a numerical perspective. Sec. 6.3.3 evaluates the material and geometric parameter estimation approach according to the parameter accuracy, with respect to the error statistics and the estimated parameter value.

6.3.1 *Verification of THz Models*

In order to verify the correctness of the single-path reflection model in (6.6) and the multi-path reflection model in (6.9), the models are compared to the data extracted from an exemplary position for each material in the measured THz datasets **MPTarget**.

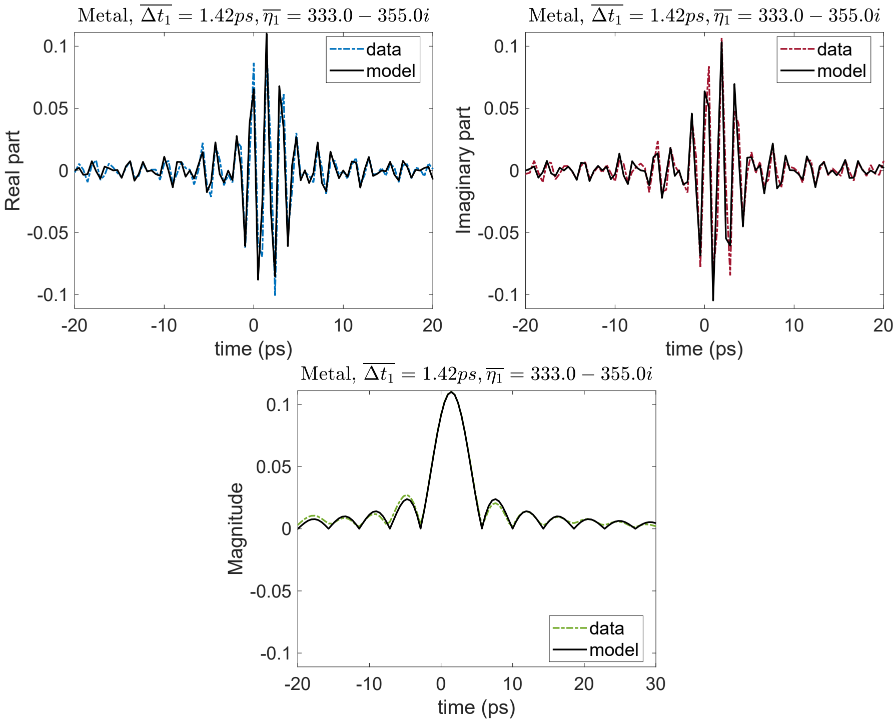


Figure 6.4: The measured THz signal (i.e. real, imaginary parts and signal envelope) at the metal position from **MPTarget** datasets match to the THz single-path reflection model in (6.6).

THZ SINGLE-PATH REFLECTION MODEL Fig. 6.4 plots one example measured THz data g_t in (6.4) and the THz single-path reflection model A_{sp} in (6.6) at metal position. The reference complex refractive index $\overline{\eta_1} = 333 - 355i$ is calculated from the copper refractive index from [HGK 75] by linear extrapolation to $300\,\mathrm{GHz}$. The reference depth position $\overline{\Delta t_1}$ is taken by empirically minimizing the difference between the data and the model.

The result shows that the single-path model A_{sp} generally matches the measured data for the signal envelope (magnitude), the oscillation frequency, and the phase.

THZ MULTI-PATH REFLECTION MODEL Fig. 6.5 plots the signal envelope (i.e. magnitude) of an exemplary measured THz data and the THz multi-path reflection model A_{mp} in (6.9) for each dielectric material - FuseSilica (top-left), Silicon (top-right), PVC (bottom-left) and COC (bottom-right). The reference complex refractive index $\overline{\eta_1}$ and the reference material thickness $\overline{\Delta t_2}$ are taken from Table 2.1 in Chapter 2. The reference depth position $\overline{\Delta t_1}$ is determined by empirically minimizing the difference between the measurement data and the model.

Fig. 6.6 plots the real and imaginary parts of an exemplary measured THz data and the THz multi-path reflection model A_{mp} for

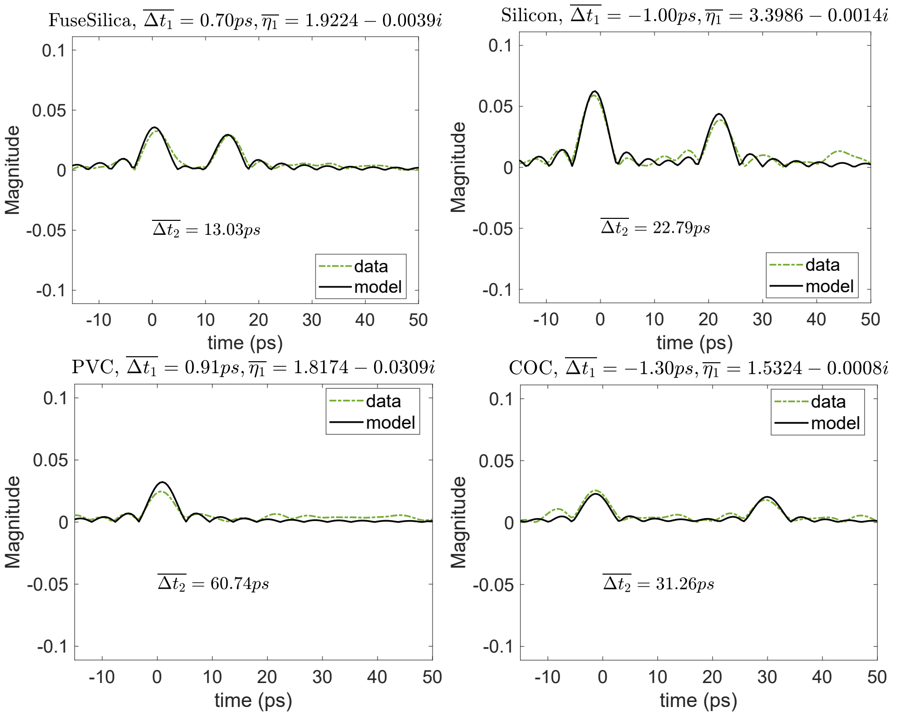
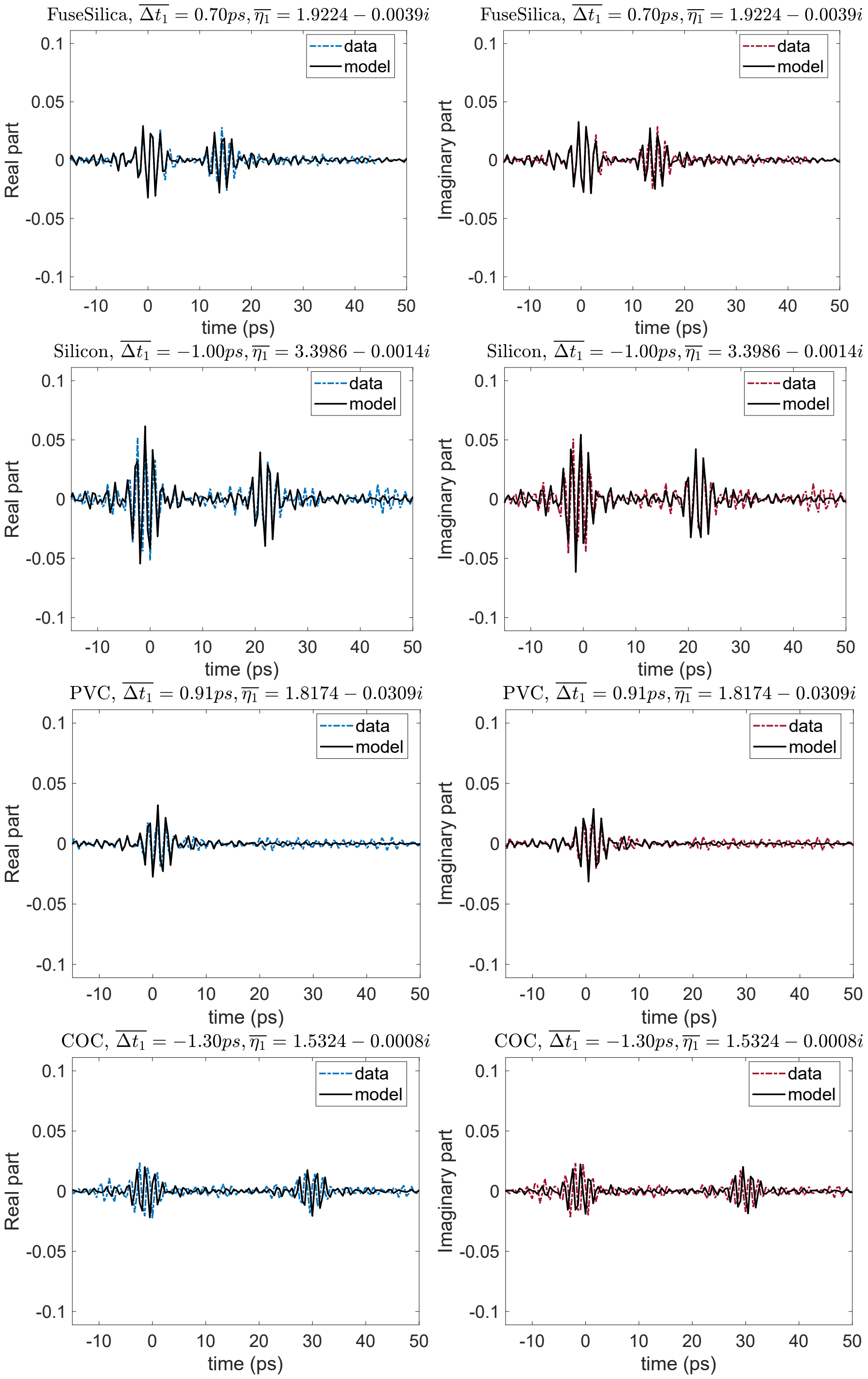


Figure 6.5: The measured signal envelop (i.e. magnitude) from **MPTarget** datasets matches to the THz multi-path reflection model in (6.9) for each dielectric material - FuseSilica (top-left), Silicon (top-right), PVC (bottom-left) and COC (bottom-right).

each dielectric material - FuseSilica (top-left), Silicon (top-right), PVC (bottom-left) and COC (bottom-right).

DISCUSSION From the figures of signal envelopes, the measured data (green dotted line) commonly match the physical model (black line) for the peak positions and the peak magnitudes for FuseSillica, Silicon, and COC, except for only a slight mismatch on PVC. The data magnitude of Silicon (top-right plot) also shows that a third-path reflection can be detectable after the second reflection path. This is because the dielectric material Silicon has a very low propagation decay term (see (6.8) α_1), which is exponentially related to the thin thickness Δt_2 and the low dielectric coefficient κ_1 (see Table 2.1). Although it would be very interesting to include all paths from reflections for consideration, this full-paths reflection is neglected in this section.

Moreover, as the reference values in Table 2.1 shown, the PVC material is indeed a thick material with a high dielectric coefficient, in which both the measured data and signal model show only the first peak signal and the non-detectable second peak. This single peak signal envelope can disturb simple peak-finding methods to locate a precise and robust second peak, and hence a more detailed peak-finding method (as depicted in Algorithm 6.1) is needed to enhance the robustness of initialization.



FuseSilica, $\overline{\Delta t_1} = 0.70ps, \overline{\eta_1} = 1.9224 - 0.0039i$

FuseSilica, $\overline{\Delta t_1} = 0.70ps, \overline{\eta_1} = 1.9224 - 0.0039i$

Silicon, $\overline{\Delta t_1} = -1.00ps, \overline{\eta_1} = 3.3986 - 0.0014i$

Silicon, $\overline{\Delta t_1} = -1.00ps, \overline{\eta_1} = 3.3986 - 0.0014i$

PVC, $\overline{\Delta t_1} = 0.91ps, \overline{\eta_1} = 1.8174 - 0.0309i$

PVC, $\overline{\Delta t_1} = 0.91ps, \overline{\eta_1} = 1.8174 - 0.0309i$

COC, $\overline{\Delta t_1} = -1.30ps, \overline{\eta_1} = 1.5324 - 0.0008i$

COC, $\overline{\Delta t_1} = -1.30ps, \overline{\eta_1} = 1.5324 - 0.0008i$

Figure 6.6: The real (left column) and imaginary (right column) parts of measured signal from **MPTarget** datasets matches to the THz multi-path reflection model in (6.9) for each dielectric material - (from top to bottom) FuseSilica, Silicon, PVC, and COC.

Besides the magnitude, the real and imaginary parts of measured data also match the forward model in terms of its oscillation frequency and the phase, except that a minor phase offset for COC is observed. One possible reason for this phase offset is that the phase of the THz multi-path reflection model in (6.8) is largely composite of the material parameters n_1 and κ_1, while these two parameters are in fact frequency dependent (Fig. 2.25) and are simplified (i.e. averaged) to one single coefficient in the model. This frequency-dependent discrepancy can introduce the observed modeling discrepancy, and hence limit the accuracy of the derived physical models. Further studies on these frequency-dependent parameters can be helpful to enhance the robustness of the THz physical model.

Nevertheless, in this part, the experiment verifies that the THz single-path and multi-path reflection models generally match the measured THz data, and hence the models can serve as the reference forward model to solve the optimization problem of parameter estimation.

6.3.2 *Theoretical Analysis of Loss Functions*

In deep learning terminology, a loss landscape plot can visualize how the loss value is changed during neural network parameters training [Li+ 18]. This is commonly used for understanding the non-convexity and the efficiency of highly non-convex optimization problems, such as deep neural network training. In order to have a proper analysis and a comparison of the loss functions mentioned in Sec. 6.2.2, i.e. $\mathcal{L}_\mathrm{m}$, $\mathcal{L}_\mathrm{c}$, $\widetilde{\mathcal{L}}_\mathrm{m}$ and $\widetilde{\mathcal{L}}_\mathrm{c}$, in this section the loss landscape of all these loss functions are studied *theoretically*. More precisely, the theoretical 1-Dimensional loss functions with respect to each parameter $(\Delta t_1, \Delta t_2, n_1, \kappa_1)$ near the global optimal are plotted, based on the *ideal* (i.e. noiseless) synthetic datasets **SynthMPT** and **SynthMPT$*$Hamm**.

Then, based on these loss function plots, gradient-based optimizers (i.e. TRA optimizer for all loss functions individually) are applied to the ideal synthetic datasets based on different parameter initialization. By comparison of the maximum range of parameter initialization that can find the global minima, the valid range of parameter initialization is determined for each optimizer and hence it can indicate the option of loss function and optimizer combination.

Fig. 6.7 plots 1-D loss landscapes by varying $\Delta t_1, \Delta t_2, n_1, \kappa_1$ individually, while all other three parameters are fixed to the ground truth values. Ground truth values are using reference values of FuseSilica as an example (more results are shown in Table 6.1). Note that the data is simulated synthetically without any noise or discrepancy, which indicates that the ground truth value (depicts in red color) serves as the theoretical global minima, i.e. the zero loss value. Four plots in the left column show 1-D loss functions $\mathcal{L}_\mathrm{c}$ and $\mathcal{L}_\mathrm{m}$ based on A_mp model

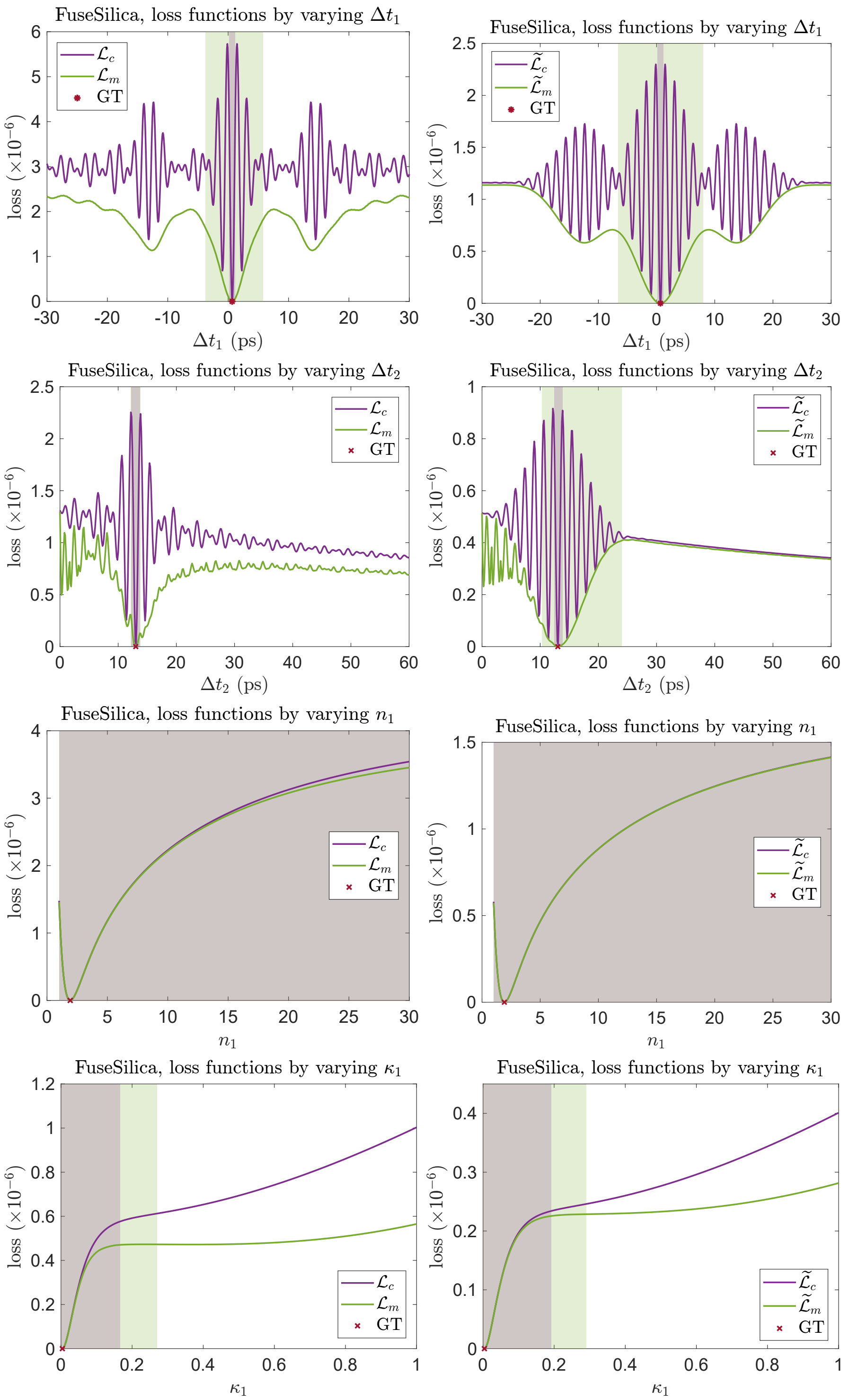


Figure 6.7: 1-D Loss landscapes are plotted by varying $\Delta t_1, \Delta t_2, n_1$ and κ_1 individually using reference value of FuseSilica as ground truth value. Left column plots show loss functions $\mathcal{L}_\text{c}$ and $\mathcal{L}_\text{m}$; Right column plots show loss functions with Hamming window $\widetilde{\mathcal{L}}_\text{c}$ and $\widetilde{\mathcal{L}}_\text{m}$. The red cross denotes the ground truth (GT) loss (i.e. the global minima). The colored regions (purple for $\mathcal{L}_\text{c}$ and green for $\mathcal{L}_\text{m}$) represents the maximum boundary of initialization (see Table 6.1).

Model				A_{mp} in Eq. (6.9)			$\widetilde{A}_{\mathrm{mp}}$ in Eq. (6.16)		
Parameter		Loss	GT	Lower bound	Upper bound	Maximum boundary	Lower bound	Upper bound	Maximum boundary
FuseSilica (Fig. 6.7)	Δt_1	$\mathcal{L}_{\mathrm{c}}$	0.6671	0.1384	1.1492	1.0108	0.1229	1.1617	**1.0388**
	Δt_2		13.0298	12.2652	13.8018	**1.5366**	12.4200	13.8323	1.4123
	n_1		1.9224	1.0002	30.0000	**28.9998**	1.0002	30.0000	**28.9998**
	κ_1		0.0039	0.0000	0.1656	0.1656	0.0000	0.1915	**0.1915**
	Δt_1	$\mathcal{L}_{\mathrm{m}}$	0.6671	-3.7457	5.7578	9.5035	-6.5922	7.9780	**14.5701**
	Δt_2		13.0298	12.1414	13.7738	1.6323	10.2943	24.0171	**13.7228**
	n_1		1.9224	1.0038	30.0000	**28.9962**	1.0002	30.0000	**28.9998**
	κ_1		0.0039	0.0000	0.2696	0.2696	0.0000	0.2904	**0.2904**
Silicon	Δt_1	$\mathcal{L}_{\mathrm{c}}$	0.6671	0.1800	1.0818	0.9017	0.1732	1.0885	**0.9153**
	Δt_2		22.7865	21.8962	23.4392	1.5429	22.0076	23.5836	**1.5760**
	n_1		3.3986	1.0048	30.0000	**28.9952**	1.0001	30.0000	**28.9999**
	κ_1		0.0014	0.0000	0.2672	**0.2672**	0.0000	0.2613	**0.2613**
	Δt_1	$\mathcal{L}_{\mathrm{m}}$	0.6671	-6.0362	7.3547	13.3909	-9.0985	12.3998	**21.4983**
	Δt_2		22.7865	21.8288	23.5499	1.7210	11.3710	34.3046	**22.9335**
	n_1		3.3986	1.0064	30.0000	**28.9936**	1.0066	30.0000	**28.9934**
	κ_1		0.0014	0.0000	0.1186	0.1186	0.0000	0.4024	**0.4024**
PVC	Δt_1	$\mathcal{L}_{\mathrm{c}}$	0.6671	-0.1066	1.1339	**1.2405**	-0.1159	1.0352	1.1511
	Δt_2		60.7444	59.7042	61.3568	**1.6526**	59.7579	61.3738	1.6158
	n_1		1.8174	1.2117	30.0000	**28.7883**	1.2192	30.0000	**28.7808**
	κ_1		0.0309	0.0000	0.0624	**0.0623**	0.0000	0.0573	0.0573
	Δt_1	$\mathcal{L}_{\mathrm{m}}$	0.6671	-0.6308	1.9670	2.5979	-4.4714	7.4538	**11.9252**
	Δt_2		60.7444	60.0096	61.6083	1.5986	59.9634	61.6465	**1.6831**
	n_1		1.8174	1.4313	10.8181	9.3868	1.3695	14.0963	**12.7269**
	κ_1		0.0309	0.0000	0.0621	**0.0621**	0.0000	0.0603	**0.0603**
COC	Δt_1	$\mathcal{L}_{\mathrm{c}}$	0.6671	0.1444	1.1757	1.0312	0.1356	1.1799	**1.0443**
	Δt_2		31.2621	30.6578	31.8683	1.2105	30.6533	31.9865	**1.3332**
	n_1		1.5324	1.0001	30.0000	**28.9999**	1.0001	30.0000	**28.9999**
	κ_1		0.0008	0.0000	0.0777	**0.0777**	0.0000	0.0515	0.0515
	Δt_1	$\mathcal{L}_{\mathrm{m}}$	0.6671	-10.6904	11.5906	**22.2810**	-10.5210	11.6037	22.1247
	Δt_2		31.2621	27.0764	34.9217	7.8454	18.3212	44.1705	**25.8492**
	n_1		1.5324	1.0012	4.9711	3.9699	1.0001	30.0000	**28.9999**
	κ_1		0.0008	0.0000	0.0164	0.0164	0.0000	0.1227	**0.1227**

Table 6.1: Comparison of the ground truth (GT) values to the maximum boundary of initialization by TRA optimizer for $\mathcal{L}_c$ and $\mathcal{L}_m$ loss functions using A_{mp} and $\widetilde{A}_{\mathrm{mp}}$ model. The wider (better) boundary of initialization is highlighted.

using **SynthMPT** datasets. For comparison to the loss functions with Hamming window, four plots in the right column show loss functions $\widetilde{\mathcal{L}}_{\mathrm{c}}$ and $\widetilde{\mathcal{L}}_{\mathrm{m}}$ based on $\widetilde{A}_{\mathrm{mp}}$ model using **SynthMPT$*$Hamm** datasets.

The maximum boundary of initialization (Fig. 6.7 and Table 6.1) is calculated by the maximum range of initialization point that can resolve the ground truth value (i.e. obtain the zero loss) by the TRA optimizer. A wider valid range implies that the accuracy requirement of parameter initialization to resolve the ground truth value is lower.

DISCUSSION The loss landscape (Fig. 6.7, using FuseSilica as an example) shows that the model with the Hamming window (right column) has a significantly smoother landscape than the model without the Hamming window (left column) for the geometric parameters Δt_1 and Δt_2. When the width of the colored region is compared (i.e. maximum boundary of initialization) between the model with or without Hamming window, the result shows a significantly wider range of initialization that obtains the global minima, indicating that a much looser initialization accuracy is required by optimization for *both* magnitude loss function $\mathcal{L}_\mathrm{m}$ and complex loss function $\mathcal{L}_\mathrm{c}$. For the other materials, Table 6.1 demonstrates that the model with Hamming window $\widetilde{A}_\mathrm{mp}$ generally obtains a wider boundary of initialization than the model without Hamming window A_mp, except a slightly narrower boundary for Δt_1 of FuseSilica using complex loss function $\mathcal{L}_\mathrm{c}$, for PVC using complex loss function $\mathcal{L}_\mathrm{c}$ and for Δt_1 of COC using magnitude loss function $\mathcal{L}_\mathrm{m}$.

Regarding the material parameters, the refractive index n_1 shows that most of the optimizers tested can obtain the global minima in the tested range (i.e. $1 \leq n_1 \leq 30$), while the model with Hamming window $\widetilde{A}_\mathrm{mp}$ still out-performs to A_mp in the other cases. The analysis of the dielectric coefficient κ_1 also demonstrates a superior performance by the model with Hamming window $\widetilde{A}_\mathrm{mp}$ for FuseSilia, Silicon, and COC, while an approximately equal maximum boundary of initialization by PVC.

On the other perspective of comparison, when the complex loss function $\mathcal{L}_\mathrm{c}$ is compared to the magnitude loss function $\mathcal{L}_\mathrm{m}$ (i.e. comparing Table 6.1 vertically for each material), the analytical results show that the magnitude loss function $\mathcal{L}_\mathrm{m}$ allows a wider range of initialization than the complex loss function $\mathcal{L}_\mathrm{c}$ significantly for geometric parameters Δt_1 and Δt_2. For dielectric coefficient κ_1, this gap is still observable for the model with Hamming window $\widetilde{A}_\mathrm{mp}$, while $\mathcal{L}_\mathrm{m}$ is only better on FuseSilica for the model without Hamming window A_mp. For refractive index n_1, all loss functions and optimizers obtain the maximum test range for FuseSilica and Silicon materials. Even though the magnitude loss function $\mathcal{L}_\mathrm{m}$ shows a slightly stricter boundary of initialization than $\mathcal{L}_\mathrm{c}$ for PVC and COC, the boundaries are still larger than the commonly known range of refractive index (i.e. $n_1 \leq 8$) for semi-conductor and plastic.

Therefore, the analytical results demonstrate that the optimizer **OptimM**, which tests on the magnitude loss function $\mathcal{L}_\mathrm{m}$ and the signal envelope A_mp, is generally better than the optimizer **OptimC** because of the less initialization accuracy required. The model with Hamming window $\widetilde{A}_\mathrm{mp}$ can further reduce the initialization accuracy needed, and hence further enhance the robustness of optimization.

However, note that this theoretical analysis is only limited to the 1-Dimension loss landscape (i.e. the simplest case) *without* any additional noise, which can easily over-simplify the real scenario of the multi-dimensional loss functions. Hence, it is necessary to verify the parameter estimation by the initialization point and the optimization using the measured THz datasets, as it will be shown in the next section.

Furthermore, in principle, a *sharper* and smooth loss landscape should be better for fast optimization, but the nearly flat loss landscape $\mathcal{L}_{\mathrm{m}}$ of κ_1 in Fig. 6.7 shows a wider range of initialization than the sharper loss landscape $\mathcal{L}_{\mathrm{c}}$. This is probably due to the fact that the initialization range is solely tested by the TRA optimizer, which as a second-order gradient-based optimizer considers not only the local slope but also the nearby curvature, and provides a less intuitive result than a first-order gradient-based optimizer. Also, the convergence rate (i.e. the number of steps to find global minima) is not recorded either, which makes it unclear the level of improvement by different optimizers. This theoretical analysis can be more conclusive if more optimization algorithms are included for testing.

6.3.3 *Parameter Accuracy*

The optimization performance is evaluated by the measurement datasets **MPTarget** and the synthetic datasets **SynthMPT**. Each optimizer is run for 5×5 pixels for each material at the center region. Statistics numbers are shown for 10% percentiles, 50% percentiles (i.e. median), and 90% percentiles. The estimated parameters $(\Delta t_1, \Delta t_2, n_1, \kappa_1)$ are compared to the reference value shown in Chapter 2 Sec. 2.5.2.2. Note that Δt_1 is removed from figures and table because the reference depth position (i.e. Δt_1) from the THz datasets are not measured.

PARAMETER ERRORS Fig. 6.8 plots the median error of $\Delta t_2, \Delta z_2, n_1, \kappa_1$ by optimizers **OptimM**, **OptimC**and **OptimJ** to the initialization (Init.) in log scale using the measured THz datasets **MPTarget** and **MPTarget∗Hamm**. Errors are calculated by the absolute difference between the estimated value to the measured reference value shown in Table 2.1.

Fig. 6.8 plots the error distribution (i.e. 10%, median and 90% percentiles) of Δt_2, Δz_2, n_1, κ_1 by optimizers to the initialization using the measured THz datasets **MPTarget** and **MPTarget∗Hamm**. Errors are calculated by the absolute difference between the estimated value to the measured reference value shown in Table 2.1.

Table 6.2 depicts the median error percentage of parameters Δt_2, Δz_2, n_1, κ_1 estimated by optimizers to the initialization (Init.), using the measured THz datasets **MPTarget** and **MPTarget∗Hamm**. Median error percentages are calculated by dividing the median of absolute difference error by the ground truth value.

DISCUSSION The error distributions from Fig. 6.8 and Fig. 6.9 show that the optimization on signal magnitude **OptimM** is out-performing to the optimization on signal complex parts **OptimC** and **OptimJ** for the parameters $\Delta t_1, \Delta z_2, \kappa_1$. However, the experiment result is not robust enough to draw a strong conclusion, about whether the opti-

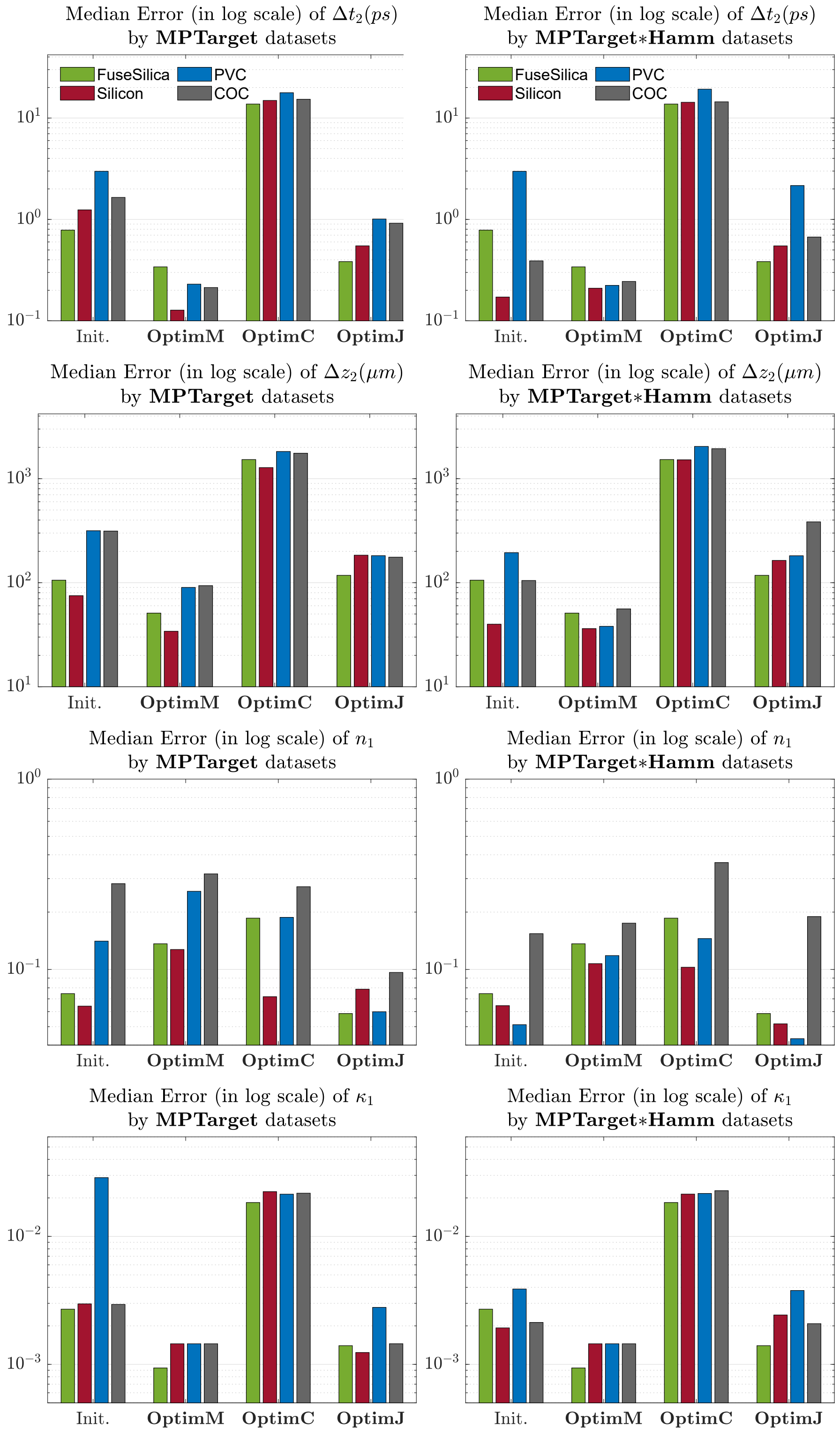


Figure 6.8: Comparison of the median error of parameters $\Delta t_2, \Delta z_2$ (i.e. material thickness), n_1, κ_1 to the initialization (Init.) in log scale using the measured THz datasets **MPTarget** (left column) and **MPTarget∗Hamm** (right column). Errors are calculated by the absolute difference to the measured reference value shown in Table 2.1. Bars indicate the median of error (the lower is better).

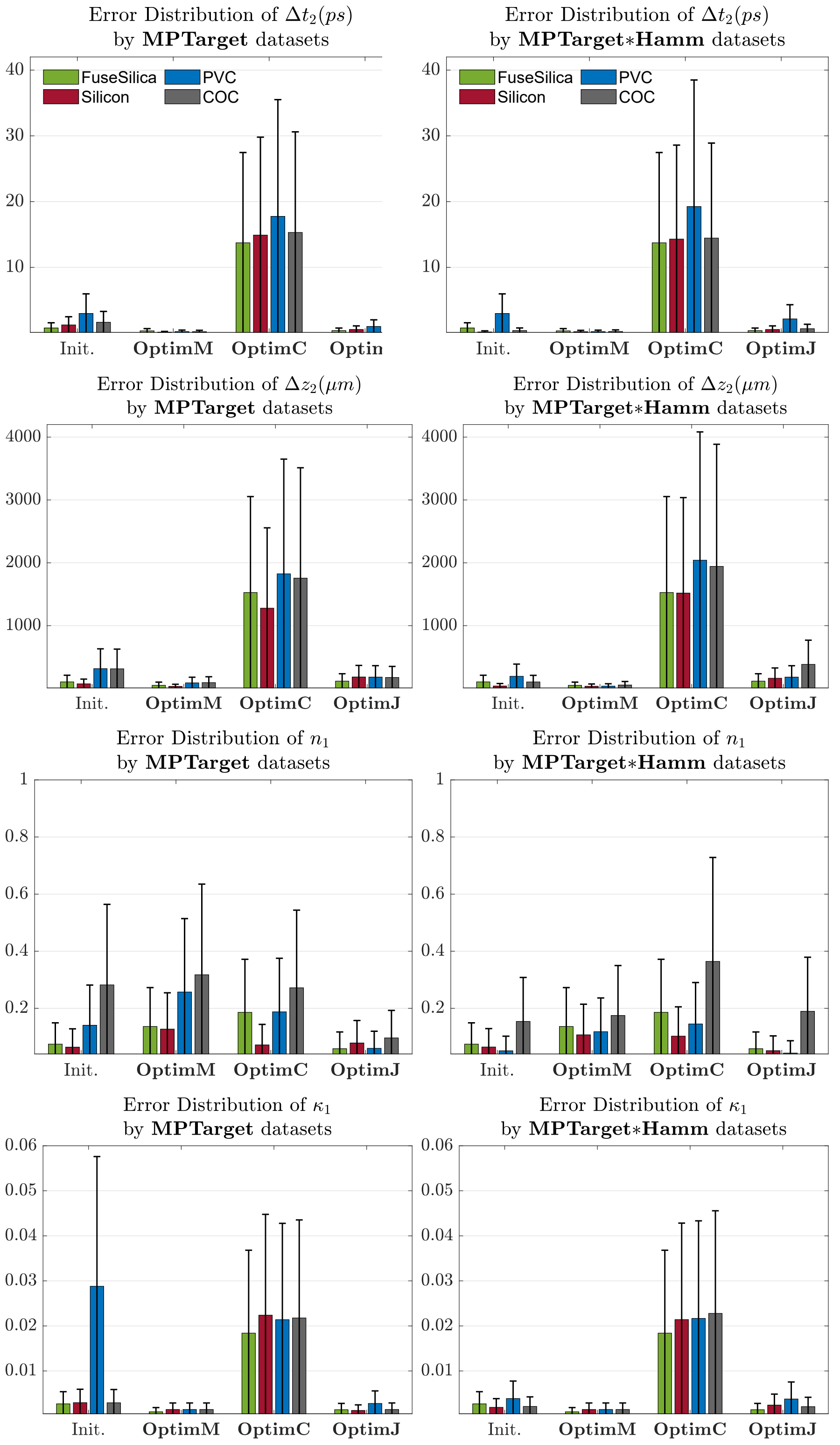


Figure 6.9: Comparison of the error distribution of parameters $\Delta t_2, \Delta z_2$ (i.e. material thickness), n_1, κ_1 to the initialization (Init.) using the measured THz datasets **MPTarget** (left column) and **MPTarget∗Hamm** (right column). Errors are calculated by the absolute difference to the measured reference value shown in Table 2.1. Bars and whiskers indicate the median, 10%, and 90% percentiles respectively.

Datasets	Optimizer	FuseSilica				Silicon			
		Δt_2(%)	Δz_2(%)	n_1(%)	κ_1(%)	Δt_2(%)	Δz_2(%)	n_1	κ_1
	Init.	6.0	10.4	3.9	69.6	1.5	5.1	4.0	**64.7**
MPTarget ∗Hamm	**OptimM**	**1.3**	**3.9**	3.4	**49.7**	0.9	3.6	**3.2**	100.0
	OptimC	22.9	19.1	**2.7**	100.0	1.0	3.8	3.5	100.0
	OptimJ	3.0	10.3	8.0	54.8	1.1	5.6	5.1	100.0
MPTarget	**OptimM**	9.5	7.4	3.3	76.6	**0.6**	**3.4**	3.7	100.0
	OptimC	22.9	31.1	7.3	742.1	1.0	9.0	7.6	100.0
	OptimJ	12.7	30.9	14.7	75.8	0.9	9.3	9.3	100.0

Datasets	Optimizer	PVC				COC			
		Δt_2(%)	Δz_2(%)	n_1(%)	κ_1(%)	Δt_2(%)	Δz_2(%)	n_1(%)	κ_1(%)
	Init.	**22.6**	30.5	10.2	**59.6**	**1.2**	**3.8**	3.8	170.8
MPTarget ∗Hamm	**OptimM**	23.5	30.3	5.6	69.4	1.8	5.4	3.4	296.5
	OptimC	31.7	40.7	8.0	70.2	6.9	5.9	**2.8**	460.8
	OptimJ	23.8	38.8	20.0	73.8	2.1	12.6	12.4	253.7
MPTarget	**OptimM**	24.5	**25.5**	**4.0**	72.5	1.8	6.0	5.1	**151.0**
	OptimC	29.2	36.4	10.3	69.3	3.2	5.9	3.9	340.1
	OptimJ	25.2	35.1	15.0	70.5	2.9	5.8	6.3	176.8

Table 6.2: Comparison of the median error percentage of parameters $\Delta t_2, \Delta z_2, n_1, \kappa_1$ estimated by optimizers to the initialization (Init.), using the measurement THz datasets **MPTarget** and **MPTarget∗Hamm**. The best optimization result is highlighted.

mizer **OptimM** *always* improves the parameter estimation regardless of the initialization accuracy. Indeed, it only shows a weak tendency, that the optimization method *can* improve the material and geometric estimation problem given that the initialization is reasonably accurate.

On the other hand, the signal magnitude optimizer **OptimM** obtains a promising improvement to the complex part optimizer **OptimC** with respect to the parameters $\Delta t_2, \Delta z_2, \kappa_1$. The percentiles in Fig. 6.9 also show that **OptimM** is more stable than **OptimC** in terms of the divergence of estimated parameter. This is novel to the common knowledge in the THz imaging community, that the complex signal should contain more information and hence should be superior to solve inverse problems like material and geometric parameter estimation. Therefore, the experimental results demonstrate that the optimization of the signal envelope (i.e. magnitude) can be more robust than the optimization of the complex signal for the material and geometric estimation problem using the THz multi-path reflection model. These results also support the finding of the theoretical analysis in Sec. 6.3.2, which analytically shows that the magnitude loss landscape obtains less local minima and hence the optimization of signal magnitude is easier to find the global minima.

For the Hamming window comparison, the median error shows that the model with Hamming window $\widetilde{A}_{\text{mp}}$ (i.e. right column of Fig. 6.8) can generally improve the initialization result, only except for n_1 of FuseSilica and Silicon. As the error distribution shown in Fig. 6.9, this improvement is probably because of the enhanced robustness (i.e. lower variance) of the initialized parameters by the Hamming window model. This robust initialization, hence, can improve the optimizers to reduce the median error and error variance of results with Hamming window, as shown in the right column of Fig. 6.8 and Fig. 6.9. However, again this is still a weak tendency, as the optimizer **OptimC** does not obtain significant improvement by the model with Hamming window, which implies that there is still more work needed to study the effect of the $\widetilde{A}_{\mathrm{mp}}$ model.

Regarding the median error percentage of material and geometric parameters, Table 6.2 shows that the best optimizers are diverging between **OptimM** and **OptimC** using using the model with or without Hamming window. In general, indeed, the model with Hamming window obtains a slightly more accurate value for FuseSilica and Silicon materials. Besides, by comparing the error percentage of initialization and the best optimizer, the result demonstrates that the best optimizer (which is mainly the **OptimM** and **OptimC** for **MP-Target∗Hamm** datasets) generally obtains a more accurate value than the initialization, especially for low-κ materials such as FuseSilica and Silicon. However, the results are still not robust enough to show a conclusive observation, of whether the optimizer obtains more accurate values compared to the peak-finding based initialization method.

6.4 SUMMARY

In this chapter, the materials and geometric parameter estimation for the THz multi-path reflection model is expressed as solving the optimization problem of the physical model by an objective loss function by an optimizer.

Based on the principle of the THz signal model, this approach derives the comprehensive formula that transfers the measured data from the numerical perspective to the physical space and interprets the THz multi-path reflection as a functional model of the material and geometric parameters.

The theoretical analysis of loss functions provides a deeper understanding of the ℓ^2 loss functions used for the optimization problem, by plotting the magnitude and complex 1-D loss landscape based on the synthetic ideal datasets and THz reflection models. The maximum boundary of initialization is tested by the Trust-Region Algorithm (TRA) optimizer for each optimizer respectively, to indicate the required accuracy of initialization for the optimization. The numerical results demonstrate that the optimizer based on the magnitude loss function and the signal envelope, is generally better than the optimizer based on the complex parts of the signal, because of the wider range of initialization.

By solving the optimization problem using a per-pixel optimizer, the experimental results demonstrate that the optimization of the signal magnitude can be more robust than the optimization of the complex signal for the joint material and geometric estimation problem using the THz multi-path reflection model. However, the experiment result is not robust enough to conclude, whether the optimizer always improves the parameter estimation regardless of the accuracy of initialization. In fact, it only shows a weak tendency, that the optimization method *can* improve the joint material and geometric estimation problem given that the initialization is reasonably accurate.

As this task can, so far, not solve with sufficient robustness, further investigation of the optimization method is needed in the future.

7

CONCLUSION

SUMMARY In this dissertation, an optimization-based THz data and image enhancement conceptual model is introduced. In this context, inverse problems in THz data and image enhancement, namely parameter estimation, image reconstruction, and resolution enhancement, are expressed as mathematical optimization problems.

This concept includes the THz computational image enhancement method, which is based on the complex curve fitting method in the azimuth direction, which is expressed as an optimization problem solved by state-of-the-art optimizers, and deconvolution methods in the lateral domain. Experiments demonstrate that this method enhances the lateral resolution and depth accuracy beyond the diffraction limit.

To enhance the optimizer, this concept is incorporated with cutting-edge machine learning technology, namely the model-based autoencoder, for the image reconstruction problem. This model-based autoencoder approach trains a convolution neural network as a per-pixel optimizer, which allows training the corresponding network with realistic measured THz data in an unsupervised way, i.e. without ground truth. The neural network can serve as an excellent initialization scheme for gradient-based optimizers. Experiments also demonstrate that this autoencoder approach significantly improves the time required for the optimization procedure.

To further enhance the optimizer, this concept introduces the deep optimization prior approach, which expresses the pixel-wise non-convex THz model parameter estimation problem as the reparameterization of the spatially coupled 3D neural network. This approach largely extends the model-based autoencoder approach from the 1D optimization problem to 3D THz image space, which makes it computationally and memory-wise highly efficient. The experiment shows that the deep optimization prior approach enhances the estimated model parameters numerically and qualitatively. Utilizing a blind deconvolution method substantially improves the lateral resolution enhancement.

To represent model parameters in a more physical-based perspective, this concept gives a first insight into the joint material and geometric estimation approach for the THz multi-path reflection model. This approach expresses the parameter estimation as the direct optimal of the optimization problem, which is strictly based on the physically derived THz multi-path reflection model. It should hence allow per-pixel classical optimizers or modern neural network-based optimizers to be utilized. However, the experimental results are not robust enough to draw a strong conclusion, but only to show a weak tendency that the tested optimizer can improve the joint parameter estimation problem given that the initialization is reasonably accurate.

Therefore, despite the limitations found in this dissertation, the optimization-based THz enhancement method can generally overcome challenges faced by the modern THz computational imaging community, and it can demonstrate the advantages brought by the cross-disciplinary collaboration between the fields of THz imaging and visual computing.

FUTURE WORK Experiments in the joint material and geometric parameter estimation have shown significant room to be improved in terms of the accuracy and the robustness of the estimated parameters. Utilizing some renowned non-convex optimizers, such as neural network-based optimizers, is a possible solution to enhance the ability to find the global minima. The regularization of the parameters can also be included in the optimizers so that 3D optimizers (such as the deep optimization prior approach) can also be utilized to improve the robustness by adopting prior knowledge (e.g. the spatial smoothness, the possible range of material parameters).

Moreover, it is possible to extend the THz physical model from the two-path reflection model to the full (i.e. three, four, etc.) paths reflection model, such that the experimental setup allows more than one dielectric material to be stacked for testing. This generalization will probably introduce more sophisticated non-convex optimization problems, but indeed it is an opportunity to demonstrate the advance of cutting-edge machine learning technology.

[AA 16] Kiarash Ahi and Mehdi Anwar. “Developing terahertz imaging equation and enhancement of the resolution of terahertz images using deconvolution”. In: *SPIE Commercial+ Scientific Sensing and Imaging*. International Society for Optics and Photonics. 2016, 98560N–98560N (cit. on p. 35).

[AK 17] Brandon Amos and J. Zico Kolter. “OptNet: Differentiable Optimization as a Layer in Neural Networks”. In: *Proc. Int. Conf. on Machine Learning*. 2017 (cit. on p. 57).

[Andr+ 16] Marcin Andrychowicz et al. “Learning to learn by gradient descent by gradient descent”. In: *Proc. Int. Conf. on Neural Information Processing Systems (NIPS)*. 2016 (cit. on p. 57).

[BFD 05] Michael E Baginski, Daniel L Faircloth, and Manohar D Deshpande. “Comparison of two optimization techniques for the estimation of complex permittivities of multilayered structures using waveguide measurements”. In: *IEEE transactions on microwave theory and techniques* 53.10 (2005), pp. 3251–3259 (cit. on pp. 56, 92).

[BZR 18] Jan Barowski, Marc Zimmermanns, and Ilona Rolfes. “Millimeter-wave characterization of dielectric materials using calibrated FMCW transceivers”. In: *IEEE Transactions on Microwave Theory and Techniques* 66.8 (2018), pp. 3683–3689 (cit. on pp. 56, 92).

[Bish 06] Christopher M Bishop. “Pattern recognition”. In: *Machine learning* 128.9 (2006) (cit. on p. 19).

[BV 99] Volker Blanz and Thomas Vetter. “A Morphable Model for the Synthesis of 3D Faces”. In: *Proc. SIGGRAPH*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194. DOI: `10.1145/311535.311556` (cit. on p. 57).

[Bore 01] Glenn D Boreman. *Modulation transfer function in optical and electro-optical systems*. Vol. 21. SPIE press Bellingham, WA, 2001 (cit. on p. 52).

[Bose+ 19] A Bose et al. “THz Multi-Layer Imaging via Nonlinear Inverse Scattering”. In: *2019 44th International Conference on Infrared, Millimeter, and Terahertz Waves (IRMMW-THz)*. IEEE. 2019, pp. 1–2 (cit. on p. 56).

[BBV 04] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004 (cit. on p. 13).

[Burg+ 19] Martin Burger, Janic Föcke, Lukas Nickel, Peter Jung, and Sven Augustin. "Reconstruction Methods in THz Single-pixel Imaging". In: *Compressed Sensing and Its Applications*. Springer, 2019, pp. 263–290 (cit. on p. 56).

[CGM 95] Walter G. Carrara, Ron S. Goodman, and Ronald M. Majewski. *Spotlight synthetic aperture radar: signal processing algorithms*. eng. Artech House, 1995. ISBN: 0890067287 (cit. on p. 7).

[Cham 04] JM Chamberlain. "Where optics meets electronics: recent progress in decreasing the terahertz gap". In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 362.1815 (2004), pp. 199–213 (cit. on p. 6).

[CDM 07] Wai Lam Chan, Jason Deibel, and Daniel M Mittleman. "Imaging with terahertz radiation". In: *Reports on progress in physics* 70.8 (2007), p. 1325 (cit. on pp. 1, 35).

[Chan+ 17] J-H. Chang, C-L. Li, B. Poczos, B.V.K. Vijaya Kumar, and A.C. Sankaranarayanan. "One network to solve them all — Solving Linear Inverse Problems using Deep Projection Models". In: *Proc. IEEE Int. Conf. on Computer Vision*. 2017 (cit. on p. 57).

[CKC 03] Hou-Tong Chen, Roland Kersting, and Gyu Cheon Cho. "Terahertz imaging with nanometer resolution". In: *Applied Physics Letters* 83.15 (2003), pp. 3009–3011 (cit. on p. 35).

[CHP 10] Yang Chen, Shengyang Huang, and Emma Pickwell-MacPherson. "Frequency-wavelet domain deconvolution for terahertz reflection imaging and spectroscopy". In: *Optics express* 18.2 (2010), pp. 1177–1190 (cit. on p. 35).

[Clar+ 18] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J Davison. "Learning to solve nonlinear least squares for monocular stereo". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 284–299 (cit. on p. 56).

[CL 96] Thomas F Coleman and Yuying Li. "An interior trust region approach for nonlinear minimization subject to bounds". In: *SIAM Journal on optimization* 6.2 (1996), pp. 418–445 (cit. on pp. 19, 39–41, 57, 62, 74).

[Coop+ 11] Ken B Cooper, Robert J Dengler, Nuria Llombart, Bertrand Thomas, Goutam Chattopadhyay, and Peter H Siegel. "THz imaging radar for standoff personnel screening". In: *IEEE Transactions on Terahertz Science and Technology* 1.1 (2011), pp. 169–182 (cit. on p. 35).

[Coop+ 17] Ken B Cooper, Stephen L Durden, Corey J Cochrane, Raquel Rodriguez Monje, Robert J Dengler, and Chad Baldi. "Using FMCW Doppler radar to detect targets up to the maximum unambiguous range". In: *IEEE Geoscience and Remote Sensing Letters* 14.3 (2017), pp. 339–343 (cit. on p. 8).

[Dabo+ 07] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. "Image denoising by sparse 3-D transform-domain collaborative filtering". In: *IEEE Transactions on image processing* 16.8 (2007), pp. 2080–2095 (cit. on p. 59).

[Dani 96] David J Daniels. "Surface-penetrating radar". In: *Electronics & Communication Engineering Journal* 8.4 (1996), pp. 165–182 (cit. on p. 7).

[Deng+ 09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255 (cit. on p. 24).

[Ding+ 13] Jinshan Ding, Matthias Kahl, Otmar Loffeld, and Peter Haring Bolívar. "THz 3-D image formation using SAR techniques: simulation, processing and experimental results". In: *IEEE Transactions on Terahertz Science and Technology* 3.5 (2013), pp. 606–616 (cit. on pp. 7, 26, 29, 35, 36).

[Ding+ 10] Sheng-Hui Ding, Qi Li, Rui Yao, and Qi Wang. "High-resolution terahertz reflective imaging and image restoration". In: *Applied optics* 49.36 (2010), pp. 6834–6839 (cit. on pp. 35, 50).

[Dong+ 14] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. "Learning a deep convolutional network for image super-resolution". In: *Proc. Europ. Conf. Computer Vision*. Springer. 2014, pp. 184–199 (cit. on p. 59).

[Dong+ 15] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. "Image super-resolution using deep convolutional networks". In: *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), pp. 295–307 (cit. on p. 24).

[DBM 01] Timothy D Dorney, Richard G Baraniuk, and Daniel M Mittleman. “Material parameter estimation with terahertz time-domain spectroscopy”. In: *JOSA A* 18.7 (2001), pp. 1562–1571 (cit. on pp. 56, 92).

[Dros+ 17] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. “Introducing mvtec itodd-a dataset for 3d object recognition in industry”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 2200–2208 (cit. on p. 31).

[Fors+ 83] MRB Forshaw, A Haskell, PF Miller, DJ Stanley, and JRG Townshend. “Spatial resolution of remotely sensed imagery A review paper”. In: *International Journal of Remote Sensing* 4.3 (1983), pp. 497–520 (cit. on p. 52).

[GBB 11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep sparse rectifier neural networks”. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 315–323 (cit. on pp. 25, 61).

[GBC 16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016 (cit. on pp. 16, 24).

[HGK 75] H-J Hagemann, W Gudat, and C Kunz. “Optical constants from the far infrared to the x-ray region: Mg, Al, Cu, Ag, Au, Bi, C, and Al 2 O 3”. In: *JOSA* 65.6 (1975), pp. 742–744 (cit. on p. 103).

[Hayk 10] Simon Haykin. *Neural networks and learning machines, 3/E*. Pearson Education India, 2010 (cit. on pp. 21, 23, 24).

[He+ 15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034 (cit. on pp. 75, 82).

[He+ 16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on p. 25).

[HH 19] R. Heckel and P. Hand. “Deep Decoder: Concise Image Representations from Untrained Non-convolutional Networks”. In: *Int. Conf. on Learning Representations*. 2019 (cit. on p. 57).

[HOT 06] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554 (cit. on p. 24).

[Hou+ 14] Lei Hou, Xiao Lou, Zhijin Yan, Hong Liu, and Wei Shi. "Enhancing terahertz image quality by finite impulse response digital filter". In: *Infrared, Millimeter, and Terahertz waves (IRMMW-THz), 2014 39th International Conference on*. IEEE. 2014, pp. 1–2 (cit. on p. 35).

[HN 95] Binbin B Hu and Martin C Nuss. "Imaging with terahertz waves". In: *Optics letters* 20.16 (1995), pp. 1716–1718 (cit. on pp. 1, 6, 7, 35).

[Huns+ 98] S Hunsche, M Koch, I Brener, and MC Nuss. "THz near-field imaging". In: *Optics communications* 150.1 (1998), pp. 22–26 (cit. on p. 46).

[IS 15] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *Proc. Int. Conf. on Machine Learning* (2015) (cit. on pp. 25, 61).

[Jans+ 10] Christian Jansen et al. "Terahertz imaging: applications and perspectives". In: *Appl. Opt.* 49.19 (2010), E48–E57 (cit. on p. 1).

[JDM 01] Jon L Johnson, Timothy D Dorney, and Daniel M Mittleman. "Interferometric imaging with terahertz pulses". In: *IEEE Journal of selected topics in quantum electronics* 7.4 (2001), pp. 592–599 (cit. on p. 35).

[Kahl+ 12] M. Kahl et al. "Stand-off real-time synthetic imaging at mm-wave frequencies". In: *Passive and Active Millimeter-Wave Imaging XV*. Vol. 8362. 1. 2012, p. 836208 (cit. on p. 35).

[KKM 16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. "Accurate image super-resolution using very deep convolutional networks". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2016, pp. 1646–1654 (cit. on p. 59).

[KB 14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on pp. 18, 61).

[Knob+ 02] P Knobloch et al. "Medical THz imaging: an investigation of histo-pathological samples". In: *Physics in Medicine & Biology* 47.21 (2002), p. 3875 (cit. on p. 35).

[Kobl+ 17] Erich Kobler, Teresa Klatzer, Kerstin Hammernik, and Thomas Pock. “Variational Networks: Connecting Variational Methods and Deep Learning”. In: *Proc. German Conf. Pattern Recognition (GCPR)*. 2017 (cit. on p. 57).

[LeCu 98] Yann LeCun. “The MNIST database of handwritten digits”. In: *http://yann. lecun. com/exdb/mnist/* (1998) (cit. on p. 19).

[LeCu+ 98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. on p. 25).

[Lee 09] Yun-Shik Lee. *Principles of terahertz science and technology*. Vol. 170. Springer Science & Business Media, 2009 (cit. on pp. 5, 6).

[Levi+ 11] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. “Understanding blind deconvolution algorithms”. In: *IEEE transactions on pattern analysis and machine intelligence* 33.12 (2011), pp. 2354–2367 (cit. on p. 42).

[Li+ 18] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. “Visualizing the loss landscape of neural nets”. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 2018, pp. 6391–6401 (cit. on p. 106).

[Li+ 08] Yue Li, Li Li, Andrew Hellicar, and Y Jay Guo. “Super-resolution reconstruction of terahertz images”. In: *SPIE Defense and Security Symposium*. International Society for Optics and Photonics. 2008, 69490J–69490J (cit. on pp. 35, 50).

[LHM 91] Hans J Liebe, George A Hufford, and Takeshi Manabe. “A model for the complex permittivity of water at frequencies below 1 THz”. In: *International Journal of Infrared and Millimeter Waves* 12.7 (1991), pp. 659–675 (cit. on p. 101).

[Litj+ 17] Geert Litjens et al. “A survey on deep learning in medical image analysis”. In: *Medical image analysis* 42 (2017), pp. 60–88 (cit. on p. 24).

[LN 89] Dong C Liu and Jorge Nocedal. “On the limited memory BFGS method for large scale optimization”. In: *Mathematical programming* 45.1-3 (1989), pp. 503–528 (cit. on pp. 19, 74, 76).

[LSD 15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440 (cit. on p. 71).

[LH 18] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *ICLR*. 2018 (cit. on pp. 18, 74, 76).

[Lucy 74] Leon B Lucy. "An iterative technique for the rectification of observed distributions". In: *The astronomical journal* 79 (1974), p. 745 (cit. on pp. 35, 43, 48–50).

[MHN+ 13] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. icml*. Vol. 30. 1. Citeseer. 2013, p. 3 (cit. on p. 25).

[Marq 63] Donald W Marquardt. "An algorithm for least-squares estimation of nonlinear parameters". In: *Journal of the society for Industrial and Applied Mathematics* 11.2 (1963), pp. 431–441 (cit. on p. 74).

[Math 16] MathWorks Inc. *Using Colormaps*. 2016. URL: `https://www.mathworks.com/help/matlab/examples/using-colormaps.html` (visited on 02/22/2017) (cit. on p. 43).

[Mein+ 17] T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers. "Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems". In: *Proc. IEEE Int. Conf. on Computer Vision*. 2017 (cit. on p. 57).

[MHL 07] Adriano Meta, Peter Hoogeboom, and Leo P Ligthart. "Signal processing for FMCW SAR". In: *IEEE Transactions on Geoscience and Remote Sensing* 45.11 (2007), pp. 3519–3532 (cit. on p. 7).

[Mitc 97] Tom Mitchell. "Machine learning". In: (1997) (cit. on p. 19).

[MJN 96] Daniel M Mittleman, Rune H Jacobsen, and Martin C Nuss. "T-ray imaging". In: *IEEE Journal of selected topics in quantum electronics* 2.3 (1996), pp. 679–692 (cit. on pp. 6, 7).

[MMC 19] M. Moeller, T. Möllenhoff, and D. Cremers. *Controlling Neural Networks via Energy Dissipation*. Online at `https://arxiv.org/abs/1904.03081`. 2019 (cit. on p. 57).

[MV 89] David C Munson and Robert L Visentin. "A signal processing view of strip-mapping synthetic aperture radar". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.12 (1989), pp. 2131–2147 (cit. on pp. 9, 37, 38).

[NHM 17] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. "Deep multi-scale convolutional neural network for dynamic scene deblurring". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2017, pp. 3883–3891 (cit. on p. 59).

[Nest 03] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media, 2003 (cit. on p. 16).

[NW 06] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006 (cit. on pp. 13, 18).

[Oppe 99] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999 (cit. on p. 8).

[Path+ 16] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. "Context encoders: Feature learning by inpainting". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2016, pp. 2536–2544 (cit. on p. 24).

[PF 14] Daniele Perrone and Paolo Favaro. "Total variation blind deconvolution: The devil is in the details". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 2909–2916 (cit. on p. 42).

[PR 17] Tobias Plotz and Stefan Roth. "Benchmarking denoising algorithms with real photographs". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2017, pp. 1586–1595 (cit. on p. 59).

[Poly 63] Boris T Polyak. "Gradient methods for the minimisation of functionals". In: *USSR Computational Mathematics and Mathematical Physics* 3.4 (1963), pp. 864–878 (cit. on p. 16).

[Requ+ 06] Maria E Requena-Pérez, Antonio Albero-Ortiz, Juan Monzó-Cabrera, and Alejandro Díaz-Morcillo. "Combined use of genetic algorithms and gradient descent optmization methods for accurate inverse permittivity measurement". In: *IEEE Transactions on Microwave Theory and Techniques* 54.2 (2006), pp. 615–624 (cit. on pp. 56, 92).

[Rich 72] William Hadley Richardson. "Bayesian-based iterative method of image restoration". In: *JOSA* 62.1 (1972), pp. 55–59 (cit. on pp. 35, 43, 48–50).

[Ripl 07] Brian D Ripley. *Pattern recognition and neural networks*. Cambridge university press, 2007 (cit. on p. 24).

[RFB 15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241 (cit. on pp. 25, 26, 68, 71).

[Rose 58] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386 (cit. on p. 21).

[Samu 59] Arthur L Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229 (cit. on p. 19).

[Schr+ 19] Nina S Schreiner, Wolfgang Sauer-Greff, Ralph Urbansky, Georg von Freymann, and Fabian Friederich. "Multilayer thickness measurements below the Rayleigh limit using FMCW millimeter and terahertz waves". In: *Sensors* 19.18 (2019), p. 3910 (cit. on p. 92).

[Schu+ 16] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. "Learning to deblur". In: *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)* 38.7 (2016), pp. 1439–1451 (cit. on p. 59).

[SB 14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014 (cit. on p. 17).

[Sieg 02] Peter H Siegel. "Terahertz technology". In: *IEEE Transactions on microwave theory and techniques* 50.3 (2002), pp. 910–928 (cit. on pp. 1, 5, 6).

[Smit 66] Warren J Smith. *Modern optical engineering*. Tata McGraw-Hill Education, 1966 (cit. on p. 52).

[Stan 59] Military Standard. *Photographic lenses*. 1959. URL: `http://www.dtic.mil/dtic/tr/fulltext/u2/a345623.pdf` (visited on 06/03/2011) (cit. on pp. 29, 30).

[Stoc+ 19] Daniel Stock, Matthias Kahl, Anna K Wigger, Tak Ming Wong, Andreas Kolb, and Peter Haring Bolívar. "Advanced signal processing techniques for THz imaging and sensing enhancement in material quality control applications". In: *Terahertz, RF, Millimeter, and Submillimeter-Wave Technology and Applications XII*. Vol. 10917. International Society for Optics and Photonics. 2019, 109170Z (cit. on p. 3).

[Stov 92] Andrew G Stove. "Linear FMCW radar techniques". In: *IEE Proceedings F-Radar and Signal Processing*. Vol. 139. 5. IET. 1992, pp. 343–350 (cit. on p. 7).

[Su+ 18] Shuochen Su, Felix Heide, Gordon Wetzstein, and Wolfgang Heidrich. "Deep end-to-end time-of-flight imaging". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6383–6392 (cit. on p. 24).

[Taka+ 09] Jun Takayanagi et al. "High-resolution time-of-flight terahertz tomography using a femtosecond fiber laser". In: *Optics express* 17.9 (2009), pp. 7533–7539 (cit. on p. 35).

[Tewa+ 17] Ayush Tewari et al. "MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction". In: *Proc. IEEE Int. Conf. on Computer Vision*. 2017 (cit. on p. 57).

[Tono 07] Masayoshi Tonouchi. "Cutting-edge terahertz technology". In: *Nature photonics* 1.2 (2007), pp. 97–105 (cit. on p. 6).

[UVL 18a] D. Ulyanov, A. Vedaldi, and V.S. Lempitsky. "Deep Image Prior". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2018 (cit. on p. 57).

[UVL 18b] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Deep image prior". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9446–9454 (cit. on pp. 26, 67, 68, 78).

[Walk+ 12] Gillian C Walker et al. "Terahertz deconvolution". In: *Optics express* 20.25 (2012), pp. 27230–27241 (cit. on p. 35).

[Wong+ 22a] Tak Ming Wong, Hartmut Bauermeister, Matthias Kahl, Peter Haring Bolívar, Michael Möller, and Andreas Kolb. "Deep Neural Network as an Optimizer for FMCW THz Image Deblurring". In: *ATHENA Research Book* 1 (2022), pp. 13–21 (cit. on p. 4).

[Wong+ 22b] Tak Ming Wong, Hartmut Bauermeister, Matthias Kahl, Peter Haring Bolívar, Michael Möller, and Andreas Kolb. "Deep Optimization Prior for THz Model Parameter Estimation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 3811–3820 (cit. on p. 4).

[Wong+ 19a] Tak Ming Wong, Matthias Kahl, Peter Haring Bolívar, and Andreas Kolb. "Computational Image Enhancement for Frequency Modulated Continuous Wave (FMCW) THz Image". In: *Journal of Infrared, Millimeter, and Terahertz Waves* 40.7 (2019), pp. 775–800 (cit. on p. 3).

[Wong+ 19b] Tak Ming Wong, Matthias Kahl, Peter Haring Bolívar, Andreas Kolb, and Michael Möller. "Training Auto-Encoder-Based Optimizers for Terahertz Image Reconstruction". In: *German Conference on Pattern Recognition*. Springer. 2019, pp. 93–106 (cit. on p. 3).

[XJ 10] Li Xu and Jiaya Jia. "Two-phase kernel estimation for robust motion deblurring". In: *European conference on computer vision*. Springer. 2010, pp. 157–170 (cit. on pp. 42, 43, 49, 50, 87, 89).

[Xu+ 14] Li Xu, Jimmy S Ren, Ce Liu, and Jiaya Jia. "Deep convolutional neural network for image deconvolution". In: *Advances in neural information processing systems* 27 (2014), pp. 1790–1798 (cit. on p. 24).

[XZJ 13] Li Xu, Shicheng Zheng, and Jiaya Jia. "Unnatural lo sparse representation for natural image deblurring". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1107–1114 (cit. on pp. 42, 43, 49, 50, 87, 89).

[XFL 14] Li-Min Xu, Wen-Hui Fan, and Jia Liu. "High-resolution reconstruction for terahertz imaging". In: *Applied optics* 53.33 (2014), pp. 7891–7897 (cit. on pp. 35, 50).

[Zhan+ 17] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising". In: *IEEE transactions on image processing* 26.7 (2017), pp. 3142–3155 (cit. on p. 24).