Detection for Through-Wall Radar Imaging based on low rank and sparse decomposition models

Hugo Brehier

université PARIS-SACLAY

Detection for Through-Wall Radar Imaging based on low rank plus sparse decompositions

Détection pour l'Imagerie Radar à Travers Murs par décompositions de rang faible et parcimonieuse

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 : Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Traitement du Signal et des Images
Graduate School : Informatique et Sciences du Numérique
Référent : CentraleSupélec

Thèse préparée dans l'unité de recherche SONDRA (CentraleSupélec, Université Paris-Saclay) sous la direction de **Guillaume GINOLHAC**, Professeur des Universités, le co-encadrement de **Chengfang REN**, maître de conférences, de **Arnaud BRELOY**, Professeur des Universités, et de **Israel HINOSTROZA**, ingénieur de recherche.

Thèse soutenue à Paris-Saclay, le 4 décembre 2024, par

Hugo BREHIER

Composition du jury

Membres du jury avec voix délibérative

Nelly PUSTELNIK Directrice de recherche, CNRS (ENS Lyon)	Présidente & Examinatrice
Michael MUMA Professeur, TU Darmstadt	Rapporteur
Jean-Yves TOURNERET Professeur des Universités, INP de Toulouse	Rapporteur
Muriel DARCES Professeure des Universités, Sorbonne Université	Examinatrice
Jeremy COHEN Chargé de recherche, CNRS (Univ. Lyon I - INSA Lyon)	Examinateur

THESE DE DOCTORAT
NNT : 2024UPASG070

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Détection pour l'Imagerie Radar à Travers Murs par décompositions de rang faible et parcimonieuse
Mots clés: radar à travers murs, detection, problème inverse, decomposition matricielle, optimisation, apprentissage profond

Résumé: L'imagerie radar à travers murs est un domaine de recherche visant à imager des pièces cachées à l'œil nu par un mur. Cela présente des défis dus notamment à la distorsion du signal causée par le mur ainsi que par la scène à imager. À cela s'ajoute un bruit ambiant qui complique la détection des signaux faibles provenant des cibles. Les travaux entrepris dans cette thèse se concentrent sur la détection et la localisation de cibles stationnaires dans un scénario en deux dimensions spatiales.

Nous introduisons des méthodes d'imagerie basées sur la reconstruction jointe des éléments constituant la scène, à savoir le mur et les cibles cachées. Nous utilisons une décomposition en rang faible et parcimonieux via une extension de RPCA. Nous étudions ensuite son extension à des bruits hétérogènes via une distance robuste dite de Huber. Nous étudions également son extension non-convexe sur la variété des matrices de rang fixe. Finalement, nous abordons la transition vers une méthode basée sur les données, en utilisant une méthode hybride dite de réseau déroulé basé sur un gradient proximal.

Les résultats montrent que les méthodes proposées surpassent les approches classiques en simulations. Toutefois, des défis persistent, notamment dans la prise en compte des effets physiques complexes sur le signal. Nous soulignons le potentiel de ces méthodes pour des applications plus larges, comme les radars à pénétration de sol et l'imagerie computationnelle.

Title: Detection for Through-Wall Radar Imaging based on low rank plus sparse decompositions
Keywords: through wall radar imaging, inverse problem, matrix decomposition, optimization, deep learning

Abstract: Through Wall Radar Imaging is a field of research aimed at imaging rooms hidden from the naked eye. This presents challenges, notably due to the signal's attenuation and distortion caused by the wall and the scene elements. Additionally, ambient noise complicates the detection of weak signals coming from the targets. The work in this thesis focuses on the detection and localization of stationary targets in a two-dimensional spatial scenario.

We introduce imaging methods based on the joint reconstruction of the elements constituting the scene, namely the wall and the hidden targets, by decomposition into low-rank and sparse components (via an extension of RPCA). We then study its extension to heterogeneous noise via robust distances, such as Huber's. We delve into optimization techniques on Riemannian manifolds using the one of fixed rank matrices. Finally, we address the transition to a data-driven method using a hybrid method known as unrolled networks, specifically a proximal gradient unrolling.

The results show that the proposed methods outperform classical approaches in simulations. However, challenges remain, particularly in accounting for the complex physical effects on the signal. We highlight the potential of these methods for broader applications, such as Ground Penetrating Radar and computational imaging.

Acknowledgements

Before switching to french, I would like to thank every person present at SONDRA during the three years and few months that I spent there. I had the opportunity of meeting many great people, both as researchers and as individuals.

Premièrement, je voudrais remercier mon équipe d'encadrement, grâce à laquelle cette thèse s'est déroulée sans encombres. Tout d'abord, merci Guillaume pour ta direction claire et ta présence tout au long de ces années. Tu as su me guider tout en me laissant explorer ce qui m'intriguait. J'ai donc eu le sentiment de pouvoir m'approprier le sujet au fil des mois. Ensuite, merci à Chengfang, que j'ai côtoyé au quotidien à SONDRA. Les discussions que nous avons eu ont toujours été fructueuses et m'ont permis d'avancer dans la thèse, grâce a tes connaissances précises des différents domaines liés au sujet. Merci à Arnaud, avec qui j'ai commencé dans la recherche. J'apprécie vraiment ton foisonnement d'idées toujours pertinentes, qui aident à aller de l'avant. Finalement, merci Israel pour ton expertise qui m'a aidé à éclaircir des sujets quelques fois obscurs pour moi. De plus, je tiens à vous remercier ensemble. Durant ces trois ans, j'ai ressenti beaucoup de respect et de confiance, ce qui m'a donné l'envie de donner de ma personne pour ces travaux de recherche.

Secondement, je tiens à remercier Stéphane et Virginie (ainsi qu'Isabelle) pour la gestion bienveillance du laboratoire SONDRA. En poussant le laboratoire à proposer des workshops, des sessions spéciales, en nous donnant l'occasion de publier sans grande restriction et de déplacer, nous avons pu découvrir la communauté scientifique nous englobant. Et donc, merci à tous les stagiaires (Alexandre, Louis, Axel, encore Louis), doctorants (Alexis, Quentin, Yanisse, Max, Steve, Hugo, Huy, Pierre, Thomas, Florent, Cyprien, Agustin, Dihia, Nathan) post-doctorants (Harsha) et chercheurs permanents (Mohammed, Jean-Phi, Laeticia, Regis) de SONDRA qui ont fait et font la vie du laboratoire. J'y associe les personnes rencontrées durant mes visites à Annecy au LISTIC ou en workshop, tel que le SLSIP.

Finalement, je remercie mes parents, ma famille et amis, m'ayant entre autres donné le goût de la curiosité, du savoir, et m'ayant doucement mené à envisager d'effectuer ce doctorat.

Contents

List of Figures

List of Tables

Notations

a scalar

$\mathbf{a}$ vector

$\mathbf{A}$ matrix

$\mathbf{A}^*$ conjugate operator

$\mathbf{A}^H$ conjugate transpose operator

$\mathrm{vec}(\mathbf{A})$ vectorisation operator

$\mathrm{vec}^{-1}(\mathbf{A})$ unvectorisation operator (such that $\mathrm{vec}^{-1}(\mathrm{vec}(\mathbf{A})) = \mathbf{A}$)

$\mathrm{diag}(\mathbf{A})$ diagonal operator

$\mathrm{tr}(\mathbf{A})$ trace operator

$\mathrm{rk}(\mathbf{A})$ rank operator

$\|\mathbf{A}\|_F$ Frobenius norm

$\|\mathbf{A}\|_*$ Nuclear norm

$\|\mathbf{A}\|_{p,q}$ $\ell_{p,q}$ norm, $\forall (p,q) \in \mathbb{N} \times \mathbb{N}$

$\langle \mathbf{A}, \mathbf{B} \rangle$ (standard) inner product

$\mathbf{A} \otimes \mathbf{B}$ Kronecker product

$\mathbf{A} \odot \mathbf{B}$ Hadamard product

$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N]$ construction of a matrix by stacking N vectors

$S_\lambda(\mathbf{A})$ Soft-thresholding of threshold λ i.e. proximal of ℓ_1 norm

$T_\lambda(\mathbf{A})$ Row-wise thresholding of threshold λ i.e. proximal of $\ell_{2,1}$ norm

$D_\mu(\mathbf{A})$ Singular Value Soft-thresholding of threshold μ i.e. proximal of nuclear norm

Introduction

Through the Wall Radar Imaging (TWRI) is a topic of research [Amin, 2017] that aims at detecting targets in an enclosed scene from its outside via radar measurements, the scene being unobservable to the naked eye. TWRI is grounded in the principles of radar aka Radio Detection and Ranging. As its name suggests, the system functions by emitting electromagnetic waves capable of penetrating walls. Upon encountering objects or persons, these waves are reflected back to the radar system. By analyzing the round-trip delay time of these waves and the properties of these reflections, the system can reconstruct an image of the occluded area. The essential components of TWRI include a transmitter that generates electromagnetic waves capable of penetrating walls, a receiver that captures the reflected signals, and a signal processing unit that interprets these signals to produce images or detect movements. Different types of TWRI systems exist, each with unique operational principles. Pulsed radar systems emit brief bursts of radio waves, using the time delay of echoes to gauge the distance to objects. Continuous Wave (CW) radar systems maintain a continuous transmission, leveraging frequency shifts (i.e. using the Doppler shift) in the reflected waves to detect motion. Ultra Wideband (UWB) radar systems use a broad frequency spectrum, which enhances the resolution. The applications of TWRI are varied. In search and rescue missions, it facilitates the location of survivors amidst rubble following building collapses. In law enforcement, it supports surveillance and strategic planning in critical situations such as hostage rescues or standoffs. Military operations benefit from its ability to enhance situational awareness in urban combat by detecting hidden adversaries. In the realm of structural health monitoring, TWRI is used to assess the integrity of buildings, identifying concealed defects. It may also be used in health monitoring [Li et al., 2021b, Yang et al., 2021].

However, TWRI also encounters several challenges and limitations. Signal attenuation is a primordial concern, as the strength of the radar signal diminishes when passing through dense or thick materials, therefore challenging the imaging process. Clutter and noise from multiple surface reflections can generate false positives and complicate image interpretation. Moreover, the ability of TWRI to see through walls raises significant privacy issues, necessitating strict regulation and ethical oversight. Future developments in TWRI are focused on advancements in materials, signal processing algorithms, and computational power, which contribute to improving the capabilities of these systems. Research aims to enhance resolution, reduce noise, and develop more portable and user-friendly systems. Integrating TWRI with artificial intelligence and machine learning holds promise for more accurate imaging.

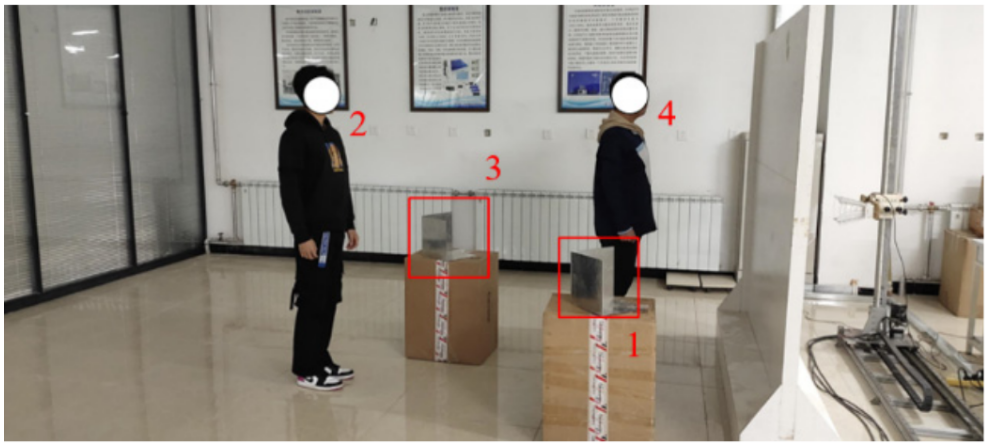


Figure 1: Typical TWRI scenario [Qu et al., 2022] with the radar on the right of the wall

On the processing side (as opposed to the acquisition), past works have focused on several aspects of TWRI: localization of targets, change detection, movement characterization [Debes et al., 2011, Clemente et al., 2013, Gennarelli et al., 2015, Li et al., 2019]. Here, we focus on the detection and localization of stationary targets, which can be readily extended to moving targets by collecting measurements over time and applying the same methodology.

We will focus on a 2D scenario that necessitates the use of multiple antennas (or a single traveling one) to achieve a sufficient resolution. A standard hypothesis in TWRI is for the wall to be homogeneous, with permittivity and thickness considered to be known, or to be estimated in a previous step [Protiva et al., 2011, Jin et al., 2013]. Other works have developed methods for the unknown case based on focusing techniques [Wang and Amin, 2006, Ahmad et al., 2007]. In an earlier phase of TWRI, some methods [Ahmad, 2008, Dehmollaian and Sarabandi, 2008] were developed that use Synthetic Aperture Radar (SAR) techniques [Soumekh, 1999] such as Back-Projection (BP). Those methods require the acquisition of measurements from an empty scene to remove the front wall. Subsequently, two-step techniques were developed [Amin and Ahmad, 2013] which consist in: a) filtering the front wall echoes based on subspace decomposition [Verma et al., 2009, Tivive et al., 2011, Tivive et al., 2015] b) recovering the target positions, based on the hypothesis of the targets sparsity w.r.t. the scene dimensions, with the possible use of Compressive Sensing (CS) methods to reduce computation times [Huang et al., 2010]. This approach requires the use of a dictionary to map the returns onto a grid covering the scene. This formalism also handles multipaths or front wall reflections more precisely [Leigsnering et al., 2014]. Building on this, we will work on one-step methods that have been explored during the past years via the framework of Robust Principal Component Analysis (RPCA)

[Candès et al., 2011, Chandrasekaran et al., 2011, Mardani et al., 2013]. RPCA decomposes a matrix in two separate components: one being low rank and the other being sparse, the two parts capturing respectively the returns of the front wall and the returns of the targets. Such one-step methods have been shown to perform better than their counterparts in several radar experiments [Tang et al., 2016, Tang et al., 2020, Breloy et al., 2018, Mériaux et al., 2019].

The manuscript is organized as follows. Chapter 1 serves as an introduction to the topic of detection and localization in TWRI. Basic notions of radar and classical techniques for TWRI are presented. Chapter 2 presents basic and advanced optimization methods used throughout the manuscript. Chapter 3 introduces our first work about a refined low rank and sparse decomposition tailored for our setup. Then, Chapter 4 presents a robust extension to the method against heterogeneous noise, with an extension to Riemannian optimization to handle a non-convex form. Finally, Chapter 5 connects these model-based approaches to data-driven ones via hybrid methods called unrolling networks to gain a number of advantages: higher detection performance and lower runtime.

The work carried out during this thesis and the results obtained have led to the following international publications:

- Brehier, H., Breloy, A., Ren, C., Hinostroza, I., and Ginolhac, G. (2022a). Robust PCA for through-the-wall radar imaging. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 2246–2250

- Brehier, H., Breloy, A., Ren, C., and Ginolhac, G. (2023b). Through the wall radar imaging via Kronecker-structured Huber-type RPCA. *Signal Processing*, page 109228

- Brehier, H., Breloy, A., Ren, C., and Ginolhac, G. (2024b). Through-the-wall radar imaging with wall clutter removal via riemannian optimization on the fixed-rank manifold. In *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8596–8600

- Brehier, H., Breloy, A., Ren, C., and Ginolhac, G. (2024a). Deep unrolling of Robust PCA and convolutional sparse coding for stationary target localization in through wall radar imaging. In *2024 32th European Signal Processing Conference (EUSIPCO)*

And the following national (french) publications:

- Brehier, H., Breloy, A., Ren, C., Hinostroza, I., and Ginolhac, G. (2022b). Robust PCA pour l'imagerie radar à travers les murs. In *XXVIIIème Colloque Francophone de Traitement du Signal et des Images, GRETSI*

• Brehier, H., Breloy, A., Ren, C., and Ginolhac, G. (2023a). Atténuation robuste du fouillis mural en imagerie radar à travers murs par optimisation riemannienne. In *XXIXème Colloque Francophone de Traitement du Signal et des Images, GRETSI*

1 - Background on Through Wall Radar Imaging

Contents

This first chapter serves as an introduction to the topic of imaging and detection through walls using a radar system. It comprises an overview of the effects the wall has on the radar signal as well as the system used for the signal acquisition. Finally, classical signal processing techniques are described as background for our methods.

1.1 . Preliminary Concepts

1.1.1 . Wall effects: attenuation and dispersion

Electromagnetic properties of walls and building materials are crucial to study and model the effects of walls on signal delay time, amplitude, and pulse shape. When electromagnetic waves traverse a medium, they experience distortion in amplitude and phase. These distortions can be attributed to the dispersive and attenuative properties of the medium through which the waves propagate.

Wall compositions are, in general, dielectric and nonmagnetic in nature. Thus, they exhibit no response to magnetic fields. when exposed to electric fields, numerous electric dipoles are generated within their molecular structures. These dipoles tend to align with the external electric field. The collective effect of the localized shifts between bound positive and negative charges is

called polarization. This polarization corresponds to a state of stress within the material, leading to potential energy storage, which is released when the external electric field is removed. A material's ability to be polarized by external fields is determined by its molecular structure. Within the wall material, polarization increases the density of electric field lines. The ratio of the number of field lines inside the material to that in free space (in the absence of material) is known as the dielectric constant, or relative permittivity, of the material and denoted ϵ_r. The real part ϵ_r' of the relative permittivity represents the material's ability to be polarized in response to an applied electric field, thereby storing energy. Its imaginary part ϵ_r'' represents the energy loss in the material, which is associated with the absorption and dissipation of energy, e.g. in the form of heat. The loss tangent $\tan\delta \triangleq \frac{\epsilon_r''}{\epsilon_r'}$, is a measure of the dielectric losses in a material relative to its ability to store energy. It is defined as the ratio of the imaginary part to the real part of the complex permittivity. Then:

$$\epsilon_r = \epsilon_r' - j\epsilon_r'' = \epsilon_r'(1 - j\tan\delta) \tag{1.1}$$

The loss tangent provides a way to quantify the effect of loss on the electromagnetic field within a material. Estimations of the dielectric constant and loss tangent of some materials are shown in Figure 1.1 on the frequency band of $1-3$ GHz.

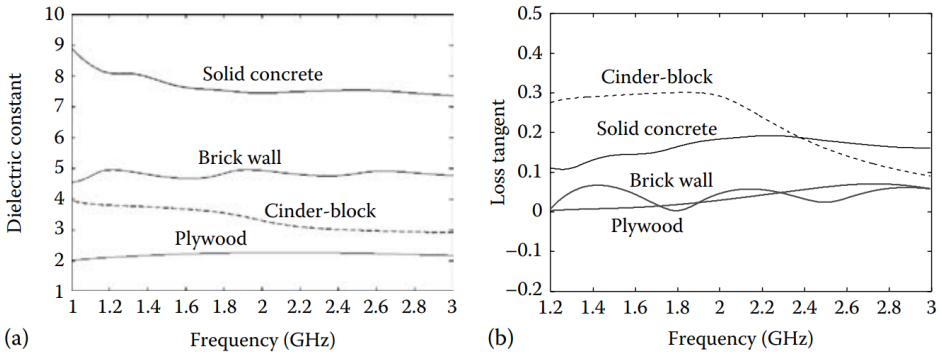
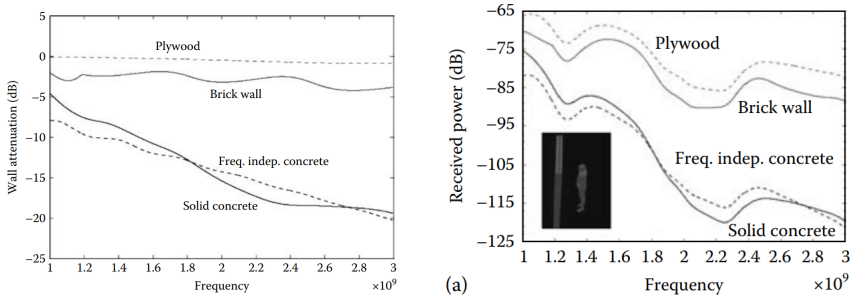


Figure 1.1: Estimation of permittivity [Amin, 2017]

In addition to the materials composing the wall, factors such as the wall's shape and thickness significantly influence propagation effects. The dielectric constants of the wall, along with its thickness, introduce varying delays in the propagation path. When high accuracy is required, the travel time through the thickness of objects along the signal path becomes critical for precise delay measurement. Multiple reflections within the wall further complicate these effects. A significant challenge also occurs when dealing with thick walls or highly lossy materials, such as reinforced concrete which exhibit substantial transmission loss due to reflection and absorption. The attenuation of a signal traversing a wall is generally caused by conductivity loss, reflection loss

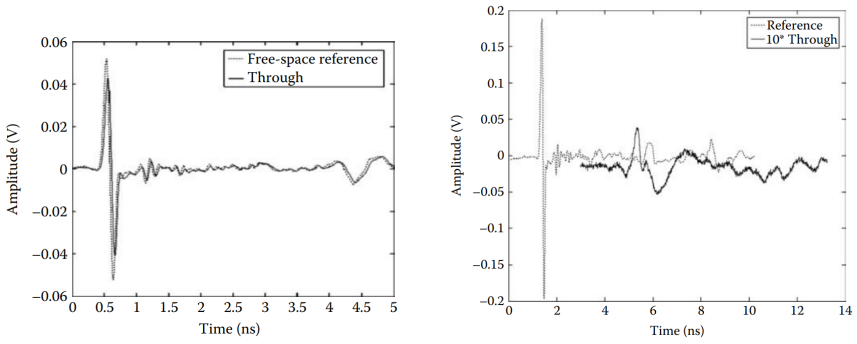
and multiple internal reflections within the wall. It is shown in Figure 1.2a. Assuming antennas and target in the far-field regions of the wall, a modified radar equation with wall and target losses for the ratio of the received to the transmitted power, gives the results in Figure 1.2b.



(a) Estimation of wall attenuation (b) Estimation of received power from a human

Figure 1.2: Attenuation through wall [Amin, 2017]

The frequency-dependent properties of materials comprising the propagation medium also impact the signal. Across a broad frequency spectrum, materials display varying behaviors when interacting with electromagnetic waves. As the frequency of the electromagnetic field increases, the molecular dipoles within the material are unable to respond instantaneously. This delayed response of the material to the electromagnetic waves leads to the phase velocity of waves to depend on its frequency, a phenomenon called dispersion. The most significant consequences of these effects are pulse broadening, amplitude loss, and overall signal distortion as shown in Figure 1.3.



(a) Signal through wood (b) Signal through reinforced concrete (through signal scaled up by a factor 10)

Figure 1.3: Measurement through (one way travel) different walls [Amin, 2017]

The presence of noise and the limited dynamic range of measurement

equipment hinder the accurate measurement of weak signal levels under these conditions. Noise can be heterogeneous i.e. varying across the radar's field of view. The statistical properties of the noise differ from one region to another, leading to a non-uniform noise environment. In a heterogeneous noise environment, the clutter can change significantly across different areas. For example, clutter might be stronger in some areas and weaker in others. This variability makes it more challenging to detect targets, as the radar system cannot assume a consistent noise level and must adapt its processing to different noise conditions.

In the first chapters on optimization techniques, we will set aside the dispersion problematic. We will consider better detection performance in the context of the wall heavy attenuation and heterogeneous noise. We will study dispersion in the last chapter on data driven techniques.

1.1.2 . System considerations

TWRI first necessitates a wavelength that is not too small compared to the wall depth (15cm - 23cm outer wall thickness and ~ 10cm inner wall thickness), in order to penetrate the wall without too much loss (see Figure 1.2b), as higher frequencies get more highly attenuated, while still being small enough to reflect off the target of interest. The radar systems used are typically Ultra-Wideband (UWB), meaning that their bandwidth is more than 20% of the center frequency i.e. more than 1 GHz of bandwidth for a center frequency of 2 GHz. This allows better resolution and thus more accurate imaging. Indeed, the range resolution is given by [Amin, 2017, Section 2.2.2]:

$$R_r = \frac{c}{2B} \tag{1.2}$$

where c is the speed of light in vacuum and B the signal bandwidth. Thus, the larger the bandwidth the better the resolution. The range resolution is 15cm at 1GHz of bandwidth and 3cm at 5Ghz of bandwidth. Two types of UWB waveforms exist:

- time-domain methods: the signal is a short impulse (in the order of the nanosecond) which can take a Gaussian form or its first or second derivatives. For example, the Ricker (or Mexican Hat) waveform is the negative, normalized second derivative of a Gaussian waveform (see Figure 1.4). The range profile is constructed from the matched filtered returned signal.

- frequency-domain methods: measurements are conducted at various frequencies using a Vector Network Analyser. The primary advantage of the frequency-domain approach compared to the time-domain method is its larger dynamic range. The signal produced is called a stepped frequency signal, shown on Figure 1.5. The range profile is then obtained by Fourier transform.

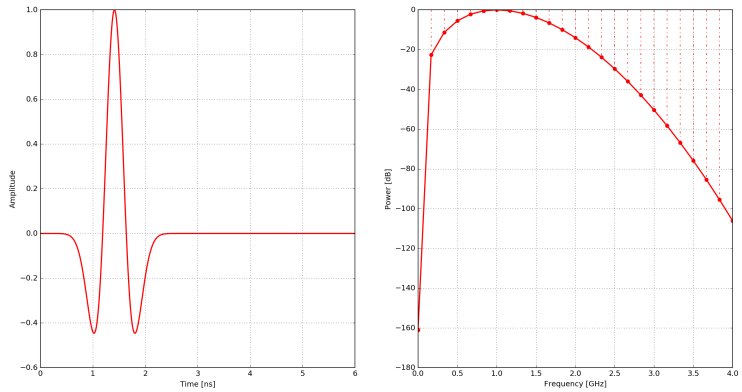


Figure 1.4: Ricker waveform (GprMax [Warren et al., 2016] documentation)

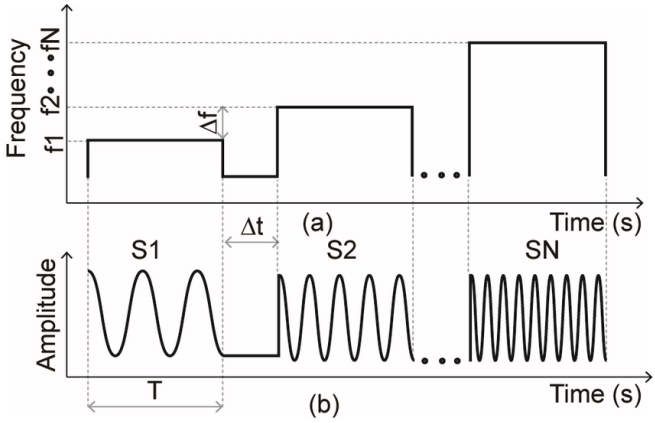


Figure 1.5: Stepped frequency signal [Kebe et al., 2020]

Using multiple transceivers, the azimuth resolution becomes a function of the radar antenna aperture and the distance to the target. For a linear array of antennas, we have the following formula [Amin, 2017, Section 2.2.2]:

$$R_c = \frac{\lambda}{D} R \tag{1.3}$$

where R is the distance between target and antenna while λ is the wavelength sent and D the length of the array (with the ratio $\frac{\lambda}{D}$ being the angular resolution) . Thus, for a typical TWRI setup where the distance to target is slightly higher than the array length, it will be in the order of the wavelength or so, implying a cross resolution of 10 cm at 3 GHz . The number of positions to consider is then ruled by Shannon's sampling theorem. Indeed, if d is the spacing between array elements, then the spatial sampling must be greater than twice the maximum spatial frequency:

$$\frac{1}{d} > 2\frac{1}{\lambda_{\text{min}}} \implies d < \frac{\lambda_{\text{min}}}{2} \tag{1.4}$$

For a minimum wavelength of 10cm corresponding to 3 GHz, the needed spacing between elements is below 5cm. Thus, a higher frequency improves the azimuth resolution but requires more sampling points (antenna elements) while also being more attenuated through the wall. Moreover, for a stepped-frequency radar, the unambiguous maximum range is:

$$R_{max} = \frac{c}{2\Delta f} \tag{1.5}$$

where Δf is the frequency step. Setting the range to 10 meters gives a frequency step of 15 MHz. This illustrates the typical characteristics of the radar system to be used for TWRI: bandwidth, frequency range, frequency step, etc. Particularly, the cross resolution implies an array length (radar aperture) as large as possible while the sampling theorem implies using a large number of array elements (spatial sampling points). This is unpractical for real arrays, while this can be alleviated using synthetic apertures.

1.1.3 . SAR imaging in free-space

Signal model

The signal returns can be collected from a single moving antenna, a type of acquisition called Synthetic Aperture Radar (SAR) [Soumekh, 1999]. This has become a staple of Remote Sensing with airborne and satellite-mounted radars.

Let the SAR move along the azimuthal axis u. The SAR emits a signal $e(t)$ of central frequency f_0 and bandwidth B at all positions u_k, $k \in [1, n]$, situated on the u axis. Each position is separated from the previous by a distance δ_u. We consider a monostatic configuration, meaning that the emitter and receptor are at the same position. After emitting the signal $e(t)$, the radar receives a signal $z_k(t)$ at the position u_k. We use the so-called "stop and go" assumption which considers that there is neither radar nor target displacement between emission and reception of a signal. We also consider the radar antenna direction is fixed across all positions u_k i.e. in "stripmap" mode. Finally, we reduce ourselves to a 2D configuration, with the radar and targets situated at null height, as shown in Figure 1.6.

Let $y_k(t)$ be the received signal at u_k reflected by a point target situated at position (x, y). Let $\sigma(x, y)$ be its reflectivity coefficient, assumed to be invariant to incidence angle (isotropic targets). So:

$$y_k(t) = \sigma(x, y)e(t - \tau_k(x, y)) \tag{1.6}$$

where $\tau_k(x, y)$ is the round-trip delay time for the signal to go from the radar at $(0, u_k)$ to the target at (x, y). In free-space (this will change for TWRI), we have:

$$\tau_k(x, y) = \frac{2\sqrt{x^2 + (y - u_k)^2}}{c} \tag{1.7}$$

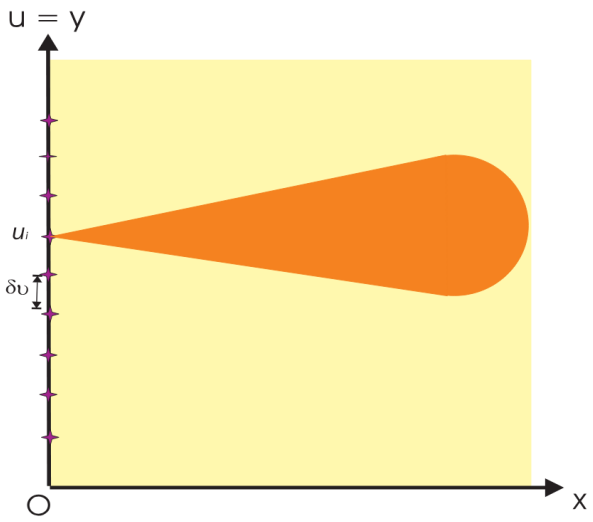


Figure 1.6: 2D stripmap SAR in free-space [Durand, 2007]

Using $\hat{e}(t)$ and $\hat{y}_k(t)$ to denote the baseband signals, we have:

$$\hat{y}_k(t) = \sigma(x,y)\hat{e}(t - \tau_k(x,y)) \exp(-j\omega_0\tau_k(x,y)) \tag{1.8}$$

where $\omega_0 = 2\pi f_0$ is the angular frequency. Let us use the baseband signal and state $e(t) = \hat{e}(t)$ and $y_k(t) = \hat{y}_k(t)$. The signal received by the radar contains the reflected signals of all P targets:

$$z_k(t) = \sum_{p=1}^{P} \sigma_p(x_p,y_p)e(t - \tau_k(x_p,y_p)) \exp(-j\omega_0\tau_k(x_p,y_p)) \tag{1.9}$$

Back-Projection algorithm

A classical SAR algorithm for image formation is the Back-Projection (BP) algorithm. Primarily, it is based on summing coherently the returned signals from all antenna positions. The signal model is:

$$m_k(t,x,y) = e(t - \tau_k(x,y)) \exp(-j\omega_0\tau_k(x,y)) \tag{1.10}$$

The intensity of a pixel at coordinates (x,y) can be expressed :

$$\begin{aligned} I(x,y) &= \sum_{k=1}^{n} \int_t z_k(t) m_k^*(t,x,y) dt \\ &= \sum_{k=1}^{n} \left(\int_t z_k(t) e^*(t - \tau_k(x,y)) dt \right) \exp(j\omega_0\tau_k(x,y)) \\ &= \sum_{k=1}^{n} p_k(\tau_k(x,y)) \exp(j\omega_0\tau_k(x,y)) \end{aligned} \tag{1.11}$$

where:

$$p_k(t) = \int_x z_k(x) e^*(x-t) dx = z_k(t) \star e(t) = z_k(t) * e^*(-t) \tag{1.12}$$

with $\star$ the cross-correlation operator and $*$ the convolution one. It is the output of a matched filter of input $z_k(t)$ by impulse response $e(t)$. The whole BP method can be viewed as the application of the radar system point spread function (PSF), i.e. the response of the radar imaging system to a point target, over the whole radar returns [Amin, 2017, Section 3.5.1]. It may also be derived in a physics approach using the Green function for the wave equation which yields a method called Kirchoff migration [Garnier and Papanicolaou, 2016, Section 4.1.7]. Using a stepped-frequency radar system, we acquire at each position u_k a signal Z_k in the frequency domain, with M equispaced frequencies over a bandwidth B. By the convolution theorem, the matched filtering can be executed by pointwise multiplication of the spectrums of the signals. Denote the Fourier Transform of e as $\mathcal{F}\{e\}$. Denote respectively P_k, Z_k and E the frequency domain representation of p_k, z_k and e. Then:

$$\begin{aligned} P_k(f) &= \mathcal{F}\{z_k\}(f) \cdot \overline{\mathcal{F}\{e\}}(f) \\ &= Z_k(f) \cdot E^*(f) \\ \implies p_k(t) &= \mathcal{F}^{-1}\{Z_k \cdot E^*\}(t) \end{aligned} \tag{1.13}$$

where $\mathcal{F}^{-1}$ is the Inverse Fourier Transform. Numerically we make use of the Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) algorithms, which output a length M frequency-domain vector from a length M time-domain vector (and vice-versa).

$$\mathbf{p}_k = \text{IFFT}\left(\mathbf{Z}_k \odot \mathbf{E}^*\right) \tag{1.14}$$

where $\odot$ is the elementwise (Hadamard) product. Retrieval of the delay $\tau_k(x,y)$ leads to a position index $i_k(x,y)$ over which to sum coherently. It is defined by:

$$i_k(x,y) = \left\lfloor \frac{\tau_k(x,y)}{\Delta t} \right\rceil \tag{1.15}$$

where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer and $\Delta t \approx \frac{1}{B}$ is the time step after IFFT. To be applied to TWRI, the BP method must be used on returns with wall returns removed, which would otherwise obscure the scene behind. This motivates us to present the most classical way to achieve this wall returns removal. Moreover, the BP must be computed with adjusted propagation delays that account for the wall, not simply using the free space described in (1.7), as it changes its path and velocity. This causes false localization of targets if not taken into account. We thus need to compensate for this by computing the delay change through the wall.

1.2 . Classical TWRI

1.2.1 . Wall returns mitigation

Let us consider a stepped-frequency radar operating as a SAR in stripmap mode. We organize the collected signal returns in a matrix $\mathbf{Y} \in \mathbb{C}^{M \times N}$ with M the number of frequencies and N the number of SAR positions:

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \tag{1.16}$$

so that $\mathbf{y}_i$ is the signal return of the i^{th} SAR position. Contrary to free-space radar SAR imaging, we need to remove the clutter from the wall before forming an image. A classical wall mitigation framework is to decompose $\mathbf{Y}$ by separating subspaces of the wall and targets. Indeed, assuming the signal acquisitions have been made parallel to the wall, the wall returns are expected to be approximately invariant across acquisitions. Since wall returns are stronger compared to target returns, they lie in the subspace spanned by the first singular vectors associated with the largest singular values. The simplest way to proceed would be to choose an arbitrary number of singular values to represent the wall subspace, such as the first one only. For completeness, we can also mention the method of [Tivive et al., 2015] as an informed way to obtain the cutoff in the range of singular values. The singular value decomposition of $\mathbf{Y}$ is written:

$$\mathbf{Y} \overset{\text{SVD}}{=} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H, \tag{1.17}$$

where $\mathbf{U} \in \mathbb{C}^{M \times M}, \mathbf{V} \in \mathbb{C}^{N \times N}$ are unitary matrices of left/right singular vectors, and $\mathbf{\Sigma} = \text{diag}\,(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0) \in \mathbb{C}^{M \times N}$ is the diagonal matrix composed of (non-negative and real) singular values $\sigma_1 > \ldots > \sigma_r > 0$ where r is the rank of $\mathbf{Y}$. The mitigation techniques assume that the wall and target subspaces can be separated. Then:

$$\mathbf{Y} = \sum_{i \in \mathcal{W}} \sigma_i \mathbf{u}_i \mathbf{v}_i^H + \sum_{i \in \mathcal{T}} \sigma_i \mathbf{u}_i \mathbf{v}_i^H + \sum_{i \in \mathcal{N}} \sigma_i \mathbf{u}_i \mathbf{v}_i^H \tag{1.18}$$

with $\mathcal{W}$,$\mathcal{T}$ and $\mathcal{N}$ the (disjoint) sets of indices for wall, target, and noise singular components. We remove the wall returns by projecting the signal matrix onto the orthogonal complement to the wall subspace. The projector onto the wall subspace is:

$$\mathbf{\Pi}_w = \sum_{i \in \mathcal{W}} \mathbf{u}_i \mathbf{v}_i^H \tag{1.19}$$

Thus, projecting the signal matrix onto the orthogonal complement of the wall subspace is achieved by:

$$\hat{\mathbf{Y}} = (\mathbf{I} - \mathbf{\Pi}_w \mathbf{\Pi}_w^H)\mathbf{Y} \tag{1.20}$$

$\hat{\mathbf{Y}}$ is mostly composed of target returns with small or no returns from the wall. A similar procedure is proposed to remove noise. An AIC (Akaike Information

Criterion) can be derived [Wax and Kailath, 1985] to determine the dimension of the target subspace. Let:

$$AIC(i) = -2\log\left(\frac{\prod_{j=i+1}^{M}\sigma_j^{1/(M-i)}}{1/(M-i)\sum_{j=i+1}^{M}\sigma_j}\right)^{(M-i)N} + 2i(2M-i) \tag{1.21}$$

where i determines the optimal dimension of the target subspace. Then the optimal i is:

$$i^\star = \arg\min_i AIC(i) \tag{1.22}$$

Thus, the index set generating the noise subspace is : $\mathcal{N} = \{i^\star + 1, \ldots, N\}$. We can project $\hat{\mathbf{Y}}$ against the orthogonal complement of the noise subspace to obtain the signal matrix with noise removed:

$$\begin{aligned}
&\hat{\mathbf{Y}} \overset{\text{SVD}}{=} \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^H \\
&\mathbf{\Pi}_n = \sum_{i\in\mathcal{N}} \hat{\mathbf{u}}_i\hat{\mathbf{v}}_i^H \\
&\bar{\mathbf{Y}} = (\mathbf{I} - \mathbf{\Pi}_n\mathbf{\Pi}_n^H)\hat{\mathbf{Y}}
\end{aligned} \tag{1.23}$$

We are thus able to mitigate the wall influence from the radar returns. We know turn to the propagation delay through a simple wall.

1.2.2 . Through wall propagation delay

Contrary to free space SAR imaging, the delay of the received signal at each antenna is impacted by a wall. We can solve it, in the simplest case of a homogeneous wall (i.e. a dielectric slab), in order to correct the BP focus. Hereby, we consider a homogeneous wall of thickness d and dielectric constant ϵ located along the x-axis at a distance z_{off} to the SAR transceivers. Consider a propagation from the n^{th} transceiver at position $\mathbf{x}_{tm}$ to a target in the q^{th} pixel, with position $\mathbf{x}_q$, as shown in Figure 1.7. The angle of refraction ψ_{mq}, as shown in Figure 1.7, is deduced from the Snell-Descartes law as:

$$\psi_{mq} = \arcsin\left(\frac{\sin\theta_{mq}}{\sqrt{\epsilon}}\right) \tag{1.24}$$

with θ_{mq} the angle of incidence. Moreover, we can deduce the delays:

$$\begin{aligned}
l_{mq,air,1} &= \frac{z_{off}}{\cos\theta_{mq}} \\
l_{mq,wall} &= \frac{d}{\cos\psi_{mq}} \\
l_{mq,air,2} &= \frac{z_q - d}{\cos\theta_{mq}}
\end{aligned} \tag{1.25}$$

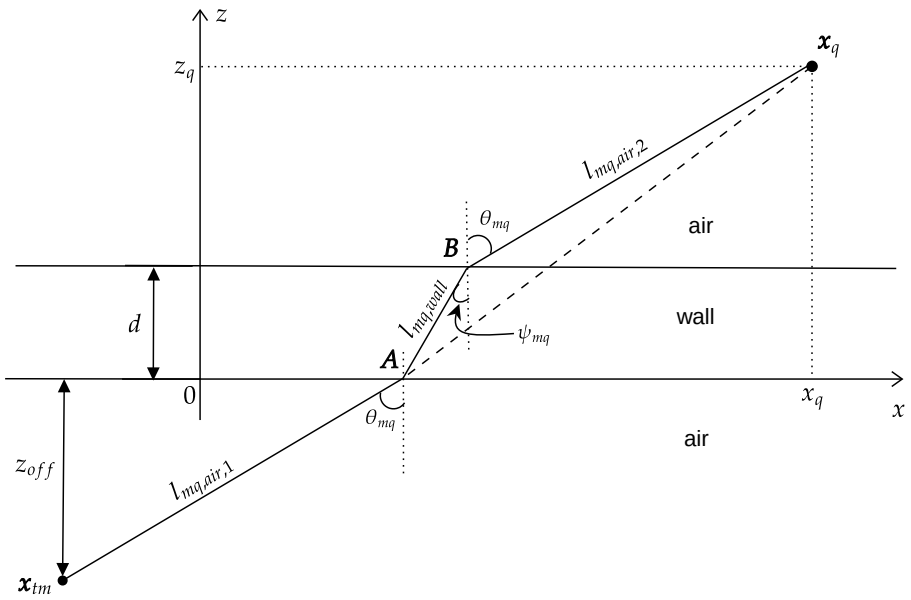


Figure 1.7: Propagation scheme through a wall

The point $\mathbf{A}$ in Figure 1.7 has coordinates $(x_{tm} + z_{off} \tan\theta_{mq}, 0)$. From the law of cosines applied to the triangle with vertices $(\mathbf{A}, \mathbf{B}, \mathbf{x}_q)$, we get:

$$\begin{aligned}(x_q - (x_{tm} + z_{off}\tan\theta_{mq}))^2 + z_q^2 = l_{mq,wall}^2 + l_{mq,air,2}^2 \\ - 2l_{mq,wall}l_{mq,air,2}\cos(\pi + \psi_{mq} - \theta_{mq})\end{aligned} \tag{1.26}$$

Equation (1.26) is a transcendantal equation in θ_{mq} as we fix the values for the radar and target positions. Solving for θ_{mq} amounts to finding the roots of $f(\theta_{mq})$ defined by:

$$\begin{aligned}f(\theta_{mq}) = (x_q - (x_{tm} + z_{off}\tan\theta_{mq}))^2 + z_q^2 \\ - (l_{mq,wall}^2 + l_{mq,air,2}^2 - 2l_{mq,wall}l_{mq,air,2}\cos(\pi + \psi_{mq} - \theta_{mq}))\end{aligned} \tag{1.27}$$

Thus, we can solve for $f(\theta_{mq}) = 0$ numerically using a root-finding algorithm, such as Newton's method, which can handle transcendental equations. At iteration $k+1$, the update rule of Newton's method is:

$$\theta_{mq}^{k+1} = \theta_{mq}^{k} - \frac{f(\theta_{mq}^{k})}{f'(\theta_{mq}^{k})} \tag{1.28}$$

The derivative f' is easily computed (by hand or using algebraic computation software). A good starting point θ_{mq}^0 so that Newton's method converges can be found using as an approximation a free-space propagation (without wall refractions). Finally, the (two-way) propagation delay is:

$$\tau_{mq} = \frac{2l_{mq,air,1}}{c} + \frac{2l_{mq,wall}}{v} + \frac{2l_{mq,air,2}}{c}, \quad v = \frac{c}{\sqrt{\epsilon}} \tag{1.29}$$

Algorithm 1 BP($\bar{\mathbf{Y}} \in \mathbb{C}^{M\times N}, N_x, N_z$)

1: Initialize the BP image: $\mathbf{I} \leftarrow \mathbf{0} \in \mathbb{C}^{N_x \times N_z}$
2: **for all** $n_x = 1, \dots N_x, n_z = 1, \dots N_z$ **do**
3: $\quad [\mathbf{I}]_{n_x,n_z} = \sum_{n=1}^{N} [\bar{\mathbf{Y}}]_{i,n}$
4: $\quad$ with $i = i_n(n_x, n_z)$ as in (1.15) with delay from (1.29)
5: **end for**
6: $\mathbf{I} \leftarrow |\mathbf{I}|$

We may apply the BP algorithm with those delays after mitigation of the wall returns to achieve our goal of imaging through a wall. However, this procedure cannot handle multipaths which causes false detections. It is also not robust to noise. This motivates the introduction of another approach to solve the problem.

1.3 . Sparse recovery approach

1.3.1 . Forward model

We introduce below a forward model [Amin and Ahmad, 2013] leading to a regularized inverse problem, in order to recover the target positions from the radar returns. It is a special case of the model in (1.9) for TWRI, by adding the returns from the wall to the model and not using a modulation signal e. Consider a N-element array with the n^{th} transceiver sending a stepped-frequency signal of M equispaced frequencies over the bandwidth $\omega_M - \omega_1$:

$$\omega_m = \omega_1 + m\Delta\omega, \quad m = 0, 1, \dots, M-1 \tag{1.30}$$

with ω_1 the lowest frequency and $\Delta\omega$ the step size in frequency. The reflections from targets is measured only at the same transceiver. Let the returned signal for the m^{th} frequency and n^{th} position be:

$$y(m,n) = \underbrace{\sum_{k=1}^{K} \sigma_w^{(k)} \exp\left(-j\omega_m \tau_w^{(k)}\right)}_{\text{wall returns}} + \underbrace{\sum_{i=1}^{R}\sum_{p=1}^{P} \sigma_p^{(i)} \exp(-j\omega_m \tau_{p,n}^{(i)})}_{\text{target returns}} \tag{1.31}$$

with K the number of reverberations in the wall, R the number of multipath and P the number of targets. Additionally, $\sigma_w^{(k)}$ and $\tau_w^{(k)}$ are the complex overall attenuation coefficient and round-trip delay for the wall returns associated with the k^{th} reverberation while $\sigma_p^{(i)}$ and $\tau_{p,n}^{(i)}$ are those associated with the p^{th} target, i^{th} multipath and n^{th} radar position. Assume first that:

- the clutter coming from the wall has been removed.
- the multipath effects are negligible.

Then the signal is the simple superposition of reflected signal from all targets:

$$y(m,n) = \sum_{p=1}^{P} \sigma_p \exp\left(-j\omega_m \tau_{p,n}\right) \tag{1.32}$$

Equation (1.32) can be rewritten in a matrix-vector form. Assume the scene is divided into a grid of pixels of size $N_x \times N_z$ (in crossrange vs downrange) as shown in Figure 1.8.

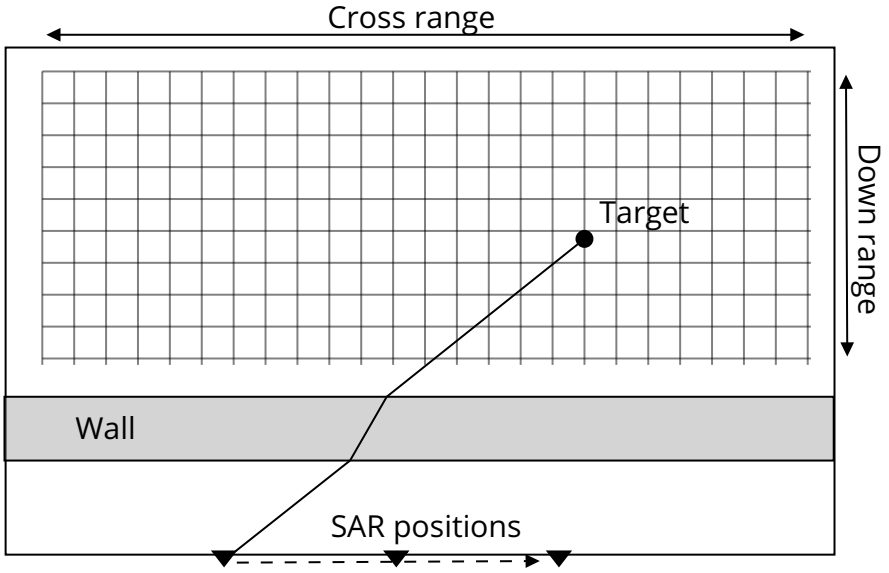


Figure 1.8: Division of the scene in a grid

Then the received signal at the n^{th} transceiver is:

$$\mathbf{y}_n = \mathbf{\Psi}_n \mathbf{r} \tag{1.33}$$

where $\mathbf{r} \in \mathbb{C}^{N_x N_z}$ is a vector whose k^{th} component equals zero except if the p^{th} target lies in the space occupied by this pixel, in which case it takes the value σ_p. $\mathbf{\Psi}_n \in \mathbb{C}^{M \times N_x N_z}$ is a dictionary with m^{th} row being:

$$[\mathbf{\Psi}_n]_m = [\exp\left(-j\omega_m \tau_{1,n}\right) \ldots \exp\left(-j\omega_m \tau_{(N_x N_z),n}\right)] \tag{1.34}$$

Expressed in another way, each column of $\mathbf{\Psi}_n$ represents the expected return at the n^{th} SAR position, from some target located at a specific grid point, to be scaled by an attenuation factor. We can consider all N antenna measurements in one vector:

$$\mathbf{y} = [\mathbf{y}_1^T \, \mathbf{y}_2^T \, \ldots \, \mathbf{y}_N^T]^T \tag{1.35}$$

As well as all N dictionnaries:

$$\mathbf{\Psi}_A = [\mathbf{\Psi}_1^T \, \mathbf{\Psi}_2^T \, \ldots \, \mathbf{\Psi}_N^T]^T \tag{1.36}$$

Leading to a linear model:

$$\mathbf{y} = \mathbf{\Psi}_A \mathbf{r} \tag{1.37}$$

The vector $\mathbf{r}$ is assumed to be sparse, i.e. with a few numbers of non-zero entries, as there are few targets assumed in the scene. This lends itself to sparse recovery methods where the linear model is regularized by a sparsity inducing function on $\mathbf{r}$:

$$\min_{\mathbf{r}} \|\mathbf{r}\|_0 \text{ s.t. } \mathbf{y} = \mathbf{\Psi}_A \mathbf{r} \tag{1.38}$$

where $\|\mathbf{r}\|_0$ is the ℓ_0 pseudo-norm counting the number of non-zero entries. It can be relaxed convexly to:

$$\min_{\mathbf{r}} \|\mathbf{r}\|_1 \text{ s.t. } \mathbf{y} = \mathbf{\Psi}_A \mathbf{r} \tag{1.39}$$

where $\|\mathbf{r}\|_1$ is the ℓ_1 norm defined as the sum of the absolute values of the entries. For now, we assume we may obtain an approximate solution via Orthogonal Matching Pursuit (OMP, see Chapter 2 for more details). Then an image is formed by unvectorizing the recovered $\mathbf{r}$ and taking its amplitude. Via Compressive Sensing [Donoho, 2006], we can reconstruct $\mathbf{r}$ based on only a subset of all measurements $\mathbf{y}$. We can use a measurement matrix $\mathbf{\Phi} \in \mathbb{R}^{Q_1 Q_2 \times MN}$ such that:

$$\mathbf{\Phi}\mathbf{y} = \mathbf{\Phi}\mathbf{\Psi}_A \mathbf{r} \tag{1.40}$$

where:

$$\mathbf{\Phi} = \boldsymbol{\theta} \otimes \mathbf{I}_{Q_1} \cdot \operatorname{diag}\{\boldsymbol{\phi}^1, \boldsymbol{\phi}^2, \ldots, \boldsymbol{\phi}^N\} \tag{1.41}$$

with $\mathbf{I}_{Q_1} \in \mathbb{R}^{Q_1 \times Q_1}$ an identity matrix. $\boldsymbol{\theta} \in \mathbb{R}^{Q_2 \times N}$ is constructed by random selection of rows of $\mathbf{I}_N$, denoting the sampling of transceiver locations, while $\boldsymbol{\phi}^n \in \mathbb{R}^{Q_1 \times M}$ with $n = 1, 2, \ldots, N$ is constructed by random sampling of rows of $\mathbf{I}_M$ and denotes the frequencies sampled at the n^{th} transceiver.

Having presented basic methods for TWRI, we present some results for illustrative purposes before delving into further details.

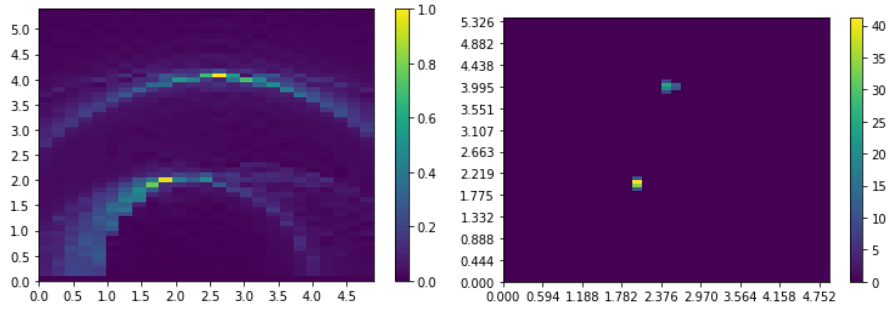
1.3.2 . Experiments on simulated data

Setup

We generate a received signal according to the model described in (1.32), i.e. via raytracing. The scene is 4.9×5.4 m in crossrange (x-axis) vs downrange(z-axis). Two targets are situated at (x, z) coordinates $(2, 2)$ and $(2.5, 4)$. The Signal to Noise Ratio (SNR) is set to 80 dB with noise modeled by a complex Gaussian white noise. The stepped-frequency signal is composed of 728 frequencies from 1 Ghz to 3 Ghz. The SAR moves 0.0187 m along the x-axis between each acquisition with 67 different positions overall. Its track is centered over the x-axis, thus it starts around $x = 1.82$ and ends at $x = 3.05$. The wall (positioned parallel to the SAR displacement axis) is at a standoff distance to the SAR of 1.2 m, of thickness 0.5 m, of relative permittivity $\epsilon = 4.5$. We compared the sparse reconstruction with the classical BP.

Results

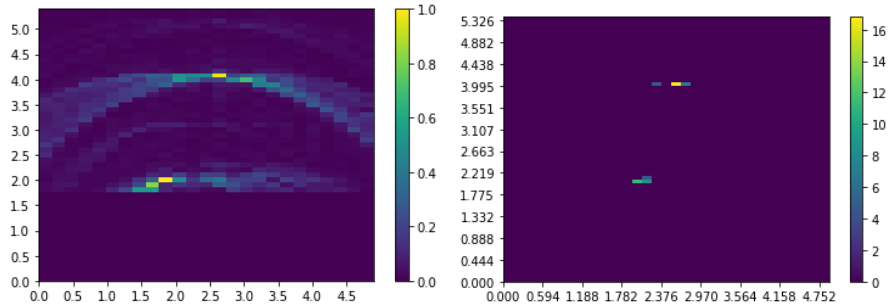
First, we compare the algorithms in free space, i.e. without a wall, displayed in Figure 1.9. Secondly, we add a wall. We use the wall mitigation technique described above to erase the wall returns from the image. The propagation time of a wave is impacted by that wall. For example in a direct path, it is refracted entering and leaving the wall. The dictionary (1.36) can take this propagation delay into account. As seen in Figure 1.10, the result is satisfying as the targets are detected at the right places with no return detected from the wall.



(a) Backprojection in free-space (targets at $(2, 2)$ and $(2.5, 4)$)

(b) Via OMP in free-space (targets at $(2, 2)$ and $(2.5, 4)$)

Figure 1.9: Methods in free-space



(a) Backprojection with wall mitigation (targets at $(2, 2)$ and $(2.5, 4)$)

(b) Via OMP with wall mitigation (targets at $(2, 2)$ and $(2.5, 4)$)

Figure 1.10: Methods with a mitigated wall

1.4 . Multipath exploitation

BP cannot handle multipath. We will present a way to achieve this via structured sparse methods. For multipath exploitation, interior wall reflections on side walls and wall ringing inside the front wall are considered. This will enable us to create a model to integrate them.

1.4.1 . Accounted types of multipath

Interior walls

For the first case, consider the p^{th} target located at $\mathbf{x}_p = (x_p, z_p)$ and the interior wall parallel to the z-axis at location $x = x_w$ (see Figure 1.11). Then, the propagation from the n^{th} antenna to the target with an interior wall reflection is along the path P'' and back from P'. With the assumption of specular reflection at the wall, the return path P' can be constructed as a direct path $\tilde{P}'$ to a virtual target at $\mathbf{x'}_p = (2x_w - x_p, z_p)$. This simplifies the calculation as we just have to compute the direct (still through the front wall) path to $\mathbf{x'}_p$. The propagation delay associated with this interior wall multipath is thus the sum of two (through the front wall) direct paths, which we know how to compute.

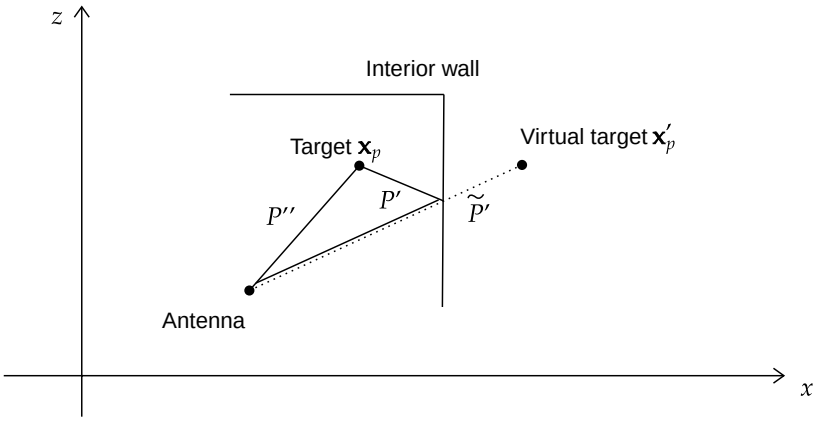


Figure 1.11: Multipath propagation via reflection at an interior wall

Wall ringing

For the wall ringing multipath, we can express from geometrical considerations (see Figure 1.12):

$$\Delta x = (\Delta z - d)\tan(\theta_{air}) + d(1 + 2i_w)\tan(\theta_{wall}) \tag{1.42}$$

with Δx and Δz the distances in cross-range and down-range between the current transceiver position and the target, θ_{air} and θ_{wall} the angles in air and the wall, respectively. The integer i_w denotes the number of internal reflections (within the wall), with $i_w = 0$ being the special case of direct through-the-wall propagation derived previously.

From Snell's law, we can state another equation in θ_{air} and θ_{wall}:

$$\frac{\sin\theta_{air}}{\sin\theta_{wall}} = \sqrt{\epsilon} \tag{1.43}$$

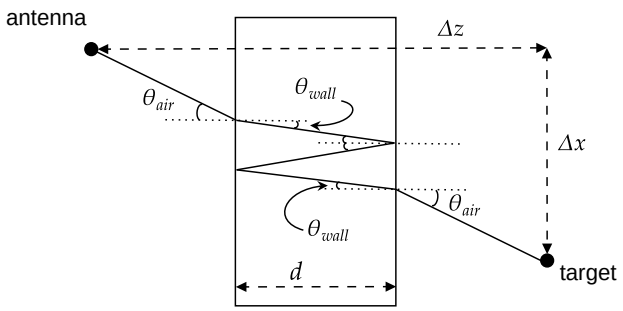


Figure 1.12: Multipath propagation via internal bounces ("wall ringing")

Equations (1.42) and (1.43) form a nonlinear system of equations in θ_{air} and θ_{wall}. Let $f : \mathbb{R}^2 \to \mathbb{R}^2$, $f(\theta_{air}, \theta_{wall}) = (f_1(\theta_{air}, \theta_{wall}), f_2(\theta_{air}, \theta_{wall}))$, where:

$$\begin{aligned} f_1(\theta_{air}, \theta_{wall}) &= (\Delta z - d)\tan\theta_{air} + d(1 + 2i_w)\tan\theta_{wall} - \Delta x \\ f_2(\theta_{air}, \theta_{wall}) &= \frac{\sin\theta_{air}}{\sin\theta_{wall}} - \sqrt{\epsilon} \end{aligned} \tag{1.44}$$

We are then searching for some root of this function. This can be solved using any standard numerical method for root-finding of vector-valued functions. For example, Newton's method extended to a system of equations is, at iteration $k + 1$:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (Df(\mathbf{x}_k))^{-1} f(\mathbf{x}_k) \tag{1.45}$$

where $Df(\mathbf{x})$ is the jacobian (the matrix of 1^{st} order partial derivatives) of f at x. In our case, it is square. This can be expressed as solving the following linear system in $\Delta\mathbf{x} \triangleq \mathbf{x}_{k+1} - \mathbf{x}_k$ and updating $\mathbf{x}_k$:

$$\begin{aligned} Df(\mathbf{x}_k)\Delta\mathbf{x} &= -f(\mathbf{x}_k) \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \Delta\mathbf{x} \end{aligned} \tag{1.46}$$

Many more advanced methods exist such as Powell's method, Levenberg-Marquardt algorithm, Broyden's Quasi-Newton methods,etc. Once this system solved for θ_{air} and θ_{wall}, we can express the associated propagation delay as:

$$\tau = \frac{\Delta z - d}{c\cos\theta_{air}} + \frac{\sqrt{\epsilon} d(1 + 2i_w)}{c\cos\theta_{wall}} \tag{1.47}$$

1.4.2 . Forward model with multipath exploitation

We are now able to compute the propagation delays of some types of multipath propagation of the signal, we can develop a model to exploit it.

Again, suppose the front wall returns suppressed. Note that a round-trip is divided into depart and return paths from emitter to target and back to receiver. Supposing R_1 possible depart paths and R_2 possible return paths, we get $R \leq R_1 R_2$ possible round-trips. In practice $R \ll R_1 R_2$ due to symmetry or strong attenuation of some paths.

The round-trip delay associated to some multipath indexed by $i \in [0, R-1]$, is:

$$\tau_{p,n}^{(i)} = \tau_{p,n}^{(i_1)} + \tau_{p,n}^{(i_2)} \tag{1.48}$$

where $i_1 \in [0, R_1 - 1]$ and $i_2 \in [0, R_2 - 1]$ index the depart and return path taken in the i^{th} multipath scheme.

Furthemore, we associate a complex amplitude $w_p^{(i)}$ to the i^{th} multipath at the p^{th} target. Assuming P point targets and the wall returns erased, the received signal at the n^{th} transceiver for the m^{th} frequency is:

$$y(m,n) = \sum_{i=1}^{R} \sum_{p=1}^{P} w_p^{(i)} \sigma_p^{(i)} \exp(-j\omega_m \tau_{p,n}^{(i)}) \tag{1.49}$$

Equivalently:

$$\mathbf{y} = \mathbf{\Psi}^{(1)}\mathbf{r}^{(1)} + \mathbf{\Psi}^{(2)}\mathbf{r}^{(2)} + \ldots + \mathbf{\Psi}^{(R)}\mathbf{r}^{(R)} \tag{1.50}$$

This can be written compactly in matrix-vector form as:

$$\begin{aligned} \underbrace{\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}}_{\mathbf{y}} &= \underbrace{[\mathbf{\Psi}^{(1)}, \mathbf{\Psi}^{(2)}, \ldots, \mathbf{\Psi}^{(R)}]}_{\mathbf{\Psi}_A} \underbrace{\begin{bmatrix} \mathbf{r}^{(1)} \\ \mathbf{r}^{(2)} \\ \vdots \\ \mathbf{r}^{(R)} \end{bmatrix}}_{\mathbf{r}} \\ \implies \mathbf{y} &= \mathbf{\Psi}_A \mathbf{r} \end{aligned} \tag{1.51}$$

where $\mathbf{r} \in \mathbb{C}^{N_x N_z R}$ and $\mathbf{\Psi}_A \in \mathbb{C}^{MN \times N_x N_z R}$. Each $\mathbf{r}^{(i)} \in \mathbb{C}^{N_x N_z}$ is the scene vector associated with the i^{th} multipath propagation scheme and $\mathbf{\Psi}^{(i)} \in \mathbb{C}^{MN \times N_x N_z}$ is the dictionary matrix with propagation delay computed according to the i^{th} multipath scheme:

$$\mathbf{\Psi}^{(i)} = \begin{bmatrix} \mathbf{\Psi}_1^{(i)} \\ \mathbf{\Psi}_2^{(i)} \\ \vdots \\ \mathbf{\Psi}_N^{(i)} \end{bmatrix} \tag{1.52}$$

Again, for Compressive Sensing purposes, we can use a measurement matrix $\mathbf{\Phi} \in \mathbb{R}^{Q_1 Q_2 \times MN}$ as in (1.41) to reduce the number of measurement used.

The crucial point to reconstruct $\mathbf{r}$ is to note that all scene vectors $\mathbf{r}^{(i)}, i \in [1, \ldots, R]$ describe the same physical scene. Thus, a target will appear at the

same pixel location in different scene vectors (associated with different propagation paths). However, ghost targets from multipath effects will not appear across all propagation paths at the same pixel location.

The idea is then to enforce group-sparsity across scene vectors, where a group is formed of the same pixel location across all scene vectors. This suggests a regularization by the $\ell_{2,1}$-norm, which is defined as the sum of the Euclidean norm of the rows of some matrix. It induces structured sparsity, across rows. The use of the $\ell_{2,1}$ norm in order to promote structured sparsity across rows [Kowalski, 2009] has been developed for multipath exploitation in [Leigsnering et al., 2014]. We are then looking to resolve the following optimization problem:

$$\min_{\mathbf{r}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Psi}_A \mathbf{r}\|^2 + \lambda \left\|\mathrm{vec}^{-1}(\mathbf{r})\right\|_{2,1} \tag{1.53}$$

where the $\ell_{2,1}$-norm is taken over an unvectorized form of $\mathbf{r}$. Indeed, vec^{-1} is the inverse operator of vectorization: $\mathrm{vec}^{-1}(\mathrm{vec}(\mathbf{A})) = \mathbf{A}$. We assume we may get an approximate solution. We will see the method (Proximal Gradient Descent i.e. PGD) in the next Chapter 2. Having obtained $\mathbf{r}$, we need to form an image $\mathbf{I}$ from all $\mathbf{r}^{(i)}, i \in [1, \ldots, R]$ composing it. One way is to average squared amplitudes across multipaths (and unvectorize):

$$[\mathrm{vec}(\mathbf{I})]_q = \frac{1}{R}\left\|[\mathbf{r}^{(1)}]_q \ \ldots \ [\mathbf{r}^{(R)}]_q]\right\|_F^2 \tag{1.54}$$

1.4.3 . Experiments on simulated data

We showcase here the advantage of considering multipath effects via the sparse recovery model versus BP imaging.

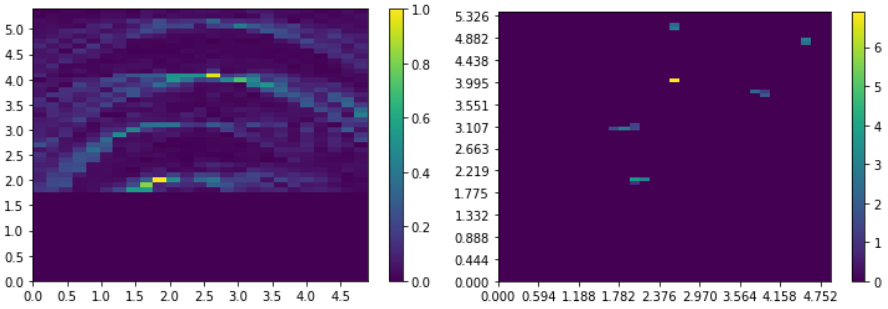
Setup

The signal is generated via raytracing according to (1.49). Interior wall multipath is considered with two side walls (parallel to the z-axis) on the edges of the scene. Wall-ringing multipath is only considered with $i_w = 1$ i.e. one internal reflection. The multiplication attenuation factor for every multipath is 0.1. The radar specifications are the same as for the previous simulations. There are still two point-like targets at coordinates $(2, 2)$ and $(2.5, 4)$.

Results

First, we look at the results of the algorithms that do not take into account the multipath when it is indeed present. We only consider the case where the front-wall returns have been suppressed. The results are shown in Figure 1.13. We need to exploit the multipath model to reduce phantom targets as seen in Figure 1.14. We can observe the effect of increasing the thresholding

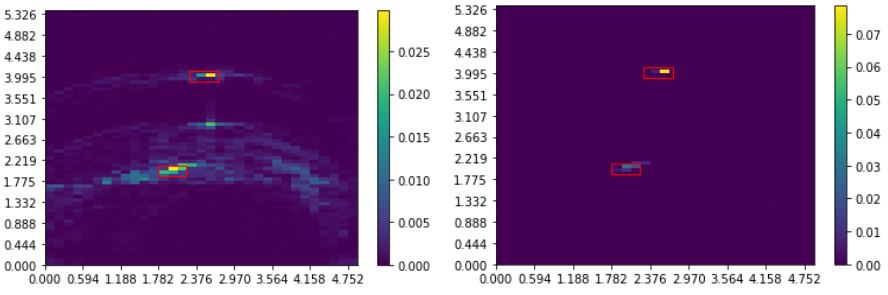
parameter λ. Next, we add the returns of a front wall and the wall mitigation technique as a preprocessing step. In Figure 1.15, on the first image, we clearly see that a target is hidden by wall returns. In the second image, we can clearly see both targets while ghosts and wall returns have been eliminated.



(a) Backprojection with multipath (front wall suppressed) and targets at $(2, 2)$, $(2.5, 4)$

(b) Compressive Sensing (OMP) with multipath (front wall suppressed) and targets at $(2, 2)$, $(2.5, 4)$

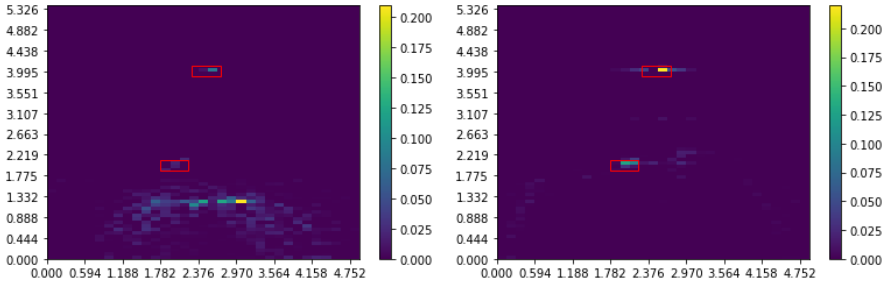
Figure 1.13: Methods without multipath considerations (front wall suppressed)



(a) Compressive Sensing (PGD) with $\lambda = 1$ (front-wall suppressed)

(b) Compressive Sensing (PGD) with $\lambda = 100$ (front-wall suppressed)

Figure 1.14: Methods with multipath considerations (front wall suppressed)



(a) Compressive Sensing (PGD) with $\lambda = 100$ and front-wall returns

(b) Compressive Sensing (PGD) with $\lambda = 100$ and wall mitigation

Figure 1.15: Methods with multipath considerations and front wall

We presented a classical TWRI method [Amin and Ahmad, 2013]. It consists in two steps: a wall mitigation followed by a structured sparse reconstruction of the targets. We hereby denote it SR-CS (Sparse Recovery - Compressed Sensing).

However, it may not be optimal as some target information may be erased by the wall mitigation. Parallel retrieval of both the wall and targets components is possible via matrix decomposition methods, which we propose to study. In the next chapter, we first present optimization methods that we leverage later on.

2 - Overview of inverse methods

Contents

Before entering the heart of the matter with our new TWRI methods, we summarize some optimization techniques in this chapter.

2.1 . Basics of sparse recovery/coding

We have seen in the previous chapter that TWRI can be formulated as the recovery of a sparse vector. More broadly, sparse recovery/sparse coding aims at recovering a solution to an underdetermined system of equations which is the sparsest. It uses the ℓ_0 norm (while being the limit of ℓ_p norms when p tends to zero but it is in fact not a true norm), defined as the number of non-zero entries of the vector. Sparse recovery problems thus consider the following problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{x}\|_0 \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{y} \end{aligned} \tag{2.1}$$

While the above problem is NP-hard, we may obtain an approximate solution using several methods. We will present the two main approaches to solving it: greedy approaches and gradient-type alternatives.

2.1.1 . Greedy methods: resolution via OMP

Greedy approachs, which also fall under the name of matching pursuit methods, represent the signal in a sparse basis taken one by one from the columns of a dictionary matrix, also called atoms. The dictionary is called complete if it spans the whole signal space and redundant if it is composed of linearly dependent atoms. Often, the dictionary is complete and redundant. Thus, these methods search for the expansion coefficients in that basis. For a signal $\mathbf{y}$ with coefficients $\{x_k\}$ in a basis of atoms $\{\mathbf{a}_k\}$, where k indexes an

element of the desired basis, we have:

$$\mathbf{y} = \sum_k x_k \mathbf{a}_k \tag{2.2}$$

Matching pursuit algorithms search for the best sparse approximation by choosing the atom of the dictionary which is most representative of the signal. This is achieved by choosing the one with the largest inner product with the residual signal $\mathbf{e}$:

$$k^\star = \arg\max_k |\langle \mathbf{e}, \mathbf{a}_k \rangle| \tag{2.3}$$

where $\mathbf{e}$ is the signal $\mathbf{y}$ from which we subtract the approximation by already chosen atoms:

$$\mathbf{e} = \mathbf{y} - \sum_{k \in \mathcal{K}} x_k \mathbf{a}_k \tag{2.4}$$

where $\mathcal{K}$ is the set of indices selected in previous iterations. The coefficient associated to the atom $\mathbf{a}_{k^\star}$ is then $x_{k^\star} = \langle \mathbf{e}, \mathbf{a}_{k^\star} \rangle$. OMP [Pati et al., 1993] further constrains the residual to be orthogonal to the previously selected atoms. This results in convergence for a d-sparse vector after at most d steps. The added procedure consists in projecting the residual on the orthogonal complement to the subspace spanned by previously selected atoms, thus removing any part of the residual lying in the subspace spanned by previously selected atoms. This ensures that already selected atoms will not be selected again. Under suitable conditions on the dictionary (mutual coherence or restricted isometry) it was shown the solution is unique and is the one recovered by OMP.

We show the algorithmic flow of OMP in Algorithm 2, where $\mathbf{\Lambda}$ denotes the support of the recovered sparse signal $\mathbf{x}$ through a dictionary $\mathbf{A}$ and measurements $\mathbf{y}$. $\mathbf{x}_{\mathbf{\Lambda}}$ denotes the vector $\mathbf{x}$ restricted to the support $\mathbf{\Lambda}$.

2.1.2 . Gradient methods: resolution via PGD

The sparse recovery problem is non-convex due to the ℓ_0 norm. It may be relaxed to a convex problem with:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{x}\|_1 \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{A}\mathbf{x} \end{aligned} \tag{2.5}$$

Indeed the ℓ_1 norm is the convex envelope of the ℓ_0 norm (on the unit ℓ_∞ ball, the dual norm of the ℓ_1 norm). It is defined as the sum of absolute values of the entries of the vector.

It can be further relaxed to include noisy observations:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{x}\|_1 \\ \text{s.t.} \quad & \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leq \epsilon \end{aligned} \tag{2.6}$$

Algorithm 2 OMP$(\mathbf{A}, \mathbf{y})$

$\mathbf{x}^0 \leftarrow 0$
$\mathbf{e}^0 \leftarrow \mathbf{y}$
$\mathbf{\Lambda}^0 \leftarrow \emptyset$
$k \leftarrow 0$
repeat
  $\mathbf{h}^{k+1} = \mathbf{A}^H \mathbf{e}^k$
  $\lambda^{k+1} = \arg\max_{j \notin \mathbf{\Lambda}^k} |\mathbf{h}_j^{k+1}|$
  $\mathbf{\Lambda}^{k+1} = \mathbf{\Lambda}^k \cup \{\lambda^{k+1}\}$
  $\mathbf{x}^{k+1} = \mathbf{0}$
  $\mathbf{x}^{k+1}_{\mathbf{\Lambda}^{k+1}} = \mathbf{A}^\dagger_{\mathbf{\Lambda}^{k+1}} \mathbf{y}$
  $\mathbf{y}^{k+1} = \mathbf{A}\mathbf{x}^{k+1}$
  $\mathbf{e}^{k+1} = \mathbf{y} - \mathbf{y}^{k+1}$
  $k \leftarrow k + 1$
until stopping criterion is met
$\mathbf{x} \leftarrow \mathbf{x}^k$ (reconstructed sparse signal)
$\mathbf{\Lambda} \leftarrow \mathbf{\Lambda}^k$ (support of the reconstructed signal)
$\mathbf{e} \leftarrow \mathbf{e}^k$ (end residual vector)

In Lagrangian form, this yields the Least Absolute Shrinkage and Selection Operator (LASSO) [Tibshirani, 1996]:

$$\min_{\mathbf{x}} \quad \lambda \|\mathbf{x}\|_1 + \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \tag{2.7}$$

where λ is a scalar hyper-parameter that balances the regularization strength. A method to solve this optimization problem is the Proximal Gradient Descent (PGD) [Parikh and Boyd, 2014]. PGD aims at minimizing a composite function:

$$f(\mathbf{X}) = g(\mathbf{X}) + h(\mathbf{X}) \tag{2.8}$$

where g is a differentiable and (proper closed i.e. lower semi-continuous) convex function whereas h is a (proper closed) convex function but not necessarily differentiable, making it impossible to use a standard gradient descent. However, h can be extended valued (taking infinite values) to encode constraints (using an indicator function). The following definitions use a matrix variable which can be reduced to a vector.

Definition 1. *The proximal operator [Parikh and Boyd, 2014] of a convex function $h : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X}$ is some Hilbert space, is:*

$$\text{Prox}_h(\mathbf{V}) = \arg\min_{\mathbf{X} \in \mathcal{X}} \quad h(\mathbf{X}) + \frac{1}{2} \|\mathbf{X} - \mathbf{V}\|_F^2 \tag{2.9}$$

which is unique due to the strict convexity of the objective function.

Gradient methods using proximals are very important in convex optimization. A good reference for more details is [Parikh and Boyd, 2014] from which the following overview is inspired.

Definition 2. *Each iteration $k = 0, 1, 2, \ldots$ of the Proximal Gradient Descent (PGD) method [Parikh and Boyd, 2014, Section 4.2] takes the form:*

$$\mathbf{X}_{k+1} = \mathrm{Prox}_{th}(\mathbf{X}_k - t\nabla g(\mathbf{X}_k)) \tag{2.10}$$

where t denotes a step-size. For a L-Lipschitz ∇g, it converges sub-linearly at the rate $\mathcal{O}(1/k)$ (in objective function value) for $t \in [0, 1/L)$.

Definition 3. *A function f defined on a normed space with norm $\|\cdot\|$ mapping to another normed space (via the induced norm) is L-Lipschitz continuous if there exist a real constant $L > 0$ such that:*

$$\|f(\mathbf{X}) - f(\mathbf{Z})\| \leq L\|\mathbf{X} - \mathbf{Z}\| \tag{2.11}$$

Any L veryfing this condition is called a Lipschitz constant of f.

In the case of LASSO:

$$g(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \quad h(\mathbf{x}) = \lambda\|\mathbf{x}\|_1 \tag{2.12}$$

Interestingly, the proximal of some norms are computable in close-form. The following propositions can be found in [Parikh and Boyd, 2014, Section 6.5].

Proposition 1. *the proximal operator of the ℓ_1-norm, the so-called soft-thresholding operator $S_\lambda : \mathbb{C} \to \mathbb{R}$ is:*

$$S_\lambda(z) = \mathrm{sgn}(z)(|z| - \lambda)_+ = \begin{cases} z - \mathrm{sgn}(z)\lambda & \text{if } |z| > \lambda \\ 0 & \text{if } |z| \leq \lambda \end{cases} \tag{2.13}$$

with the sign operator $\mathrm{sgn} : \mathbb{C} \to \mathbb{R}, z \mapsto \mathrm{sgn}(z) = \frac{z}{|z|}$ (if $z \neq 0$ in which case the sign is zero) and the positive part operator $(\cdot)_+ : \mathbb{R} \to \mathbb{R}^+, x \mapsto (x)_+ = \max(0, x)$. Over matrices, it is used element-wise, as the proximal can be separated across entries:

$$[S_\lambda(\mathbf{X})]_{ij} = S_\lambda([\mathbf{X}]_{ij}) \tag{2.14}$$

with $[\mathbf{X}]_{ij}$ the element in the i row and j column of $\mathbf{X}$.

Proposition 2. *Consider the nuclear norm, defined as the sum of the singular values of a matrix. the proximal operator of the nuclear norm is $D_\lambda(\mathbf{X}) = \mathbf{U}S_\lambda(\mathbf{\Sigma})\mathbf{V}^H$ where $\mathbf{X} \overset{\text{SVD}}{=} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ is the SVD of $\mathbf{X}$.*

Proposition 3. *the proximal operator of the $\ell_{2,1}$ norm, denoted T_λ , operates row by row over some $\mathbf{A}$. For the i^{th} row $\mathbf{a}_i$, it is:*

$$[T_\lambda \mathbf{A}]_{i:} = \left(1 - \frac{\lambda}{\|\mathbf{a}_i\|}\right)_+ \mathbf{a}_i \tag{2.15}$$

where $(x)_+ = max(x, 0)$.

We may easily derive the gradient of g at x:

$$\nabla g(\mathbf{x}) = -\mathbf{A}^H(\mathbf{y} - \mathbf{A}\mathbf{x}) \tag{2.16}$$

As well as the Hessian:

$$Hg(\mathbf{x}) = \mathbf{A}^H\mathbf{A} \tag{2.17}$$

which takes the form of a Gram matrix. The choice of t follows from the derivation of PGD as a Majorization Minimization (MM) method [Parikh and Boyd, 2014], by taking a quadratic upper bound of the function to minimize, leading to (2.10) with $t \in (0, 1/L]$ (the method actually converges for $t \in (0, 2/L]$) where L is the Lipschitz constant of ∇g. For our particular problem, we can easily find L from the definition of a Lipschitz function and sub-multiplicativity of the spectral norm (the matrix norm induced by the ℓ_2 norm, which equals the largest singular value):

$$\begin{aligned} \|\nabla g(\mathbf{x}) - \nabla g(\mathbf{z})\|_2 &= \left\|\mathbf{A}^H(\mathbf{y} - \mathbf{A}\mathbf{x}) - \mathbf{A}^H(\mathbf{y} - \mathbf{A}\mathbf{z})\right\| \\ &= \left\|\mathbf{A}^H\mathbf{A}(\mathbf{x} - \mathbf{z})\right\|_2 \\ &\leq \left\|\mathbf{A}^H\mathbf{A}\right\|_2 \|\mathbf{x} - \mathbf{z}\|_2 \\ &= \|\mathbf{A}\|_2^2 \|\mathbf{x} - \mathbf{z}\|_2 \end{aligned} \tag{2.18}$$

Thus, we set the step-size to $t = \frac{1}{\|\mathbf{A}\|_2^2}$ where we can note that $\|\mathbf{A}\|_2^2$ is also the largest eigenvalue value of the Hessian Hg. We recapitulate this in Algorithm 3. Note that this method is also called the Iterative Shrinkage and Thresholding Algorithm (ISTA).

Algorithm 3 PGD for LASSO (ISTA) $(\lambda, \mathbf{y}, \mathbf{A})$

1: $t \leftarrow 1/\|\mathbf{A}\|_2^2$
2: $\mathbf{x}_0 \leftarrow \mathbf{0}$
3: **repeat** (for $k = 0, 1, 2, \ldots$)
4: $\quad \mathbf{x}_{k+1} = S_{\lambda t}\left(\mathbf{x}_k + t\mathbf{A}^H(\mathbf{y} - \mathbf{A}\mathbf{x}_k)\right)$
5: **until** stopping criterion is met
6: $\mathbf{x} \leftarrow \mathbf{x}_k$

This may be accelerated in Nesterov's way by an extrapolation step to give the so-called Accelerated Proximal Gradient Descent (APGD) [Parikh and

Boyd, 2014, Section 4.3]. This method is also called the Fast Iterative Shrinkage and Thresholding Algorithm (FISTA) [Beck and Teboulle, 2009]. This changes the rate of convergence from $\mathcal{O}(1/k)$ for PGD to $\mathcal{O}(1/k^2)$ for APGD. This is summarized in Algorithm 4.

Algorithm 4 APGD for LASSO (FISTA) $(\lambda, \mathbf{y}, \mathbf{A})$

$t \leftarrow 1/\|\mathbf{A}\|_2^2$
$\theta_0 \leftarrow 0$
$\mathbf{x}_0 \leftarrow \mathbf{0}$
repeat (for $k = 0, 1, 2, \ldots$)
    $\hat{\mathbf{x}}_{k+1} = \mathbf{x}_k + \theta_k(\mathbf{x}_k - \mathbf{x}_{k-1})$
    $\mathbf{x}_{k+1} = S_{\lambda t}\left(\hat{\mathbf{x}}_{k+1} + t\mathbf{A}^H(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{k+1})\right)$
    $\theta_{k+1} = \frac{k+1}{(k+1)+3}$
until stopping criterion is met
$\mathbf{x} \leftarrow \mathbf{x}_k$

Later on, we will be interested in solving more involved problems that involve several variables under constraints whose algorithmic resolution calls for more advanced methods. We present an overview before using them in the next chapters. Those include various notions that may not usually be seen along each other.

2.2 . More advanced methods

We present more advanced notions that we will use in later chapters.

2.2.1 . MM

The framework of Majorization Minimization (MM) [Sun et al., 2016] is a generic one that englobes a wide variety of methods: Expectation-Maximisation (EM), Proximal Gradient Descent (PGD), etc. It consists of iteratively:

- majorizing tightly the function of interest by another function
- minimizing this majorizer

which is shown in Figure 2.1. More precisely, consider the problem:

$$\min_{\mathbf{X}} f(\mathbf{X}) \tag{2.19}$$

We may construct a local majorizer of $f(\mathbf{X})$ at the point $\mathbf{X}_t$ which we denote $g(\mathbf{X}|\mathbf{X}_t)$ that satisfies:

$$\begin{aligned} f(\mathbf{X}) &\leq g(\mathbf{X}|\mathbf{X}_t) \quad \forall \mathbf{X} \\ g(\mathbf{X}_t|\mathbf{X}_t) &= f(\mathbf{X}_t) \end{aligned} \tag{2.20}$$

By this construct, we can choose the next iterate by:

$$\mathbf{X}_{t+1} = \arg\min_{\mathbf{X}} g(\mathbf{X}|\mathbf{X}_t) \tag{2.21}$$

Then, we have:

$$f(\mathbf{X}_{t+1}) \leq g(\mathbf{X}_{t+1}|\mathbf{X}_t) \leq g(\mathbf{X}_t|\mathbf{X}_t) = f(\mathbf{X}_t) \tag{2.22}$$

This points out that this iteration scheme converges to a local minimum of the function f. The convergence speed depends on the majorizing function: the more it aligns with the original function, the less iterations are required, as the approximation error is reduced. However, the function g should be simple to minimize in order to have a straightforward/economical update. This means that there is a trade-off between these two objectives.

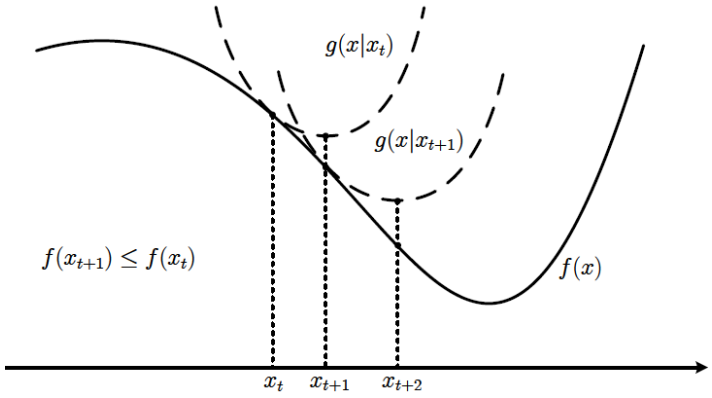


Figure 2.1: MM principle

2.2.2 . ADMM

The framework of Alternating Directions Method of Multipliers (ADMM) [Lions and Mercier, 1979, Boyd et al., 2011] aims at minimizing a function of two variables that can be separated, with linear constraints. It first constructs the Augmented Lagrangian [Hestenes, 1969], then minimizes it by alternating over the different variables (instead of minimizing jointly like in the Augmented Lagrangian framework) and finally updates the dual variable by dual ascent. Consider some separable function of two variables under linear constraints:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{Z}} f(\mathbf{X},\mathbf{Z}) = g(\mathbf{X}) + h(\mathbf{Z}) \\ &\text{s.t.} \quad \mathbf{X} + \mathbf{Z} = \mathbf{C} \end{aligned} \tag{2.23}$$

where g, h are simple i.e. their proximal can be computed in closed form. The Augmented Lagrangian is the standard (unaugmented) Lagrangian with an added quadratic penalty:

$$l(\mathbf{X},\mathbf{Z},\mathbf{\Gamma}) = g(\mathbf{X}) + h(\mathbf{Z}) + \langle \mathbf{\Gamma}, \mathbf{C} - \mathbf{X} - \mathbf{Z} \rangle + \frac{\mu}{2}\|\mathbf{C} - \mathbf{X} - \mathbf{Z}\|_F^2 \tag{2.24}$$

with $\mathbf{\Gamma}$ the dual variable of Lagrange multipliers. Then, it alternates minimization over the two variables. Over $\mathbf{X}$:

$$\begin{aligned}\arg\min_{\mathbf{X}} l(\mathbf{X}, \mathbf{Z}, \mathbf{\Gamma}) &= \arg\min_{\mathbf{X}} g(\mathbf{X}) + \langle \mathbf{\Gamma}, \mathbf{C} - \mathbf{X} - \mathbf{Z} \rangle + \frac{\mu}{2} \|\mathbf{C} - \mathbf{X} - \mathbf{Z}\|_F^2 \\ &= \arg\min_{\mathbf{X}} g(\mathbf{X}) + \frac{\mu}{2} \left\| \mathbf{C} - \mathbf{X} - \mathbf{Z} + \frac{\mathbf{\Gamma}}{\mu} \right\|_F^2 \\ &= \operatorname{prox}_{\frac{1}{\mu} g} \left(\mathbf{C} - \mathbf{Z} + \frac{\mathbf{\Gamma}}{\mu} \right)\end{aligned} \tag{2.25}$$

Similarly for $\mathbf{Z}$:

$$\arg\min_{\mathbf{Z}} l(\mathbf{X}, \mathbf{Z}, \mathbf{\Gamma}) = \operatorname{prox}_{\frac{1}{\mu} h} \left(\mathbf{C} - \mathbf{X} + \frac{\mathbf{\Gamma}}{\mu} \right) \tag{2.26}$$

Finally, the dual update is a so-called dual ascent step which stems from optimality conditions. Indeed, consider optimal variables $(\mathbf{X}^\star, \mathbf{Z}^\star, \mathbf{\Gamma}^\star)$. Then, optimality conditions on the standard Lagrangian state that:

$$\partial g(\mathbf{X}^\star) - \mathbf{\Gamma}^\star = \mathbf{0} \tag{2.27}$$

Then, as $(\mathbf{X}^\star, \mathbf{Z}^\star)$ minimize the Augmented Lagrangian, optimality conditions on the Augmented Lagrangian give:

$$\partial g(\mathbf{X}^\star) - (\mathbf{\Gamma} + \mu(\mathbf{C} - \mathbf{X}^\star - \mathbf{Z}^\star)) = \mathbf{0} \tag{2.28}$$

Leading to the choice:

$$\mathbf{\Gamma}^\star = \mathbf{\Gamma} + \mu(\mathbf{C} - \mathbf{X}^\star - \mathbf{Z}^\star) \tag{2.29}$$

makes the iterate dual-feasible (i.e. respect the standard Lagrangian optimality conditions). Note that deriving w.r.t. $\mathbf{Z}$ instead of $\mathbf{X}$ in this derivation would yield the same result. ADMM converges sublinearly at rate $\mathcal{O}(1/k)$ for convex functions. The method gained popularity in the last decade thanks to its distributed optimization possibilities for large scale problems as well as its fast convergence rate in practice.

A variable under consideration may be under the action of a composition of a generic convex function and a linear operator:

$$\min_{\mathbf{X}} \quad g(\mathbf{X}) + h(\mathbf{A}\mathbf{X}) \tag{2.30}$$

The proximal can't be computed if $\mathbf{A}$ is not orthogonal thus PGD cannot be used. In order to use ADMM, the problem above can be decoupled using an auxiliary variable $\mathbf{Y}$:

$$\begin{aligned}\min_{\mathbf{X},\mathbf{Y},\mathbf{Z}} \quad & g(\mathbf{X}) + h(\mathbf{Z}) \\ \text{s.t.} \quad & \mathbf{Z} = \mathbf{A}\mathbf{Y} \\ & \mathbf{Y} = \mathbf{X}\end{aligned} \tag{2.31}$$

2.2.3 . Chambolle-Pock

The Chambolle-Pock method [Chambolle and Pock, 2011] is a primal-dual method i.e. considering a saddle-point formulation. It is an alternative method which, contrary to ADMM, can handle a linear operator $\mathbf{A}$ that is not orthogonal (thus not the identity) without further decoupling. It was first applied to total variation image denoising.

Definition 4. *We denote h^* the convex (Legendre-Fenchel) conjugate of a function h acting on some vector space, defined as:*

$$h^*(\mathbf{y}) = \sup_{\mathbf{x}} \{\langle \mathbf{x}, \mathbf{y} \rangle - h(\mathbf{x})\} \tag{2.32}$$

Consider the primal problem:

$$\min_{\mathbf{x}} g(\mathbf{x}) + h(\mathbf{A}\mathbf{x}) \tag{2.33}$$

where h, g are convex functions taking values in a normed space. Its Fenchel dual problem is:

$$\max_{\mathbf{y}} -h^*(\mathbf{y}) - g^*(-\mathbf{A}\mathbf{y}) \tag{2.34}$$

In between them is the primal-dual saddle point formulation:

$$\max_{\mathbf{y}} \min_{\mathbf{x}} g(\mathbf{x}) - h^*(\mathbf{y}) + \langle \mathbf{A}\mathbf{x}, \mathbf{y} \rangle \tag{2.35}$$

We may separate the optimization of the two variables and alternate between them. The addition of a trust region yields the so-called Arrow–Hurwicz method. The original version [Uzawa, 1958] does not consider proximals (they weren't discovered yet...) and uses the Lagrangian saddle-point problem. The method arises naturally from using gradient descent/ascent on the primal/dual variables. Anyhow, we have:

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} g(\mathbf{x}) + \left\langle \mathbf{x}, \mathbf{A}^H \mathbf{y}^k \right\rangle + \frac{1}{2\tau} \left\| \mathbf{x} - \mathbf{x}^k \right\|_F^2 \\ \mathbf{y}^{k+1} &= \arg\max_{\mathbf{y}} -h^*(\mathbf{y}) + \left\langle \mathbf{A}\mathbf{x}^k, \mathbf{y} \right\rangle - \frac{1}{2\sigma} \left\| \mathbf{y} - \mathbf{y}^k \right\|_F^2 \end{aligned} \tag{2.36}$$

where τ, σ can be considered as stepsizes. Adding an extrapolation step with parameter θ and rearranging the problem to reveal the proximals gives:

$$\begin{aligned} \mathbf{y}^{k+1} &= \operatorname{prox}_{\sigma h^*}(\mathbf{y}^k + \sigma \mathbf{A}\hat{\mathbf{x}}^k) \\ \mathbf{x}^{k+1} &= \operatorname{prox}_{\tau g}(\mathbf{x}^k - \tau \mathbf{A}^H \mathbf{y}^{k+1}) \\ \hat{\mathbf{x}}^{k+1} &= \mathbf{x}^{k+1} + \theta(\mathbf{x}^{k+1} - \mathbf{x}^k) \end{aligned} \tag{2.37}$$

with convergence proved in [Chambolle and Pock, 2011] for $\theta = 1$ and $\tau\sigma\|\mathbf{A}\|_2^2 \leq 1$ with rate $\mathcal{O}(1/k)$ in the decrease of the primal dual gap. Note that primal-dual Fenchel optimality conditions state that for optimal $(\mathbf{x}^\star, \mathbf{y}^\star)$:

$$\begin{aligned} \mathbf{A}\mathbf{x}^\star &\in \partial h^*(\mathbf{y}^\star) \\ -\mathbf{A}^H \mathbf{y}^\star &\in \partial g(\mathbf{x}^\star) \end{aligned} \tag{2.38}$$

where ∂ is the subdifferential operator. Knowing those subgradients highlights the method being a primal-dual (accelerated) hybrid gradient method. Note that the proximal of the dual of some function is readily computed from the proximal of the original function via the Moreau decomposition.

Proposition 4. *The Moreau decomposition in general form states that:*

$$\mathbf{x} = \operatorname{prox}_{\lambda h}(\mathbf{x}) + \lambda \operatorname{prox}_{h^*/\lambda}(\mathbf{x}/\lambda) \tag{2.39}$$

2.2.4 . Riemannian optimization

Basics on an embedded manifold

It may happen that the variable is constrained to some space called a Riemannian manifold $\mathcal{M}$:

$$\min_{\mathbf{X} \in \mathcal{M}} f(\mathbf{X}) \tag{2.40}$$

Such manifolds are ones on which we may use calculus and optimization procedures (for example via specific gradient methods). Later on, we will notably work on the manifold of fixed rank matrices of some fixed dimension. We shortly present some notions of Riemannian optimization on a embedded manifold, although we refer to [Boumal, 2023] for a recent and complete reference.

Definition 5. *A smooth (differentiable) manifold $\mathcal{M}$ is a locally diffeomorphic space to a vector space i.e. there exists an invertible function that maps one differentiable manifold to another such that both the function and its inverse are continuously differentiable. Consider $\mathbb{R}^d$ such that d is the dimension of the manifold. In other words, for all $\mathbf{X} \in \mathcal{M}$ there exists a neighbourhood of $\mathbf{X}$ denoted $\mathcal{U}_{\mathbf{X}} \subset \mathcal{M}$ and a diffeomophism $\phi_{\mathbf{X}} : \mathcal{U}_{\mathbf{X}} \to \mathbb{R}^d$.*

Then, we can use some calculus tools, such as the directional derivative.

Definition 6. *Let $f : \mathbb{R}^{m \times n} \to \mathbb{R}^{p \times q}$ be differentiable at $\mathbf{X} \in \mathbb{R}^{m \times n}$.*

Consider a linear application (in $\boldsymbol{\xi}$) : $\mathrm{D}\, f(\mathbf{X}) : \mathbb{R}^{m \times n} \to \mathbb{R}^{p \times q}$ such that:

$$\lim_{\|\boldsymbol{\xi}\| \to 0} \frac{\|f(\mathbf{X} + \boldsymbol{\xi}) - f(\mathbf{X}) - \mathrm{D}\, f(\mathbf{X})[\boldsymbol{\xi}]\|}{\|\boldsymbol{\xi}\|} = 0 \tag{2.41}$$

Then if $\mathrm{D}\, f(\mathbf{X})$ exists, it is unique and called the directional derivative of f.

Equivalently:

$$\mathrm{D}\, f(\mathbf{X})[\boldsymbol{\xi}] = f(\mathbf{X} + \boldsymbol{\xi}) - f(\mathbf{X}) + o(\|\boldsymbol{\xi}\|) \tag{2.42}$$

which allows for its derivation in many cases.

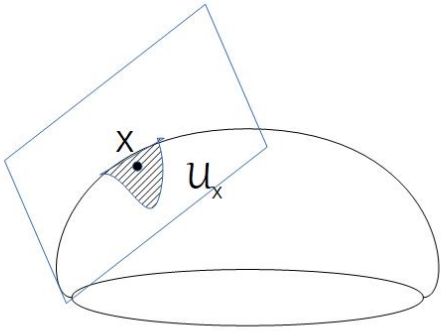


Figure 2.2: Diffeomorphic neighbourhood

On differentiable manifolds, it is practical to work with curves. Denote as γ such a curve mapping from an interval of the real line to the manifold at hand. Indeed, composing a function f defined on the whole Euclidean space with the curve mapping onto the manifold (i.e. $f \circ \gamma$) allows us to work in a straightforward manner on the manifold.

Definition 7. *The tangent space $T_{\mathbf{X}}\mathcal{M}$ of $\mathbf{X} \in \mathcal{M}$ is defined as $T_{\mathbf{X}}\mathcal{M} = \{\gamma'(0) \,|\, \gamma : \mathbb{R} \to \mathcal{M},$ differentiable$, \gamma(0) = \mathbf{X}\}$ and $dim(T_{\mathbf{X}}\mathcal{M}) = dim(\mathcal{M}) = d$*

Then, $\mathbf{X}^*$ is a critical point of f if $\forall \boldsymbol{\xi} \in T_x\mathcal{M}$

$$\mathrm{D}\, f(\mathbf{X}^*)[\boldsymbol{\xi}] = \mathbf{0} \tag{2.43}$$

Definition 8. *A Riemannian metric $\langle \cdot, \cdot \rangle_{\mathbf{X}}$ at $\mathbf{X} \in \mathcal{M}$ is a map of $T_{\mathbf{X}}\mathcal{M} \times T_{\mathbf{X}}\mathcal{M} \to \mathbb{R}$: bilinear, symetric, positive definite (with $\langle \boldsymbol{\xi}, \boldsymbol{\xi} \rangle_{\mathbf{X}} \geq 0$ and equality only at $\boldsymbol{\xi} = 0_{\mathbf{X}}$).*

Equipping a differentiable manifold with a metric makes it into a Riemannian manifold which allows the introduction of geometric notions such as a gradient.

Definition 9. *The Riemannian gradient of f at $\mathbf{X} \in \mathcal{M}$ is the unique element of $T_{\mathbf{X}}\mathcal{M}$ such that $\forall \boldsymbol{\xi} \in T_{\mathbf{X}}\mathcal{M}$*

$$\langle \mathrm{grad}\, f(\mathbf{X}), \boldsymbol{\xi} \rangle_{\mathbf{X}} = \mathrm{D}\, f(\mathbf{X})[\boldsymbol{\xi}] \tag{2.44}$$

Then $\mathbf{X}^* \in \mathcal{M}$ being a critical point of f implies that $\mathrm{grad}\, f(\mathbf{X}^*) = \mathbf{0}$. In most cases, manifolds are submanifolds embedded in a euclidean space. Let $\overline{\mathcal{M}}$ be a Riemannian manifold with metric $\langle \cdot, \cdot \rangle_{\overline{\mathcal{M}}}$.

Definition 10. *A submanifold $\mathcal{M}$ of $\overline{\mathcal{M}}$ is a space included in $\overline{\mathcal{M}}$ such that:*

$$\mathcal{M} = \{x \in \overline{\mathcal{M}} : \varphi(x) = 0\} \tag{2.45}$$

where $\varphi : \overline{\mathcal{M}} \to \mathcal{M}$ is a smooth map and $\mathcal{M} = \varphi^{-1}(\{0\})$.

When considering a manifold embedded in another one, e.g. an Euclidean space, the Riemannian gradient can be defined the orthogonal projection $P^t_{\mathbf{X}}$ of the embedding Euclidean gradient to the tangent space (which is a subspace of the embedding space):

$$\operatorname{grad} f(\mathbf{X}) = P^t_{\mathbf{X}}(\nabla f(\mathbf{X})) \tag{2.46}$$

with ∇f denoting the *Euclidean* gradient of f. The final tool is the retraction which maps from tangent space back to the manifold.

Definition 11. *$R_{\mathbf{X}} : T_{\mathbf{X}}\mathcal{M} \to \mathcal{M}$ is a retraction if:*
1) $R_{\mathbf{X}}(\mathbf{0}_{\mathbf{X}}) = \mathbf{X}$
2) $DR_{\mathbf{X}}(\mathbf{0}_{\mathbf{X}})[\boldsymbol{\eta}] = \boldsymbol{\eta}$

Then, the Riemannian Gradient Descent (RGD) has j^{th} iteration:

$$\mathbf{X}_{j+1} = \mathrm{R}_{\mathbf{X}_j}(-\alpha_j P^t_{\mathbf{X}_j}(\nabla f(\mathbf{X}_j))) \tag{2.47}$$

with α_k a step size found by line-search. This is illustrated in Figure 2.3.

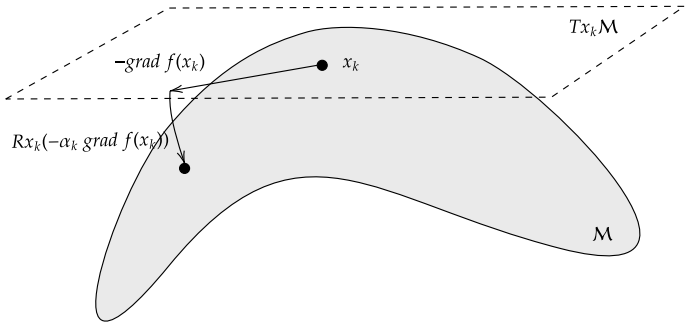


Figure 2.3: RGD

It can be shown that RGD converges to critical points with the main assumption being equivalent to a Lipschitz-type condition on the objective function, i.e. the second-order term of its Taylor expansion is bounded. Moreover, it has a (sublinear) convergence rate of $\mathcal{O}(1/\sqrt{k})$ in terms of the norm of the (Riemannian) gradient of the iterates, although faster convergence (linear) can happen locally, typically when we get in the neighbourhood of a minimum.

Extension to a quotient manifold

Furthemore, we may consider quotient manifolds as the parametrization may be subject to some invariances. Let $\overline{\mathcal{M}}$ be a Riemannian manifold. If there exist an equivalence relation $\sim$ on $\overline{\mathcal{M}}$ verifying, for all X, Y and $Z \in \overline{\mathcal{M}}$:

- $X \sim X$ (reflexivity)
- $X \sim Y$ et $Y \sim Z$ alors $X \sim Z$ (transitivity)
- $X \sim Y \iff Y \sim X$ (symmetry)

then $\mathcal{M} = \overline{\mathcal{M}}/\sim = \{\pi(X) : X \in \overline{\mathcal{M}}\}$ is a quotient manifold where $\pi(X)$ is the equivalent class defined by $\pi(X) = \{Y \in \overline{\mathcal{M}} : Y \sim X\}$. Working in the abstract space defined by the quotient manifold is complicated. We may work in the total (non-quotient) space, which is more practical. However, it necessitates a further space to be introduced. Indeed, the horizontal space is the 'interesting' part of the tangent space for quotient manifolds. There is a one-to-one correspondence between abstract tangent vectors and concrete horizontal vectors so that horizontal vectors may be taken as representations for the underlying abstract tangent vectors. The remaining part is the vertical space. Their direct sum forms the whole tangent space. This is illustrated in Figure 2.4.

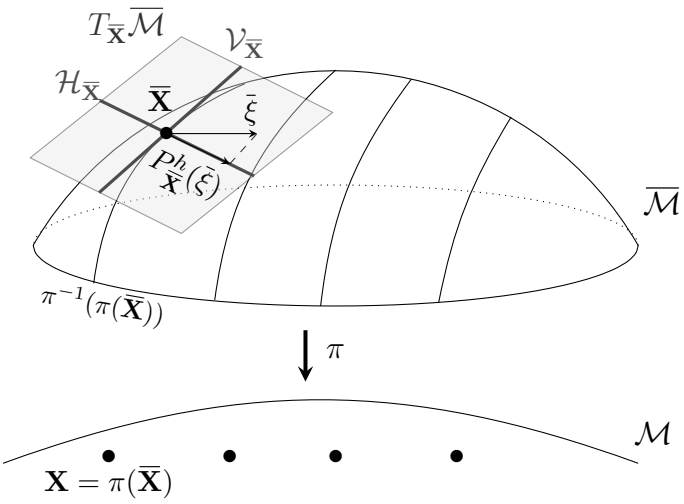


Figure 2.4: A generic quotient manifold $\mathcal{M}$ embedded in its total space $\bar{\mathcal{M}}$ and the decomposition of the tangent space $T_{\overline{\mathbf{X}}}$ at a point $\overline{\mathbf{X}}$ in the direction $\bar{\xi}$ in the horizontal $\mathcal{H}_{\overline{\mathbf{X}}}$ and vertical spaces $\mathcal{V}_{\overline{\mathbf{X}}}$.

RGD for a quotient manifold has j^{th} iteration:

$$\mathbf{X}_{j+1} = \mathrm{R}_{\mathbf{X}_j}(-\alpha_j P^h_{\mathbf{X}_j}(P^t_{\mathbf{X}_j}(\nabla f(\mathbf{X}_j)))) \tag{2.48}$$

with ∇f denoting the *Euclidean* gradient of f and α_k a step size found by line-search. $P^t_{\mathbf{X}_j}$ is the projection from ambient space to tangent space while $P^h_{\mathbf{X}}$ is the projection from the tangent space to the horizontal space and $\mathrm{R}_{\mathbf{X}}$ denotes the retraction of a horizontal vector to the manifold at the point $\mathbf{X}$.

This concludes this chapter of introduction of some notions of advanced optimization. We are ready to delve into the main subject, the methods we developed for TWRI using them.

3 - Inversion via decomposition methods

Contents

We presented a two-step method in the first chapter, which can be found in [Amin and Ahmad, 2013] and we hereby denote SRCS (Sparse Recovery - Compressed Sensing). It considers a vectorized model where the total signal model is created by stacking the measurements at the N radar positions in a long composite vector. Then, it consists of a sequential method in two steps as we: *a*) filter the front wall (for example via methods using the SVD [Tivive et al., 2011, Tivive et al., 2015]), *b*) recover the target positions via sparse reconstruction (e.g. a LASSO-like method).

Recent works [Tang et al., 2016, Tang et al., 2020] suggest that a parallel recovery of both components can improve performances, although they do not consider robustness to multipaths or complex noise cases. Indeed, we may consider a one-step method where we filter the wall and achieve target detection in parallel. This can be done through a low rank plus sparse decomposition of the data which was notably developed in the framework of Robust PCA (RPCA) [Candès et al., 2011, Chandrasekaran et al., 2011]. In this chapter, we present a new method of low rank and sparse decomposition for TWRI detection. The results were published in [Brehier et al., 2022a].

3.1 . Refresher on RPCA

The framework of RPCA [Candès et al., 2011] searchs for a decomposition of a matrix $\mathbf{Y}$ as a sum of a low rank matrix $\mathbf{L}$ and a sparse matrix $\mathbf{S}$. The low rank matrix represent the low dimensional subspace where the data points

lie, except for some outliers which are accounted for in the sparse matrix. In [Mardani et al., 2013], it was extended to a setting with a compressing operator acting on the sparse component. In our case, the role of the two components is reversed: the outlier matrix is not the one we will discard but the one we are interested while the low rank component contains the wall clutter to be discarded.

The solution of RPCA is commonly achieved by considering a convex relaxation of the problem called Principal Component Pursuit (PCP):

$$\begin{aligned} \min_{\mathbf{L},\mathbf{S}} \quad & \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \\ \text{s.t.} \quad & \mathbf{Y} = \mathbf{L} + \mathbf{S} \end{aligned} \tag{3.1}$$

where $\|\cdot\|_*$ is the nuclear norm (the sum of singular values) and $\|\cdot\|_1$ the ℓ_1 norm (the sum of the entries absolute values). Those two norms are known to be the convex envelope of the rank and cardinality (i.e. ℓ_0 norm) of a matrix (restricted to the unit dual norm ball [Fazel, 2002]).

Assuming the above model and under suitable conditions, it was shown that RPCA can accomplish the recovery of the true components [Candès et al., 2011, Chandrasekaran et al., 2011]. Firstly, the true components of the decomposition should be low-rank and sparse enough. Secondly, we should be concerned with the identifiability of the two components: the low rank term should not be sparse and vice-versa. We can mention the so-called incoherence property that the low rank component should verify, which asserts that its singular vectors are reasonably spread out. The sparse component may be assumed to have its support chosen uniformly at random. Although such recovery results have been the subject of many works, we will focus on the effective resolution and its application of TWRI. Indeed, a variety of algorithms have been proposed to solve the PCP problem for RPCA. We give a few examples for the reader to get an idea.

3.1.1 . ADMM resolution

RPCA via PCP can be solved through ADMM. The Augmented Lagrangian associated with (3.1) is:

$$l(\mathbf{L},\mathbf{S},\mathbf{\Gamma}) = \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \langle \mathbf{\Gamma}, \mathbf{Y}-\mathbf{L}-\mathbf{S}\rangle + \frac{\mu}{2}\|\mathbf{Y}-\mathbf{L}-\mathbf{S}\|_F^2 \tag{3.2}$$

where $\mathbf{\Gamma}$ is the matrix Lagrange multiplier associated with the equality constraint. Its optimization over $\mathbf{L}, \mathbf{S}, \mathbf{\Gamma}$ gives the updated variables $\mathbf{L}^\star, \mathbf{S}^\star, \mathbf{\Gamma}^\star$.

$\mathbf{L}$ step

For one variable, we have:

$$\begin{aligned} \mathbf{L}^\star = \arg\min_{\mathbf{L}} l(\mathbf{L},\mathbf{S},\mathbf{\Gamma}) &= \arg\min_{\mathbf{L}} \|\mathbf{L}\|_* + \frac{\mu}{2}\left\|\mathbf{Y}-\mathbf{L}-\mathbf{S}+\mu^{-1}\mathbf{\Gamma}\right\|_F^2 \\ &= D_{1/\mu}(\mathbf{Y}-\mathbf{S}+\mu^{-1}\mathbf{\Gamma}) \end{aligned} \tag{3.3}$$

where D is the proximal of the nuclear norm i.e. singular value thresholding.

$\mathbf{S}$ step

And for the other:

$$\begin{aligned}\mathbf{S}^\star = \arg\min_{\mathbf{S}} l(\mathbf{L},\mathbf{S},\mathbf{\Gamma}) &= \arg\min_{\mathbf{S}} \lambda\|\mathbf{S}\|_1 + \frac{\mu}{2}\left\|\mathbf{Y}-\mathbf{L}-\mathbf{S}+\mu^{-1}\mathbf{\Gamma}\right\|_F^2 \\ &= S_{\lambda/\mu}(\mathbf{Y}-\mathbf{L}+\mu^{-1}\mathbf{\Gamma})\end{aligned} \tag{3.4}$$

where S is the proximal of the ℓ_1 norm i.e. soft-thresholding.

$\mathbf{\Gamma}$ step

Lastly, the dual update is a standard ADMM dual ascent step:

$$\mathbf{\Gamma}^\star = \mathbf{\Gamma} + \mu(\mathbf{Y}-\mathbf{L}-\mathbf{S}) \tag{3.5}$$

This is summarized in Algorithm 5.

Algorithm 5 ADMM for RPCA $(\lambda, \mu > 0, \mathbf{Y})$

$\mathbf{S}_0 \leftarrow \mathbf{0}$
$\mathbf{\Gamma}_0 \leftarrow \mathbf{0}$
repeat (for $k = 0, 1, 2, \ldots$):
    $\mathbf{L}_{k+1} = D_{1/\mu}(\mathbf{Y} - \mathbf{S}_k + \mu^{-1}\mathbf{\Gamma}_k)$
    $\mathbf{S}_{k+1} = S_{\lambda/\mu}(\mathbf{Y} - \mathbf{L}_{k+1} + \mu^{-1}\mathbf{\Gamma}_k)$
    $\mathbf{\Gamma}_{k+1} = \mathbf{\Gamma}_k + \mu(\mathbf{Y} - \mathbf{L}_{k+1} - \mathbf{S}_{k+1})$
until stopping criterion is met
$\mathbf{L} \leftarrow \mathbf{L}_k$
$\mathbf{S} \leftarrow \mathbf{S}_k$

3.1.2 . Proximal Gradient Descent resolution

Another possible algorithm for RPCA is the use of the Proximal Gradient Descent method. Indeed, the equality constraint in RPCA (3.1) may be relaxed to obtain:

$$\min_{\mathbf{L},\mathbf{S}} \quad \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \frac{1}{2\mu}\|\mathbf{Y}-\mathbf{L}-\mathbf{S}\|_F^2 \tag{3.6}$$

which recovers the same solution as RPCA via PCP when μ approaches zero. Now, let us consider $\mathbf{Z} = \begin{bmatrix}\mathbf{L}\\ \mathbf{S}\end{bmatrix}$. Then, this can be rewritten as:

$$\min_{\mathbf{Z}} f(\mathbf{Z}) + g(\mathbf{Z}) \tag{3.7}$$

where $f(\mathbf{Z}) = \mu\|\mathbf{L}\|_* + \mu\lambda\|\mathbf{S}\|_1$ and $g(\mathbf{Z}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{L} - \mathbf{S}\|_F^2 = \frac{1}{2}\|\mathbf{Y} - \mathbf{K}\mathbf{Z}\|_F^2$ with $\mathbf{K} = [\mathbf{I}, \mathbf{I}]$ and g being differentiable. Indeed:

$$\nabla g(\mathbf{Z}) = \mathbf{K}^H(\mathbf{Y} - \mathbf{K}\mathbf{Z}) = -\begin{bmatrix}\mathbf{Y} - \mathbf{L} - \mathbf{S} \\ \mathbf{Y} - \mathbf{L} - \mathbf{S}\end{bmatrix} \tag{3.8}$$

Then, one iteration of PGD is:

$$\text{prox}_{tf}\left(\mathbf{Z} - t\nabla g(\mathbf{Z})\right) \tag{3.9}$$

with $t > 0$ a step-size. Since f consists in a separable sum across $\mathbf{L}$ and $\mathbf{S}$, its proximal can be written as:

$$\text{prox}_{tf}(\mathbf{Z}) = \begin{bmatrix}D_{t\mu}(\mathbf{L}) \\ S_{\lambda\mu t}(\mathbf{S})\end{bmatrix} \tag{3.10}$$

Thus, the PGD iteration above can be decomposed as:

$$\begin{aligned}\mathbf{L}^\star &= D_{t\mu}(\mathbf{L} + t(\mathbf{Y} - \mathbf{L} - \mathbf{S})) \\ \mathbf{S}^\star &= S_{\lambda\mu t}(\mathbf{S} + t(\mathbf{Y} - \mathbf{L} - \mathbf{S}))\end{aligned} \tag{3.11}$$

Note that in this case, as the Hessian of g is known: $H_g(\mathbf{Z}) = \mathbf{K}^H\mathbf{K}$ (with only eigenvalue 2) so that we may use $t = \frac{1}{2}$. This is summarized in Algorithm 6. We may also use the accelarated version, APGD, as summarized in Algorithm 7.

Algorithm 6 PGD for RPCA $(\lambda, \mu > 0, \mathbf{Y})$

$t \leftarrow \frac{1}{2}$
$\mathbf{S}_0 \leftarrow \mathbf{0}$
$\mathbf{L}_0 \leftarrow \mathbf{0}$
repeat (for $k = 0, 1, 2, \ldots$):
    $\mathbf{L}_{k+1} = D_{\mu t}(\mathbf{L}_k + t((\mathbf{Y} - \mathbf{L}_k - \mathbf{S}_k)))$
    $\mathbf{S}_{k+1} = S_{\lambda\mu t}(\mathbf{S}_k + t((\mathbf{Y} - \mathbf{L}_k - \mathbf{S}_k)))$
until stopping criterion is met
$\mathbf{L} \leftarrow \mathbf{L}_k$
$\mathbf{S} \leftarrow \mathbf{S}_k$

3.1.3 . Chambolle-Pock resolution

Another method (the list not being exhaustive) for RPCA is to use the Chambolle-Pock (CP) method [Chambolle and Pock, 2011] which is a primal dual algorithm. Indeed, consider the standard Lagrangian formulation of the RPCA problem:

$$\begin{aligned}&\min_{\mathbf{L},\mathbf{S}}\max_{\mathbf{\Gamma}} \quad \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \langle\mathbf{\Gamma}, \mathbf{Y} - \mathbf{L} - \mathbf{S}\rangle \\ =&\min_{\mathbf{Z}}\max_{\mathbf{\Gamma}} \quad f(\mathbf{Z}) + \langle\mathbf{K}\mathbf{Z}, \mathbf{\Gamma}\rangle - h^*(\mathbf{\Gamma})\end{aligned} \tag{3.12}$$

Algorithm 7 APGD for RPCA $(\lambda, \mu > 0, \mathbf{Y})$

$t \leftarrow \frac{1}{2}$
$\theta_0 = 0$
$\mathbf{L}_0, \hat{\mathbf{L}}_0 \leftarrow \mathbf{0}$
$\mathbf{S}_0, \hat{\mathbf{S}}_0 \leftarrow \mathbf{0}$
repeat (for $k = 0, 1, 2, \ldots$):
    $\hat{\mathbf{L}}_{k+1} = \mathbf{L}_k + \theta_k(\mathbf{L}_k - \mathbf{L}_{k-1})$
    $\hat{\mathbf{S}}_{k+1} = \mathbf{S}_k + \theta_k(\mathbf{S}_k - \mathbf{S}_{k-1})$
    $\mathbf{L}_{k+1} = D_{\mu t}(\hat{\mathbf{L}}_{k+1} + t((\mathbf{Y} - \hat{\mathbf{L}}_{k+1} - \hat{\mathbf{S}}_{k+1})))$
    $\mathbf{S}_{k+1} = S_{\lambda\mu t}(\hat{\mathbf{S}}_{k+1} + t((\mathbf{Y} - \hat{\mathbf{L}}_{k+1} - \hat{\mathbf{S}}_{k+1})))$
    $\theta_{k+1} = \frac{k+1}{(k+1)+3}$
until stopping criterion is met
$\mathbf{L} \leftarrow \mathbf{L}_k$
$\mathbf{S} \leftarrow \mathbf{S}_k$

where f and $\mathbf{K}$ are as defined in the previous section and $h^*(\mathbf{\Gamma}) = \langle \mathbf{Y}, \mathbf{\Gamma} \rangle$ being the convex conjugate of some function h. The problem is suited for the Chambolle-Pock method. We get, at iteration $k+1$:

$$\begin{aligned} \mathbf{\Gamma}_{k+1} &= \text{prox}_{\sigma h^*}(\mathbf{\Gamma}_k + \sigma \mathbf{K} \hat{\mathbf{Z}}_k) \\ \mathbf{Z}_{k+1} &= \text{prox}_{\tau f}(\mathbf{Z}_k - \tau \mathbf{K}^H \mathbf{\Gamma}_{k+1}) \\ \hat{\mathbf{Z}}_{k+1} &= \mathbf{Z}_{k+1} + \theta(\mathbf{Z}_{k+1} - \mathbf{Z}_k) \end{aligned} \tag{3.13}$$

Note that we need the proximal of h^* which is readily found as:

$$\begin{aligned} \text{prox}_{\lambda h^*}(\mathbf{V}) &= \arg\min_{\mathbf{X}} \frac{1}{2\lambda} \|\mathbf{X} - \mathbf{V}\|_F^2 + \langle \mathbf{Y}, \mathbf{X} \rangle \\ &= \arg\min_{\mathbf{X}} \|\mathbf{X} - (\mathbf{V} - \lambda \mathbf{Y})\|_F^2 \\ &= \mathbf{V} - \lambda \mathbf{Y} \end{aligned} \tag{3.14}$$

We may again separate the proximal over the composite variable $\mathbf{Z}$ in its components $\mathbf{L}, \mathbf{S}$ giving the final result. Notice that we can use the convergence properties of the method to choose $\theta = 1$ and $\tau, \sigma > 0$ such that $\tau\sigma \leq \|\mathbf{K}\|_2^2 = 2$. The method is summarized in Algorithm 8.

3.1.4 . Simulation results

We compare the different methods quickly to get an idea of their difference. More importantly, we will look at their convergence speed. We create a rank k matrix $\mathbf{L} \in \mathbb{C}^{100 \times 80}$ by constructing it from its rank factorization:

$$\mathbf{L} = \mathbf{L}_L \mathbf{L}_R^T \tag{3.15}$$

where $\mathbf{L}_L \in \mathbb{C}^{100 \times 4}$ and $\mathbf{L}_R \in \mathbb{C}^{80 \times 4}$ are randomly sampled from a standard normal distribution. The sparse component $\mathbf{S}$ is created by random choice of

Algorithm 8 CP for RPCA $(\lambda, \mu, \theta, \tau, \sigma > 0, \mathbf{Y})$

$\mathbf{L}_0, \hat{\mathbf{L}}_0 \leftarrow \mathbf{0}$
$\mathbf{S}_0, \hat{\mathbf{S}}_0 \leftarrow \mathbf{0}$
$\mathbf{\Gamma}_0 \leftarrow \mathbf{0}$
repeat (for $k = 0, 1, 2, \ldots$):
    $\mathbf{\Gamma}_{k+1} = \mathbf{\Gamma}_k + \sigma(\hat{\mathbf{L}}_k + \hat{\mathbf{S}}_k - \mathbf{Y})$
    $\mathbf{L}_{k+1} = D_\tau(\mathbf{L}_k - \tau\mathbf{\Gamma}_{k+1})$
    $\mathbf{S}_{k+1} = S_{\lambda\tau}(\mathbf{S}_k - \tau\mathbf{\Gamma}_{k+1})$
    $\hat{\mathbf{L}}_{k+1} \leftarrow \mathbf{L}_{k+1} + \theta(\mathbf{L}_{k+1} - \mathbf{L}_k)$
    $\hat{\mathbf{S}}_{k+1} \leftarrow \mathbf{S}_{k+1} + \theta(\mathbf{S}_{k+1} - \mathbf{S}_k)$
until stopping criterion is met
$\mathbf{L} \leftarrow \mathbf{L}_k$
$\mathbf{S} \leftarrow \mathbf{S}_k$

the non-zero matrix entries. Those components are shown in Figure 3.1. All

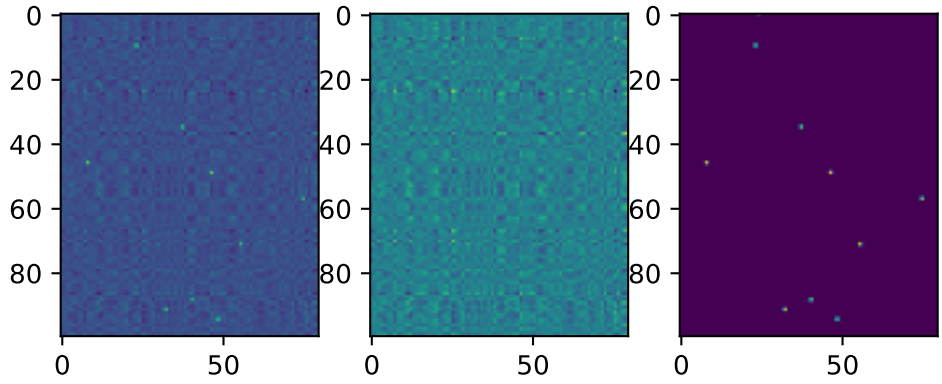


Figure 3.1: Data matrix $\mathbf{Y}$ (left) decomposed via ADMM in $\mathbf{L}$ (middle) plus $\mathbf{S}$ (right)

methods do manage to recover the true components as expected from the recovery theorems of [Candès et al., 2011, Chandrasekaran et al., 2011]. We can monitor the convergence at each iteration (they are comparable across algorithms) by the distance to the true components that we know as we generated them:

$$\frac{\|\mathbf{L} - \mathbf{L}_{alg}\|_F}{\|\mathbf{L}\|_F} + \frac{\|\mathbf{S} - \mathbf{S}_{alg}\|_F}{\|\mathbf{S}\|_F} \tag{3.16}$$

where $\mathbf{L}_{alg}, \mathbf{L}_{alg}$ are the components retrieved by some algorithm at some iteration. The results are shown in Figure 3.2 in log scale. We can see the faster practical convergence of ADMM which motivates its use in the next chapters.

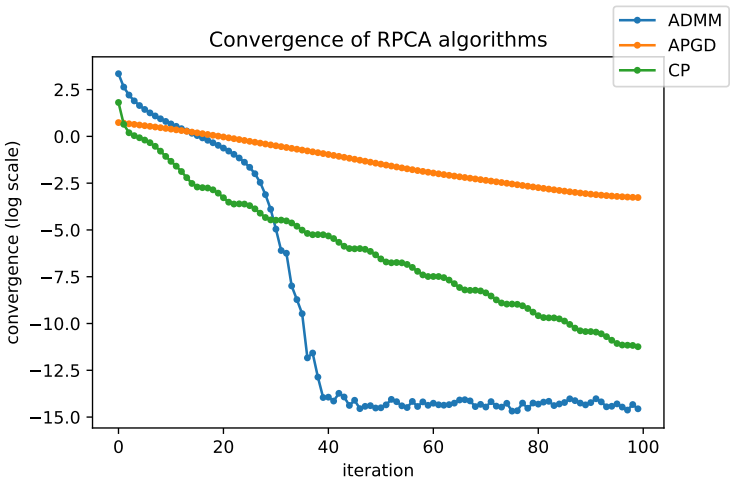


Figure 3.2: Convergence of RPCA algorithms (log scale)

3.2 . A low rank and sparse decomposition for TWRI

3.2.1 . Signal model

Let us reconsider the model from Section 1.4.2. We have:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} = \begin{bmatrix} \mathbf{l} \\ \mathbf{l} \\ \vdots \\ \mathbf{l} \end{bmatrix} + [\boldsymbol{\Psi}^{(1)}, \boldsymbol{\Psi}^{(2)}, \dots, \boldsymbol{\Psi}^{(R)}] \begin{bmatrix} \mathbf{r}^{(1)} \\ \mathbf{r}^{(2)} \\ \vdots \\ \mathbf{r}^{(R)} \end{bmatrix} \tag{3.17}$$

Where:

$$\boldsymbol{\Psi}^{(i)} = \begin{bmatrix} \boldsymbol{\Psi}_1^{(i)} \\ \boldsymbol{\Psi}_2^{(i)} \\ \vdots \\ \boldsymbol{\Psi}_N^{(i)} \end{bmatrix} \tag{3.18}$$

So:

$$\begin{aligned} \mathbf{y}_n &= \mathbf{l} + \underbrace{[\boldsymbol{\Psi}_n^{(1)}, \boldsymbol{\Psi}_n^{(2)}, \dots, \boldsymbol{\Psi}_n^{(R)}]}_{=\boldsymbol{\Psi}_n} \underbrace{\begin{bmatrix} \mathbf{r}^{(1)} \\ \mathbf{r}^{(2)} \\ \vdots \\ \mathbf{r}^{(R)} \end{bmatrix}}_{=\mathbf{r}} \\ \implies \mathbf{y}_n &= \mathbf{l} + \boldsymbol{\Psi}_n \mathbf{r} \end{aligned} \tag{3.19}$$

Again, we can concatenate the observations $\{\mathbf{y}_i\}_{i=1}^N$ in a matrix as:

$$\underbrace{[\mathbf{y}_1 | \dots | \mathbf{y}_N]}_{=\mathbf{Y}} = \underbrace{[\mathbf{l} | \dots | \mathbf{l}]}_{=\mathbf{L}} + \underbrace{[\boldsymbol{\Psi}_1 | \dots | \boldsymbol{\Psi}_N]}_{=\boldsymbol{\Psi}} \underbrace{\begin{bmatrix} \mathbf{r} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{r} \end{bmatrix}}_{=\mathbf{R}} \tag{3.20}$$

In short, this is written:

$$\mathbf{Y} = \mathbf{L} + \boldsymbol{\Psi}\mathbf{R} = \mathbf{L} + \boldsymbol{\Psi}\left(\mathbf{I}_N \otimes \mathbf{r}\right) \tag{3.21}$$

with $\otimes$ denoting the Kronecker product. Note that $\mathbf{Y} \in \mathbb{C}^{M \times N}$, $\mathbf{L} \in \mathbb{C}^{M \times N}$ is a rank-one matrix while $\boldsymbol{\Psi} \in \mathbb{C}^{M \times N_x N_z RN}$ is a dictionnary matrix and $\mathbf{R} \in \mathbb{C}^{N_x N_z RN \times N}$ is a sparse matrix.

3.2.2 . RPCA with dictionary

RPCA with dictionary [Mardani et al., 2013] aims at solving:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{R}} \quad & \|\mathbf{L}\|_* + \lambda \|\mathbf{R}\|_1 \\ \text{s.t.} \quad & \mathbf{Y} = \mathbf{L} + \mathbf{\Psi}\mathbf{R} \end{aligned} \tag{3.22}$$

where $\mathbf{\Psi}$ is a dictionary (also called compression matrix). Notice that we cannot use a structured sparsity constraint as we ignore the Kronecker structure. As for RPCA, there exist exact recovery conditions. This fits our problems stated in (3.21) if we denote $\mathbf{R} = \mathbf{I}_N \otimes \mathbf{r}$, thus neglecting the inner structure in this first approach. While we could directly tackle the above problem via PGD over the concatenation of the two variables, this yields difficult ℓ_1 norm minimization subproblems as the dictionary $\mathbf{\Psi}$ couples the entries of the two matrix variables. Another option consists in decoupling the sparse matrix from the low rank one. A common technique is to introduce an auxiliary variable , and formulate the following optimization problem:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{R}} \quad & \|\mathbf{L}\|_* + \lambda \|\mathbf{R}\|_1 \\ \text{s.t.} \quad & \mathbf{Y} = \mathbf{L} + \mathbf{\Psi}\mathbf{S} \\ & \mathbf{S} = \mathbf{R} \end{aligned} \tag{3.23}$$

Then, the associated Augmented Lagrangian is:

$$\begin{aligned} l(\mathbf{L},\mathbf{R},\mathbf{S},\mathbf{\Gamma},\tilde{\mathbf{\Gamma}}) = \|\mathbf{L}\|_* + \lambda\|\mathbf{R}\|_1 + \langle \mathbf{\Gamma}, \mathbf{Y} - \mathbf{L} - \mathbf{\Psi}\mathbf{S}\rangle + \frac{\mu}{2}\|\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}\mathbf{S}\|_F^2 \\ + \left\langle \tilde{\mathbf{\Gamma}}, \mathbf{S} - \mathbf{R} \right\rangle + \frac{\mu}{2}\|\mathbf{S} - \mathbf{R}\|_F^2 \end{aligned} \tag{3.24}$$

where $\mathbf{\Gamma}, \tilde{\mathbf{\Gamma}}$ are the matrix Lagrange multipliers associated with the (respectively) first and second constraints. We are then in the position to use ADMM, that is optimizing over each variable alternatively. The updates of the first group of primal variables are simple proximals.

$\mathbf{L}$ step

The minimization over $\mathbf{L}$ is a standard procedure in the literature, which reduces to:

$$\mathbf{L}^\star = \arg\min_{\mathbf{L}} l(\mathbf{L},\mathbf{R},\mathbf{S},\mathbf{\Gamma},\tilde{\mathbf{\Gamma}}) = D_{1/\mu}(\mathbf{Y} - \mathbf{\Psi}\mathbf{S} + \mu^{-1}\mathbf{\Gamma}) \tag{3.25}$$

$\mathbf{R}$ step

Similarly for $\mathbf{R}$, the optimization yields:

$$\mathbf{R}^\star = \arg\min_{\mathbf{R}} l(\mathbf{L},\mathbf{R},\mathbf{S},\mathbf{\Gamma},\tilde{\mathbf{\Gamma}}) = S_{\lambda/\mu}(\mathbf{S} + \mu^{-1}\tilde{\mathbf{\Gamma}}) \tag{3.26}$$

$\mathbf{S}$ step

The update of the decoupling variable $\mathbf{S}$ can be derived using the first-order condition for optimality:

$$\nabla_{\mathbf{S}^\star} l(\mathbf{L}, \mathbf{R}, \mathbf{S}^\star, \mathbf{\Gamma}, \tilde{\mathbf{\Gamma}}) = 0 \tag{3.27}$$

$$\implies \mathbf{S}^\star = \mathbf{R} + \left(\mathbf{\Psi}^H\mathbf{\Psi} + \mathbf{I}\right)^{-1}\left[\mathbf{\Psi}^H(\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}\mathbf{R}) - \mu^{-1}(\tilde{\mathbf{\Gamma}} - \mathbf{\Psi}^H\mathbf{\Gamma})\right] \tag{3.28}$$

where we use the matrix inversion identity $(\mathbf{I} + \mathbf{P})^{-1} = \mathbf{I} - (\mathbf{I} + \mathbf{P})^{-1}\mathbf{P}$ to obtain this factorization. However, the inversion of $\left(\mathbf{\Psi}^H\mathbf{\Psi} + \mathbf{I}\right)$ is costly as the dimension of this square matrix is $N_x N_z N$. A way to overcome this obstacle is through the Woodbury matrix inversion lemma. Indeed:

$$\begin{aligned} \mathbf{\Psi} &\overset{\text{SVD}}{=} \mathbf{U_d}\mathbf{S_d}\mathbf{V_d}^H \\ \implies \mathbf{\Psi}^H\mathbf{\Psi} &= \mathbf{V_d}(\mathbf{S_d})^2\mathbf{V_d}^H \\ \left(\mathbf{\Psi}^H\mathbf{\Psi} + \mathbf{I}\right)^{-1} &\overset{\text{Woodbury}}{=} \mathbf{I} - \mathbf{V_d}\mathbf{\Pi}\mathbf{V_d}^H \end{aligned} \tag{3.29}$$

where $\mathbf{\Pi} = \left((\mathbf{S_d}^2)^{-1} + \mathbf{I}\right)^{-1}$ is a diagonal matrix with $(\mathbf{\Pi})_{i,i} = \frac{s_i^2}{1+s_i^2}$, s_i being the i^{th} singular value of $\mathbf{\Psi}$. By developping the expression for the $\mathbf{S}$-update, we have:

$$\begin{aligned} \mathbf{S}^\star = &\left(\mathbf{R} + \mathbf{\Psi}^H(\mathbf{Y} - \mathbf{L}) - \mu^{-1}(\tilde{\mathbf{\Gamma}} - \mathbf{\Psi}^H\mathbf{\Gamma})\right) \\ &- \left(\mathbf{V_d}\mathbf{\Pi}\left(\mathbf{V_d}^H\mathbf{R} + \mathbf{S_d}\mathbf{U_d}^H(\mathbf{Y} - \mathbf{L}) - \mu^{-1}(\mathbf{V_d}^H\tilde{\mathbf{\Gamma}} - \mathbf{S_d}\mathbf{U_d}^H\mathbf{\Gamma})\right)\right) \end{aligned} \tag{3.30}$$

This does not use any square matrix in $N_x N_z N$ which solves our problem. The computation of the SVD of $\mathbf{\Psi}$ can be done once in the initialization and reused for the $\mathbf{S}$-update.

$\mathbf{\Gamma}, \tilde{\mathbf{\Gamma}}$ steps

The dual updates are standard dual ascent steps:

$$\begin{aligned} \arg\min_{\mathbf{\Gamma}} l(\mathbf{L}, \mathbf{R}, \mathbf{S}, \mathbf{\Gamma}, \tilde{\mathbf{\Gamma}}) &= \mathbf{\Gamma} + \mu(\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}\mathbf{S}) \\ \arg\min_{\tilde{\mathbf{\Gamma}}} l(\mathbf{L}, \mathbf{R}, \mathbf{S}, \mathbf{\Gamma}, \tilde{\mathbf{\Gamma}}) &= \tilde{\mathbf{\Gamma}} + \mu(\mathbf{S} - \mathbf{R}) \end{aligned} \tag{3.31}$$

This is summarized in Algorithm 9. Using the same setup as in previous sections, we get the results in Figure 3.3. We cannot achieve a much higher resolution as it is very time-consuming. These simulations show the underwhelming quality of the recovery via this method. Moreover, it is not suitable for multipath exploitation as not structured sparsity may be enforced. Clearly, we need to enforce sparsity on the true sparse vector, not a matrix containing several replicas of it as for RPCA with a dictionary. This motivates the introduction of a finer model.

Algorithm 9 ADMM for RPCA with dictionary $(\lambda, \mu, \mathbf{Y}, \mathbf{\Psi})$

$\mathbf{L}_0, \mathbf{Y}_0 \leftarrow \mathbf{0}_{M \times N}$
$\mathbf{R}_0, \mathbf{S}_0, \tilde{\mathbf{Y}}_0 \leftarrow \mathbf{0}_{N_x N_z N \times N}$
$\mathbf{\Psi} \overset{\text{SVD}}{=} \mathbf{U_d} \mathbf{S_d} \mathbf{V_d}^H$ with $(\mathbf{S_d})_{i,i} = s_i$
$\mathbf{\Pi}$ diagonal s.t. $(\mathbf{\Pi})_{i,i} = \frac{s_i^2}{1+s_i^2}$
repeat (for $k = 0, 1, 2, \ldots$):
    $\mathbf{\Gamma}_{k+1} = \mathbf{\Gamma}_k + \mu(\mathbf{Y} - \mathbf{L}_k - \mathbf{\Psi}\mathbf{S}_k)$
    $\tilde{\mathbf{\Gamma}}_{k+1} = \tilde{\mathbf{\Gamma}}_k + \mu(\mathbf{S}_k - \mathbf{R}_k)$
    $\mathbf{L}_{k+1} = D_{1/\mu}(\mathbf{Y} - \mathbf{\Psi}\mathbf{S}_k + \mu^{-1}\mathbf{\Gamma}_{k+1})$
    $\mathbf{R}_{k+1} = S_{\lambda/\mu}(\mathbf{S}_k + \mu^{-1}\tilde{\mathbf{\Gamma}}_{k+1})$
    $\mathbf{S}_{k+1} = \left(\mathbf{R}_{k+1} + \mathbf{\Psi}^H(\mathbf{Y} - \mathbf{L}_{k+1}) - \mu^{-1}(\tilde{\mathbf{\Gamma}}_{k+1} - \mathbf{\Psi}^H\mathbf{\Gamma}_{k+1})\right) - \left(\mathbf{V_d}\mathbf{\Pi}\left(\mathbf{V_d}^H\mathbf{R}_{k+1} + \mathbf{S_d}\mathbf{U_d}^H(\mathbf{Y} - \mathbf{L}_{k+1}) - \mu^{-1}(\mathbf{V_d}^H\tilde{\mathbf{\Gamma}}_{k+1} - \mathbf{S_d}\mathbf{U_d}^H\mathbf{\Gamma}_{k+1})\right)\right)$
until stopping criterion is met
$\mathbf{L} \leftarrow \mathbf{L}_k$
$\mathbf{R} \leftarrow \mathbf{R}_k$

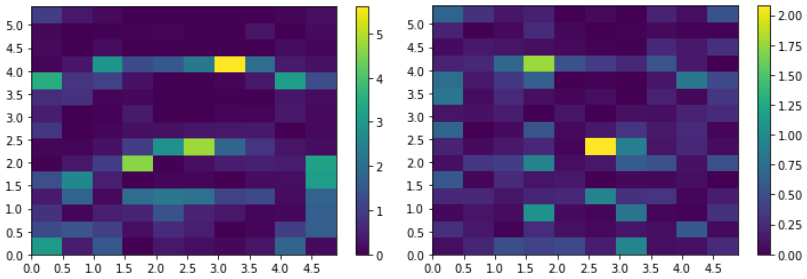


Figure 3.3: Two samples of detection maps of RPCA-dict on a scenario without multipath (targets at $(2, 2)$, $(2.5, 4)$)

3.3 . KRPCA: a specific decomposition for TWRI

3.3.1 . First resolution without decoupling

We can formalize the TWRI problem via the framework of RPCA, with the sparse component appearing in a Kronecker product, a special case of the model in [Mardani et al., 2013]. We may then tailor a resolution via the ADMM framework [Boyd et al., 2011]. Some works of low-rank plus sparse matrix decomposition exist in the context of TWRI [Tang et al., 2016, Tang et al., 2020]. We propose here a matrix model called KRPCA (for Kronecker-structured RPCA):

$$\boxed{\begin{array}{ll}\min\limits_{\mathbf{L},\mathbf{R}} & \|\mathbf{L}\|_* + \lambda\|\mathbf{R}\|_{2,1} \\ \text{s.t.} & \mathbf{Y} = \mathbf{L} + \mathbf{\Psi}\left(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R})\right)\end{array}} \tag{3.32}$$

where we define $\mathbf{R} \triangleq \mathrm{vec}^{-1}(\mathbf{r})$ for ease of notation. A solution can be found via ADMM. The Augmented Lagrangian associated with (3.32) is:

$$\begin{aligned} l(\mathbf{L},\mathbf{R},\mathbf{\Gamma}) = \|\mathbf{L}\|_* + \lambda\|\mathbf{R}\|_{2,1} + \langle\mathbf{\Gamma}, \mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R}))\rangle \\ + \frac{\mu}{2}\|\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R}))\|_F^2 \end{aligned} \tag{3.33}$$

where $\mathbf{\Gamma}$ is the matrix of Lagrange multipliers, λ is the sparsity regularization parameter and μ is the augmented Lagrangian penalty parameter.

$\mathbf{L}$ step

The subproblem for $\mathbf{L}$ is obtained simply by completing the squared norm and recognizing the proximal of the nuclear norm (with threshold λ), denoted D_λ:

$$\mathbf{L}^\star = D_{1/\mu}(\mathbf{Y} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathbf{r}) + \mu^{-1}\mathbf{\Gamma}) \tag{3.34}$$

$\mathbf{R}$ step

For the subproblem for $\mathbf{R}$, we may directly use a PGD step since the objective function of this step is a sum of two convex terms, one derivable and the other one (the $\ell_{2,1}$-norm) on which we may use its proximal T_λ. We can use a fixed step-size which is readily computed via the Hessian of the derivable part of the objective function.

The derivable part whose gradient we need is:

$$\langle\mathbf{\Gamma}, \mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R}))\rangle + \frac{\mu}{2}\|\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R}))\|_F^2 \tag{3.35}$$

whose optimization over $\mathbf{R}$ is equivalent to the one of:

$$\frac{\mu}{2}\left\|\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R})) + \frac{\mathbf{\Gamma}}{\mu}\right\|_F^2 \tag{3.36}$$

i.e. we have completed the squared norm. Then, we may use the fact that the Frobenius norm of a matrix is the same as the ℓ_2 norm on its vectorized version. Consider $\mathbf{y}, \mathbf{l}, \boldsymbol{\gamma}$ the vectorized version of $\mathbf{Y}, \mathbf{L}, \boldsymbol{\Gamma}$. Also remember that $\mathrm{vec}(\boldsymbol{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R}))) = \boldsymbol{\Psi}_A \mathbf{r}$. Then we have:

$$\mathbf{g} = \nabla_{\mathbf{r}} \frac{\mu}{2} \left\| \mathbf{y} - \mathbf{l} - \boldsymbol{\Psi}_A \mathbf{r} + \frac{\boldsymbol{\gamma}}{\mu} \right\|_F^2 = -\mu \boldsymbol{\Psi}_A^H (\mathbf{y} - \mathbf{l} - \boldsymbol{\Psi}_A \mathbf{r} + \frac{\boldsymbol{\gamma}}{\mu}) \tag{3.37}$$

To get the PGD iteration:

$$\mathbf{R}^\star = T_{\lambda t}(\mathrm{vec}^{-1}(\mathrm{vec}(\mathbf{R}) - t\mathbf{g})) \tag{3.38}$$

$\boldsymbol{\Gamma}$ step

Finally, the subproblem for $\boldsymbol{\Gamma}$ is a standard ADMM step of dual ascent. The interesting point is that the step-size is already known: it is the parameter of the Augmented Lagrangian:

$$\boldsymbol{\Gamma} = \boldsymbol{\Gamma} + \mu(\mathbf{Y} - \mathbf{L} - \boldsymbol{\Psi}(\mathbf{I}_N \otimes \mathbf{r})) \tag{3.39}$$

The whole method is summarized in Algorithm 10.

Algorithm 10 KRPCA via PGD

Have: $\{\mathbf{y}_i\}_{i=1}^N, \{\boldsymbol{\Psi}_i\}_{i=1}^N$
Choose: λ, μ
$\mathbf{Y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_N]$
$\boldsymbol{\Psi} \triangleq [\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \dots \boldsymbol{\Psi}_N]$
$\boldsymbol{\Psi}_A \triangleq [\boldsymbol{\Psi}_1^T \boldsymbol{\Psi}_2^T \dots \boldsymbol{\Psi}_N^T]^T$
$\mathbf{P} = {\boldsymbol{\Psi}_A}^H \boldsymbol{\Psi}_A$
$t = 1/\lambda_{\mathsf{max}}(\mu \mathbf{P})$
Initialize: $\mathbf{L}, \mathbf{R}, \boldsymbol{\Gamma}$
repeat
    $\mathbf{L} = D_{1/\mu}(\mathbf{Y} - \boldsymbol{\Psi}(\mathbf{I}_N \otimes \mathbf{r}) + \mu^{-1}\boldsymbol{\Gamma})$
    repeat:
        $\mathbf{g} = \mu \boldsymbol{\Psi}_A^H (\mathrm{vec}(\mathbf{L} - \mathbf{Y} - \frac{\boldsymbol{\Gamma}}{\mu})) + \mathbf{P}\mathbf{r}$
        $\mathbf{R} = T_{\lambda t}(\mathrm{vec}^{-1}(\mathbf{r} - t\mathbf{g}))$
        $\mathbf{r} = \mathrm{vec}\,(\mathbf{R})$
    until stopping criterion is met
    $\boldsymbol{\Gamma} = \boldsymbol{\Gamma} + \mu(\mathbf{Y} - \mathbf{L} - \boldsymbol{\Psi}(\mathbf{I}_N \otimes \mathbf{r}))$
until stopping criterion is met

Analysis

The convergence of the algorithm is ensured by the theory surrounding ADMM. In our case, both functions in the objective functions are proper closed convex functions. Assuming the (non-augmented) Lagrangian has a saddle point, this ADMM algorithm for KRPCA guarantees residual convergence, objective convergence and dual convergence [Boyd et al., 2011].

The computational complexity of the derived algorithm for KRPCA is dependent on some assumptions on the order of the dimensions considered. Assume that $MN > D > M > N$ where $D = N_x N_z R$ is the discretized scene grid size for all multipaths (and recall that M, N are respectively the number of frequencies and radar snapshots). Under those assumptions, the major cost of the overall algorithm is the computation of $\mathbf{P}$ during the initialization, which is of complexity $\mathcal{O}(MND^2)$. We may consider $\mathbf{P}$ as cached beforehand and study the rest of the method. As $\mathbf{\Psi}(\mathbf{I}_N \otimes \mathbf{r}) = \mathrm{vec}^{-1}(\Psi_A \mathbf{r})$, this operation can be seen to be of complexity $\mathcal{O}(MND)$. The PGD step has complexity $\mathcal{O}(KD^2)$. We then conclude that the algorithm (with $\mathbf{P}$ cached) has complexity $\mathcal{O}(KMND)$. In practice, K is relatively small.

3.3.2 . Alternative method via decoupling

We may actually use a similar decoupling strategy for KRPCA as for 'RPCA with dictionary':

$$\boxed{\begin{aligned} \min_{\mathbf{L},\mathbf{R},\mathbf{S}} \quad & \|\mathbf{L}\|_* + \lambda \|\mathbf{R}\|_{2,1} \\ \text{s.t.} \quad & \mathbf{Y} = \mathbf{L} + \mathbf{\Psi}\left(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{S})\right) \\ & \mathbf{S} = \mathbf{R} \end{aligned}} \tag{3.40}$$

The associated Augmented Lagrangian is then:

$$\begin{aligned} l(\mathbf{L}, \mathbf{R}, \mathbf{S}, \mathbf{\Gamma}, \tilde{\mathbf{\Gamma}}) = \|\mathbf{L}\|_* + \lambda \|\mathbf{R}\|_{2,1} + \langle \mathbf{\Gamma}, \mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{S})) \rangle \\ + \frac{\mu}{2} \|\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{S}))\|_F^2 + \left\langle \tilde{\mathbf{\Gamma}}, \mathbf{S} - \mathbf{R} \right\rangle + \frac{\nu}{2} \|\mathbf{S} - \mathbf{R}\|_F^2 \end{aligned} \tag{3.41}$$

$\mathbf{L}$ step

Over $\mathbf{L}$, we have a (well-known by now) proximal update:

$$\mathbf{L} = D_{1/\mu}(\mathbf{Y} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{S})) + \mu^{-1}\mathbf{\Gamma}) \tag{3.42}$$

$\mathbf{R}$ step

Over $\mathbf{R}$, we also have a proximal update:

$$\mathbf{R} = S_{\lambda/\nu}(\mathbf{S} + \nu^{-1}\tilde{\mathbf{\Gamma}}) \tag{3.43}$$

$\mathbf{S}$ step

Over $\mathbf{S}$, we have a sum of quadratic terms which can be solved by first-order optimality conditions. Indeed, this can be more easily handled when noting that the Frobenius inner product and norm are equivalent to working on vectorized variables. With $\mathbf{l}, \mathbf{r}, \mathbf{s}, \boldsymbol{\gamma}, \tilde{\boldsymbol{\gamma}}$ the vectorized variables $\mathbf{L}, \mathbf{R}, \mathbf{S}, \boldsymbol{\Gamma}, \tilde{\boldsymbol{\Gamma}}$, we have:

$$\begin{aligned} l(\mathbf{l}, \mathbf{r}, \mathbf{s}, \boldsymbol{\gamma}, \tilde{\boldsymbol{\gamma}}) = \left\|\mathrm{vec}^{-1}(\mathbf{l})\right\|_* + \lambda\left\|\mathrm{vec}^{-1}(\mathbf{r})\right\|_{2,1} + \langle \boldsymbol{\gamma}, \mathbf{y} - \mathbf{l} - \boldsymbol{\Psi}_A \mathbf{s}\rangle \\ + \frac{\mu}{2}\|\mathbf{y} - \mathbf{l} - \boldsymbol{\Psi}_A \mathbf{s}\|_F^2 + \langle \tilde{\boldsymbol{\gamma}}, \mathbf{s} - \mathbf{r}\rangle + \frac{\nu}{2}\|\mathbf{s} - \mathbf{r}\|_F^2 \end{aligned} \tag{3.44}$$

Then, considering the first-order optimality conditions, this amounts to solving a linear system in $\mathbf{s}$:

$$(\mu \boldsymbol{\Psi}_A^H \boldsymbol{\Psi}_A + \nu \mathbf{I})\mathbf{s} = \mu \boldsymbol{\Psi}_A^H (\mathbf{y} - \mathbf{l} + \mu^{-1}\boldsymbol{\gamma}) + \nu(\mathbf{r} - \nu^{-1}\tilde{\boldsymbol{\gamma}}) \tag{3.45}$$

and unvectorizing to retrieve $\mathbf{S}$.

$\boldsymbol{\Gamma}$,$\tilde{\boldsymbol{\Gamma}}$ step

Finally, the dual variables are updated by the usual dual ascent.

$$\begin{aligned} \boldsymbol{\Gamma} &= \boldsymbol{\Gamma} + \mu(\mathbf{Y} - \mathbf{L} - \boldsymbol{\Psi}(\mathbf{I}_N \otimes \mathbf{r})) \\ \tilde{\boldsymbol{\Gamma}} &= \tilde{\boldsymbol{\Gamma}} + \nu(\mathbf{S} - \mathbf{R}) \end{aligned} \tag{3.46}$$

We summarize this in Algorithm 11.

Algorithm 11 KRPCA via decoupling

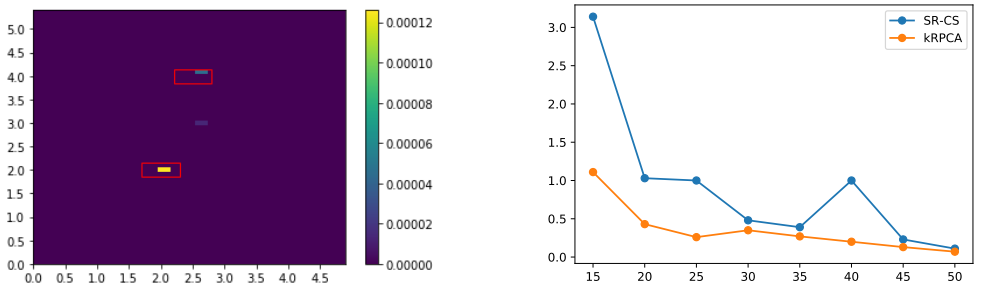
Have: $\{\mathbf{y}_i\}_{i=1}^N, \{\boldsymbol{\Psi}_i\}_{i=1}^N$
Choose: λ, μ
$\mathbf{Y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_N]$
$\boldsymbol{\Psi} \triangleq [\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \dots \boldsymbol{\Psi}_N]$
$\boldsymbol{\Psi}_A \triangleq [\boldsymbol{\Psi}_1^T \boldsymbol{\Psi}_2^T \dots \boldsymbol{\Psi}_N^T]^T$
Initialize: $\mathbf{L}, \mathbf{R}, \mathbf{S}, \boldsymbol{\Gamma}, \tilde{\boldsymbol{\Gamma}}$
repeat
    $\mathbf{L} = D_{1/\mu}(\mathbf{Y} - \boldsymbol{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{S})) + \mu^{-1}\boldsymbol{\Gamma})$
    $\mathbf{R} = T_{\lambda/\nu}(\mathbf{S} + \nu^{-1}\tilde{\boldsymbol{\Gamma}})$
    $\mathbf{S} = \mathrm{vec}^{-1}\left[(\mu \boldsymbol{\Psi}_A^H \boldsymbol{\Psi}_A + \nu \mathbf{I})^{-1}\left[\mu \boldsymbol{\Psi}_A^H (\mathbf{y} - \mathbf{l} + \mu^{-1}\boldsymbol{\gamma}) + \nu(\mathbf{r} - \nu^{-1}\tilde{\boldsymbol{\gamma}})\right]\right]$
    $\boldsymbol{\Gamma} = \boldsymbol{\Gamma} + \mu(\mathbf{Y} - \mathbf{L} - \boldsymbol{\Psi}(\mathbf{I}_N \otimes \mathbf{r}))$
    $\tilde{\boldsymbol{\Gamma}} = \tilde{\boldsymbol{\Gamma}} + \nu(\mathbf{S} - \mathbf{R})$
until stopping criterion is met

In this form, the optimization procedures fits the consensus variant of ADMM, where local variables $(\mathbf{L}, \mathbf{R})$ are tied via a global consensus variable

(S). It still inherits the convergence properties of ADMM. The computational complexity is also identical to the non-decoupled case with the difference that there is no inner loop.

3.3.3 . Some simulation results

We test KRPCA on simulated raytracing data as done previously, with multipath. The resulting detection map is shown in Figure 3.4a. For more representative results, we run a Monte-Carlo simulation over a range of SNR. We use 100 draws at each SNR, and stop each method at iterate k when two iterates are close by, meaning $\left\|\mathbf{r}^k - \mathbf{r}^{k-1}\right\|_F \leq 1\text{e}^{-6}$. This generally amounts to around 20 iterations for both algorithms. The error evaluated is defined as the count of false alarms plus non-detections. To this end, we consider blocks of 2 by 2 pixels to allow for small clusters of pixels and set a detection threshold equal to 10 % of the highest pixel intensity. The hyper-parameters are set to a value manually found to be optimal and remain constant over all draws. Typically, we first tune μ (which controls the wall mitigation) to a level that visibly erases the wall contribution. Then, we tune λ (which controls scene sparsity) in order to clean the remaining image. It would be hard to set it in order to control the probability of false alarm (PFA) as the thresholding value is implicitly set via λ and not explicitly, which usually allows to control the PFA via the noise statistics. We observe in Figure 3.4b that KRPCA performs better than SRCS (the method detailled in Chapter 1), as our error tends to zero while SRCS stays around one. The likely reason is that the ghost target (visible in the previous images) is difficult to erase with SRCS.



(a) Results of KRPCA

(b) Error vs SNR for SRCS and KRPCA

Figure 3.4: Comparison of KRPCA results and Error vs SNR for SRCS and KRPCA

3.4 . Conclusion

The performance of KRPCA and other methods using a least squares data fidely term is susceptible to be impacted by heterogeneous noise or outliers that appears in the context of TWRI. Indeed, as described in [Ollila et al., 2012],

most radar clutter types can be described as heterogeneous. For example, in the context of TWRI, a drywall will not have homogeneous returns in power across measurement positions. Moreover, the wall characteristics (permittivity and conductivity) may be dependent on frequency i.e. the wall is dispersive [Amin, 2017]. This motivates us to inspect robust extensions of KRPCA.

4 - Robustifying KRPCA

Contents

In order to alleviate the potential problems in estimation caused by heterogeneous noise or outliers, we set out to include a robust distance [Maronna et al., 2019] in our problem formulation to model the data closeness. The works in this chapter were published in [Brehier et al., 2023b] and [Brehier et al., 2024b].

4.1 . HKRPCA: a robust low rank and sparse decomposition

4.1.1 . Problem statement

Recall KRPCA's problem:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{R}} \quad & \|\mathbf{L}\|_* + \lambda \|\mathbf{R}\|_{2,1} \\ \text{s.t.} \quad & \mathbf{Y} = \mathbf{L} - \boldsymbol{\Psi}\left(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R})\right) \end{aligned} \tag{4.1}$$

which may be relaxed to handle noise:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{R}} \quad & \|\mathbf{L}\|_* + \lambda \|\mathbf{R}\|_{2,1} \\ \text{s.t.} \quad & \|\mathbf{Y} - \mathbf{L} - \boldsymbol{\Psi}\left(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R})\right)\|_F^2 \leq \epsilon \end{aligned} \tag{4.2}$$

This can be tackled via the following regularized form:

$$\min_{\mathbf{L},\mathbf{R}} \quad \|\mathbf{L}\|_* + \lambda \|\mathbf{R}\|_{2,1} + \frac{\mu}{2} \|\mathbf{Y} - \mathbf{L} + \boldsymbol{\Psi}\left(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R})\right)\|_F^2 \tag{4.3}$$

which shows the balance between a regularization term and a data fitting term. In an Euclidean space, without constraints, the squared distance between two points $\mathbf{X}_1, \mathbf{X}_2$ is defined as:

$$\text{dist}^2(\mathbf{X}_1, \mathbf{X}_2) = \|\mathbf{X}_1 - \mathbf{X}_2\|_F^2 \tag{4.4}$$

We may extend this distance to many others. Of particular interest are distances that are robust to outliers in the data. An outlier is a data point that differs significantly from other observations (see Figure 4.1).

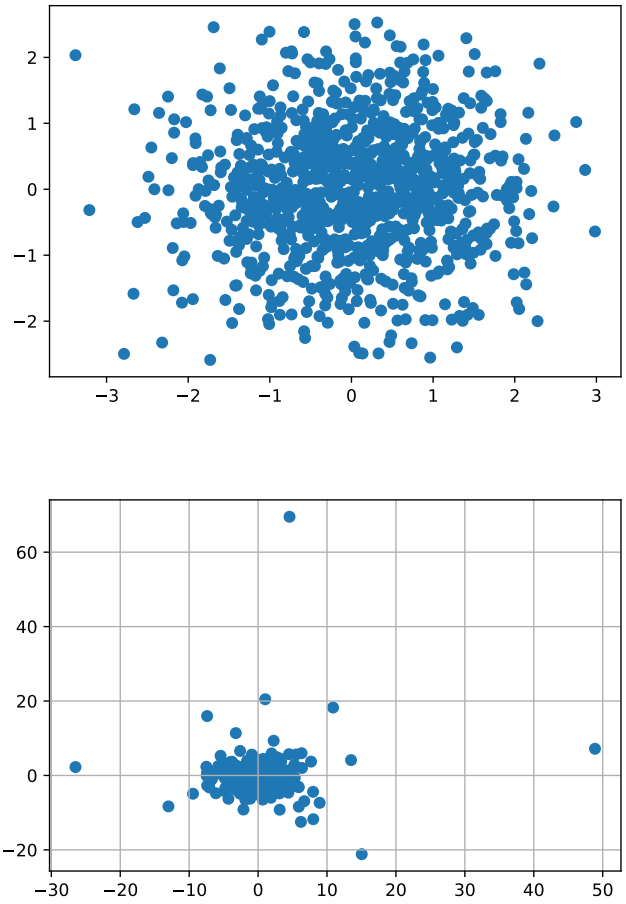


Figure 4.1: 1000 samples points of a 2D standard normal distribution (top) and a standard student-t distribution with 2 degrees of freedom (down). Outliers appear for the t-distribution

In statistical terms, the presence of outliers implies that the noise is not adequately represented by a normal distribution. Rather it may be represented by heavy-tailed ones, such as the student-t distribution, which arises

as the ratio of a normally distributed random variable and the square root of a chi-squared one divided by its degrees of freedom. For example, we may construct a robust distance via the Huber function [Huber, 1964] (with threshold $c \in \mathbb{R}^+$) denoted H_c and defined for any $x \in \mathbb{C}$ as:

$$H_c(x) = \begin{cases} \frac{1}{2}|x|^2 & \text{if } |x| \leq c \\ c(|x| - \frac{1}{2}c) & \text{if } |x| > c \end{cases} \tag{4.5}$$

The rationale behind such a function is that outliers are higher contributors to the data fitting term than other points. Having a linear term in the loss specifically for them will lower their influence while the inliers will contribute to the loss via a quadratic term, similar to classical least squares. The Huber function is shown in Figure 4.2 versus a simple quadratic (squared) function.This function H_c is convex and continuously differentiable since it has equal slopes from both ends at the two junction points where $|x| = c$. From the theory of M-estimators [Maronna et al., 2019], we know that, for the estimation of a location parameter, least squares minimization would yield the mean whereas least absolute deviations minimization the median. Using H_c yields a solution closely related to winsorizing (setting outliers points outside some central percentile range to the value at this is cutoff percentile) as explained in the seminal work [Huber, 1964]. It manages to balance robustness while not neglecting the overall behavior and has become a staple of robust estimation. It is not the only function which can accomplish this task: we can mention the $\log\cosh$ function (see Figure 4.2) among others (many aim at smoothing the absolute value function at zero). Now, recall that the norm inducing the Euclidean distance is separable across entries. For $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{C}^{M \times N}$:

$$\|\mathbf{X}_1 - \mathbf{X}_2\|_F^2 = \sum_{i=1}^{M}\sum_{j=1}^{N}[\mathbf{X}_1 - \mathbf{X}_2]_{i,j}^2 \tag{4.6}$$

Then, we may change the distance function as follows:

$$\sum_{i=1}^{M}\sum_{j=1}^{N} H_c([\mathbf{X}_1 - \mathbf{X}_2]_{i,j}^2) \tag{4.7}$$

Moreover, we are not restricted to taking the sum over entries. We may consider any block structure as long as it forms a partition of the entries. With this in mind, we define a new optimization problem, which we call HKRPCA (for Huber-type KRPCA):

$$\boxed{\min_{\mathbf{L},\mathbf{R}} \quad \|\mathbf{L}\|_* + \lambda\|\mathbf{R}\|_{2,1} + \frac{\mu}{2}\sum_{p_i \in \mathcal{P}} H_c\left(\|[\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R}))]_{p_i}\|_F\right)} \tag{4.8}$$

with $\mathcal{P}$ a **partition** of the entries of the residual matrix with i^{th} element p_i. Thus, p_i represents the support of the i^{th} block. The block-wise partition of

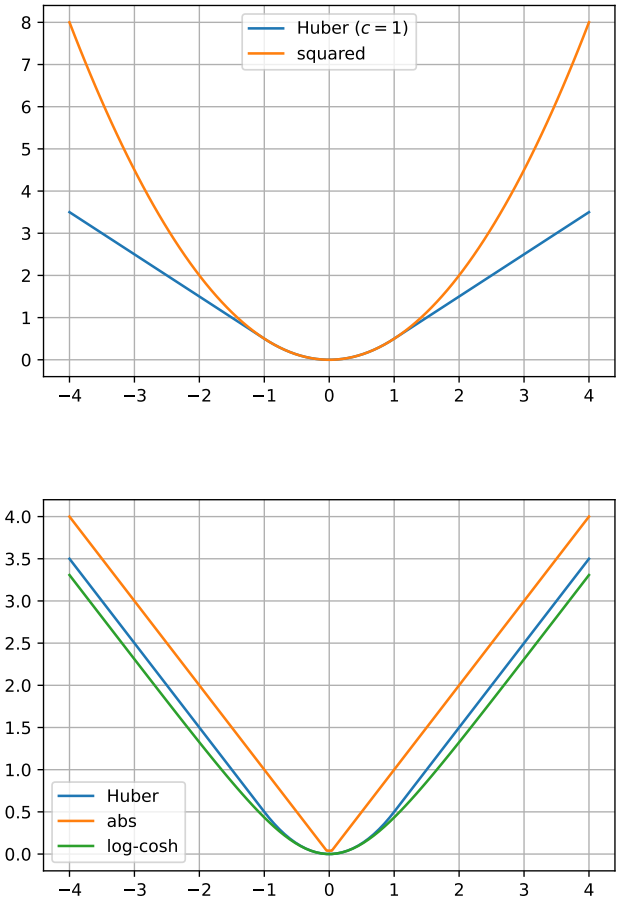


Figure 4.2: Huber function ($c = 1$) vs quadratic/squared (top) and other robust functions (down)

entries allows us to model the outliers flexibly: we may consider an entriwise partition or a columnwise partition,etc. For example, if the wall materials are structured rather than homogeneous, the noise power may be variable by radar position, which induces a column-wise heterogeneity that can be taken into account in a column-wise partition.

4.1.2 . Resolution: ADMM algorithm with a semi-split of variables

Handling the problem (4.8) directly can be achieved by proximal gradient descent alternated on the two variables. However, a strategy to obtain closed form updates is to introduce auxiliary variables to decouple the terms of the objective function. We introduce one auxiliary variable $\mathbf{M} = \mathbf{L}$ to decouple the nuclear norm from the Huber cost. We will see later that the split of $\mathbf{r}$ does

not yield a similar proximal closed form. We consider the problem:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{R},\mathbf{M}} \quad & \|\mathbf{M}\|_* + \lambda\|\mathbf{R}\|_{2,1} + \frac{\mu}{2}\sum_{p_i\in\mathcal{P}} H_c(\|[\mathbf{Y}-\mathbf{L}-\mathbf{\Psi}(\mathbf{I}_N\otimes\mathrm{vec}(\mathbf{R}))]_{p_i}\|_F) \\ \text{s.t.} \quad & \mathbf{M}=\mathbf{L} \end{aligned} \tag{4.9}$$

This semi-splitting problem (4.9) can be tackled through the ADMM framework. The Augmented Lagrangian associated with (4.9) is:

$$\begin{aligned} l(\mathbf{L},\mathbf{R},\mathbf{M},\mathbf{\Gamma}) = \|\mathbf{M}\|_* + \lambda\|\mathbf{R}\|_{2,1} + \langle\mathbf{\Gamma},\mathbf{M}-\mathbf{L}\rangle + \frac{\nu}{2}\|\mathbf{M}-\mathbf{L}\|_F^2 \\ +\frac{\mu}{2}\sum_{p_i\in\mathcal{P}} H_c(\|[\mathbf{Y}-\mathbf{L}-\mathbf{\Psi}(\mathbf{I}_N\otimes\mathrm{vec}(\mathbf{R}))]_{p_i}\|_F) \end{aligned} \tag{4.10}$$

As for KRPCA, the following subsections will detail the update of each variable for minimizing $l(\mathbf{L},\mathbf{R},\mathbf{M},\mathbf{\Gamma})$.

L-update

For this variable, the minimization consists in finding:

$$\arg\min_{\mathbf{L}} \quad \frac{\mu}{2}\sum_{p_i\in\mathcal{P}} H_c(\|[\mathbf{Y}-\mathbf{L}-\mathbf{\Psi}(\mathbf{I}_N\otimes\mathrm{vec}(\mathbf{R}))]_{p_i}\|_F) + \frac{\nu}{2}\left\|\mathbf{M}-\mathbf{L}+\frac{1}{\nu}\mathbf{\Gamma}\right\|_F^2 \tag{4.11}$$

The resulting update solving for (4.11) is given in the following proposition.

Proposition 5. *The solution is* $\forall p_i \in \mathcal{P}$*:*

$$\begin{aligned} [\mathbf{L}]_{p_i} = \mathit{prox}_{(\mu/2\nu)H_c\circ\|\cdot\|_F}\left([\mathbf{M}+\frac{1}{\nu}\mathbf{\Gamma}-\mathbf{Y}+\mathbf{\Psi}(\mathbf{I}_N\otimes\mathrm{vec}(\mathbf{R}))]_{p_i}\right) \\ +[\mathbf{Y}-\mathbf{\Psi}(\mathbf{I}_N\otimes\mathrm{vec}(\mathbf{R}))]_{p_i} \end{aligned} \tag{4.12}$$

with the proximal defined in the proof below (equations (4.15) *and* (4.16)*).*

Proof. The problem (4.11) is separable in the blocks $\{[\mathbf{L}]_{p_i}\}$:

$$\begin{aligned} \min_{\{[\mathbf{L}]_{p_i}\}} \quad & \frac{\mu}{2}\sum_{p_i\in\mathcal{P}} H_c(\|[\mathbf{Y}-\mathbf{L}-\mathbf{\Psi}(\mathbf{I}_N\otimes\mathrm{vec}(\mathbf{R}))]_{p_i}\|_F) \\ & +\frac{\nu}{2}\sum_{p_i\in\mathcal{P}}\left\|[\mathbf{M}-\mathbf{L}+\frac{1}{\nu}\mathbf{\Gamma}]_{p_i}\right\|_F^2 \end{aligned} \tag{4.13}$$

By the separability property of the proximals [Parikh and Boyd, 2014], we can consider the proximal over each block separately:

$$\min_{[\mathbf{L}]_{p_i}} \quad \frac{\mu}{2}H_c(\|[\mathbf{Y}-\mathbf{L}-\mathbf{\Psi}(\mathbf{I}_N\otimes\mathrm{vec}(\mathbf{R}))]_{p_i}\|_F) + \frac{\nu}{2}\left\|[\mathbf{M}-\mathbf{L}+\frac{1}{\nu}\mathbf{\Gamma}]_{p_i}\right\|_F^2 \tag{4.14}$$

We then compute the proximal of $f(\mathbf{X}) = H_c(\|\mathbf{X} + \mathbf{B}\|_F)$ with $\mathbf{B}$ a constant term. The proximal of the Huber function has a known form [Beck, 2017]:

$$\text{prox}_{aH_c}(x) = \left(1 - \frac{a}{\max(|\frac{x}{c}|, a+1)}\right) x \tag{4.15}$$

We can then leverage a theorem of norm composition [Beck, 2017] to get:

$$\text{prox}_{aH_c \circ \|\cdot\|_F}(\mathbf{X}) = \begin{cases} \text{prox}_{aH_c}(\|\mathbf{X}\|_F) \cdot \frac{\mathbf{X}}{\|\mathbf{X}\|_F} & \text{if } \mathbf{X} \neq \mathbf{0} \\ \mathbf{0} & \text{if } \mathbf{X} = \mathbf{0} \end{cases} \tag{4.16}$$

We finally use the translation properties of proximal operators, so that, $\forall p_i \in \mathcal{P}$, the update is:

$$\begin{aligned} [\mathbf{L}]_{p_i} = \text{prox}_{(\mu/2\nu)H_c \circ \|\cdot\|_F} &\left([\mathbf{M} + \frac{1}{\nu}\mathbf{\Gamma} - \mathbf{Y} + \mathbf{\Psi}(\mathbf{I}_N \otimes \text{vec}(\mathbf{R}))]_{p_i} \right) \\ &+ [\mathbf{Y} - \mathbf{\Psi}(\mathbf{I}_N \otimes \text{vec}(\mathbf{R}))]_{p_i} \end{aligned} \tag{4.17}$$

□

This gives a closed-form update for the $\mathbf{L}$-step. We will see later that the auxiliary variable $\mathbf{M}$ as well as the dual variable $\mathbf{\Gamma}$ also have closed-forms.

$\mathbf{R}$-update: via PGD

The minimization problem over $\mathbf{R}$ is:

$$\min_{\mathbf{R}} \quad \frac{\mu}{2} \sum_{p_i \in \mathcal{P}} H_c(\|[\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \text{vec}(\mathbf{R}))]_{p_i}\|_F) + \lambda \|\mathbf{R}\|_{2,1} \tag{4.18}$$

It is possible to use PGD for the minimization of this variable. We will consider the vectorized variable $\mathbf{r}$ to compute the gradient and unvectorize the solution to apply the proximal operator. At iteration $t+1$, with step-size s, we have:

$$\mathbf{R}_{t+1} = T_{\lambda s}\left(\text{vec}^{-1}\left(\mathbf{r}_t - s\frac{\mu}{2}\mathbf{g}_t\right)\right) \tag{4.19}$$

where $\mathbf{g}$ is the needed gradient of the sum of Huber functions.

Proposition 6. *The gradient $\mathbf{g}$ w.r.t. $\mathbf{r}$ is:*

$$\mathbf{g} = -\sum_{p_i \in \mathcal{P}} \frac{H_c'(\|[\mathbf{E}]_{p_i}\|_F)}{\|[\mathbf{E}]_{p_i}\|_F} \left(\sum_{(j,k) \in p_i} [\mathbf{E}]_{j,k} (\mathbf{\Psi}_k)_{j,:}^H \right) \tag{4.20}$$

where $\mathbf{E} = \mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathbf{r})$ and $(\mathbf{\Psi}_k)_{j,:}$ denotes the j^{th} line of $\mathbf{\Psi}_k$.

Proof. The gradient is computed according to the Wirtinger calculus, since we have an objective function of complex variables. This is detailed in the Appendix A. Gradient descent in this setting is achieved with:

$$\mathbf{g} = 2\frac{d}{d\mathbf{r}^*}\left(\sum_{p_i\in\mathcal{P}} H_c(\|[\mathbf{Y}-\mathbf{L}-\mathbf{\Psi}(\mathbf{I}_N\otimes\mathbf{r})]_{p_i}\|_F)\right) \tag{4.21}$$

Using the chain rule, we get:

$$\mathbf{g} = 2\frac{d}{d\mathbf{r}^*}\sum_{p_i\in\mathcal{P}} H_c(\|[\mathbf{E}]_{p_i}\|_F) = \sum_{p_i\in\mathcal{P}} \frac{H_c'(\|[\mathbf{E}]_{p_i}\|_F)}{\|[\mathbf{E}]_{p_i}\|_F}\cdot\frac{d}{d\mathbf{r}^*}\|[\mathbf{E}]_{p_i}\|_F^2 \tag{4.22}$$

with the derivative of H_c being:

$$H_c'(x) = \begin{cases} x & \text{if } |x|\leq c \\ c\,\mathrm{sgn}(x) & \text{if } |x|>c \end{cases} \tag{4.23}$$

where sgn denotes the sign function. Finally, we compute:

$$\begin{aligned}\frac{d}{d\mathbf{r}^*}\|[\mathbf{E}]_{p_i}\|_F^2 &= \sum_{(j,k)\in p_i}\frac{d}{d\mathbf{r}^*}\left|[\mathbf{Y}]_{j,k}-[\mathbf{L}]_{j,k}-(\mathbf{\Psi}_k)_{j,:}\mathbf{r}\right|^2 \\ &= \sum_{(j,k)\in p_i} -([\mathbf{Y}]_{j,k}-[\mathbf{L}]_{j,k}-(\mathbf{\Psi}_k)_{j,:}\mathbf{r})(\mathbf{\Psi}_k)_{j,:}^H\end{aligned} \tag{4.24}$$

where $(\mathbf{\Psi}_k)_{j,:}\mathbf{r}$ is a scalar as $(\mathbf{\Psi}_k)_{j,:}$ denotes the j^{th} line of $\mathbf{\Psi}_k$. Then:

$$\mathbf{g} = -\sum_{p_i\in\mathcal{P}}\frac{H_c'(\|[\mathbf{E}]_{p_i}\|_F)}{\|[\mathbf{E}]_{p_i}\|_F}\left(\sum_{(j,k)\in p_i}[\mathbf{E}]_{j,k}(\mathbf{\Psi}_k)_{j,:}^H\right) \tag{4.25}$$

□

The stepsize can be found by backtracking line-search (via Armijo's rule), which consists of iteratively shrinking an initially large step size until a sufficient decrease has been achieved. In practice, the stepsize does not vary over iterations so that it can be fixed to one precomputed value (linked to the Lipschitz constant of the gradient above). The gradient $\mathbf{g}$ may be compactly written for faster implementation:

$$\mathbf{g} = -\mathbf{\Psi}_g\mathrm{bdiag}(\mathbf{e}_g)[h]_g = -\mathbf{\Psi}_g(\mathbf{e}_g\odot([h]_g\otimes\mathbf{1})) \tag{4.26}$$

where $\mathbf{1}$ is a vector of ones and $\odot$ denotes the Hadamard product. The operator bdiag assigns a block diagonal matrix to a composite vector, $\mathbf{\Psi}_g$ collects the dictionary vectors in the innermost sum, $\mathbf{e}_g$ the associated residues, and $[h]_g$ the fraction of norms in the outermost sum. Note that $\mathbf{e}_g\odot([h]_g\otimes\mathbf{1})$ is faster to compute than $\mathrm{bdiag}(\mathbf{e}_g)[h]_g$ as it avoids summing over the zeros of the block-diagonal matrix.

$\mathbf{M}$-update

Thanks to the variable split, the update $\mathbf{M}$ appears as a classical proximal problem with a closed form solution. Indeed, after completing the squared norm, the problem of solving (4.10) over $\mathbf{M}$ consists in finding:

$$\underset{\mathbf{M}}{\arg\min} \quad \|\mathbf{M}\|_* + \frac{\nu}{2}\left\|\mathbf{M} - \mathbf{L} + \frac{1}{\nu}\mathbf{\Gamma}\right\|_F^2 \tag{4.27}$$

which is a proximal of the nuclear norm. Thus:

$$\mathbf{M} = D_{1/\nu}(\mathbf{L} - \frac{1}{\nu}\mathbf{\Gamma}) \tag{4.28}$$

$\mathbf{\Gamma}$-update

Finally, the $\mathbf{\Gamma}$ update is a standard step of ADMM, the dual ascent step:

$$\mathbf{\Gamma} = \mathbf{\Gamma} + \nu(\mathbf{M} - \mathbf{L}) \tag{4.29}$$

The method is summarized in Algorithm 12.

Algorithm 12 Algorithm for HKRPCA (semi variable splitting)

Have: $\{\mathbf{y}_i\}_{i=1}^N, \{\mathbf{\Psi}_i\}_{i=1}^N$
Choose: $\lambda, \mu, \nu, \eta, c, t$ and $\mathcal{P}$
$\mathbf{Y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_N]$
$\mathbf{\Psi} \triangleq [\mathbf{\Psi}_1, \mathbf{\Psi}_2, \dots \mathbf{\Psi}_N]$
$\mathbf{\Psi}_A \triangleq [\mathbf{\Psi}_1^T \mathbf{\Psi}_2^T \dots \mathbf{\Psi}_N^T]^T$
Initialize: $\mathbf{L}, \mathbf{R}, \mathbf{M}, \mathbf{\Gamma}$
repeat:
    $[\mathbf{L}]_{p_i} = \text{prox}_{(\mu/2\nu)H_c \circ \|\cdot\|_F}\left([\mathbf{M} + \frac{1}{\nu}\mathbf{\Gamma} - \mathbf{Y} + \mathbf{\Psi}(\mathbf{I}_N \otimes \text{vec}(\mathbf{R}))]_{p_i}\right)$
        $+ [\mathbf{Y} - \mathbf{\Psi}(\mathbf{I}_N \otimes \text{vec}(\mathbf{R}))]_{p_i} \quad \forall p_i \in \mathcal{P}$
    repeat:
        $\mathbf{E} = \mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \text{vec}(\mathbf{R}))$
        $\mathbf{G} = -\text{vec}^{-1}\left(\sum_{p_i \in \mathcal{P}} \frac{H_c'(\|[\mathbf{E}]_{p_i}\|_F)}{\|[\mathbf{E}]_{p_i}\|_F}\left(\sum_{(j,k)\in p_i}[\mathbf{E}]_{j,k}(\mathbf{\Psi}_k)_{j,:}^H\right)\right)$
        $\mathbf{R} = T_{\lambda s}\left(\mathbf{R} - s\frac{\mu}{2}\mathbf{G}\right)$
    until stopping criterion is met
    $\mathbf{M} = D_{1/\nu}(\mathbf{L} - \frac{1}{\nu}\mathbf{\Gamma})$
    $\mathbf{\Gamma} = \mathbf{\Gamma} + \nu(\mathbf{M} - \mathbf{L})$
until stopping criterion is met

4.1.3 . Alternative resolution: ADMM algorithm with full variable splitting

The update for $\mathbf{r}$ via PGD is not the only option, we may avoid the use of an unknown stepsize tuned via linesearch. To do so, we begin by splitting $\mathbf{r}$ similarly to $\mathbf{L}$, in order to decouple the terms it appears in. If we take this route, the formulation is:

$$\boxed{\begin{aligned} \min_{\mathbf{L},\mathbf{R},\mathbf{M},\mathbf{S}} \quad & \|\mathbf{M}\|_* + \lambda\|\mathbf{S}\|_{2,1} + \frac{\mu}{2}\sum_{p_i\in\mathcal{P}} H_c(\|[\mathbf{Y}-\mathbf{L}-\mathbf{\Psi}(\mathbf{I}_N\otimes\mathrm{vec}(\mathbf{R}))]_{p_i}\|_F) \\ \text{s.t.} \quad & \mathbf{M}=\mathbf{L}, \quad \mathbf{S}=\mathbf{R} \end{aligned}} \tag{4.30}$$

This full variable splitting problem (4.30) can be tackled through the ADMM framework. The Augmented Lagrangian associated with (4.30) is:

$$\begin{aligned} l(\mathbf{L},\mathbf{R},\mathbf{M},\mathbf{S},\mathbf{\Gamma},\tilde{\mathbf{\Gamma}}) = \|\mathbf{M}\|_* + \lambda\|\mathbf{S}\|_{2,1} + \langle\mathbf{\Gamma},\mathbf{M}-\mathbf{L}\rangle + \frac{\nu}{2}\|\mathbf{M}-\mathbf{L}\|_F^2 \\ + \left\langle\tilde{\mathbf{\Gamma}},\mathbf{S}-\mathbf{R}\right\rangle + \frac{\eta}{2}\|\mathbf{S}-\mathbf{R}\|_F^2 \\ + \frac{\mu}{2}\sum_{p_i\in\mathcal{P}} H_c(\|[\mathbf{Y}-\mathbf{L}-\mathbf{\Psi}(\mathbf{I}_N\otimes\mathrm{vec}(\mathbf{R}))]_{p_i}\|_F) \end{aligned} \tag{4.31}$$

$\mathbf{L},\mathbf{M},\mathbf{\Gamma}$-updates

The $\mathbf{L}$, $\mathbf{M}$ and $\mathbf{\Gamma}$ updates do not change from the semi-variable splitting method. Indeed, the major difference is in the $\mathbf{R}$ update.

$\mathbf{R}$-update via MM

The objective function is in this case:

$$\min_{\mathbf{R}} \quad \frac{\mu}{2}\sum_{p_i\in\mathcal{P}} H_c(\|[\mathbf{Y}-\mathbf{L}-\mathbf{\Psi}(\mathbf{I}_N\otimes\mathrm{vec}(\mathbf{R}))]_{p_i}\|_F) + \frac{\eta}{2}\left\|\mathbf{S}-\mathbf{R}+\frac{1}{\eta}\tilde{\mathbf{\Gamma}}\right\|_F^2 \tag{4.32}$$

Via decoupling, we cannot find a similar closed-form proximal evaluation for $\mathbf{r}$ as for $\mathbf{L}$ in Proposition 5. Indeed, the sum of Huber functions is not separable over $\mathbf{r}$. Instead, we will show that the Majorization-Minimization (MM) framework [Sun et al., 2016] gives us a way to solve this subproblem iteratively.

Proposition 7. *Let $e_i(\mathbf{r}_t) = \|[\mathbf{Y}-\mathbf{L}-\mathbf{\Psi}(\mathbf{I}_N\otimes\mathbf{r}_t)]_{p_i}\|_F$. Then, $\mathbf{W}$ is defined as $[\mathbf{W}]_{j,k} = w_i(\mathbf{r}_t)$ where the $(j,k)^{th}$ entry is in the i^{th} patch, with:*

$$w_i^2(\mathbf{r}_t) = \begin{cases} 1 \text{ if } e_i(\mathbf{r}_t) \le c \\ \frac{c}{e_i(\mathbf{r}_t)} \text{ else} \end{cases} \tag{4.33}$$

Moreover, we have $\mathbf{L}_W = \mathbf{W} \odot \mathbf{L}$, $\mathbf{Y}_W = \mathbf{W} \odot \mathbf{Y}$, $\mathbf{\Psi}_{AW} = \text{vec}(\mathbf{W})\mathbf{1}^T \odot \mathbf{\Psi}_A$. *Then, a MM scheme can be tailored which converges to a critical point of* (4.32)*, with iteration* $t+1$*:*

$$\begin{aligned}\mathbf{r}_{t+1} = &\left(\frac{\mu}{2}\mathbf{\Psi}^H_{AW(\mathbf{r}_t)}\mathbf{\Psi}_{AW(\mathbf{r}_t)} + \eta\mathbf{I}\right)^{-1} \times \\ &\left(\frac{\mu}{2}\mathbf{\Psi}^H_{AW(\mathbf{r}_t)}(\text{vec}\,\mathbf{Y}_{W(\mathbf{r}_t)} - \text{vec}\,\mathbf{L}_{W(\mathbf{r}_t)}) + (\eta\,\text{vec}\,\mathbf{S} + \text{vec}\,\tilde{\mathbf{\Gamma}})\right)\end{aligned} \tag{4.34}$$

Proof. Consider the vectorized variable $\mathbf{r}$ whose update we can unvectorize for $\mathbf{R}$. The first step is to find a majorizing function of $H_c(x)$ at some point x_t that we will denote $G_c(x|x_t)$. It must be equal to H_c at the point x_t and greater at all other points. We can use the result from [de Leeuw and Lange, 2009, Theorem 4.5]:

$$G_c(x|x_t) = \frac{H_c'(x_t)}{2x_t}(x^2 - x_t^2) + H_c(x_t) \tag{4.35}$$

This is the sharpest quadratic majorizer. We can obtain:

$$G_c(x|x_t) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x_t| \leq c \\ \frac{1}{2}\frac{c}{|x_t|}x^2 + \frac{1}{2}c(|x_t| - c) & \text{if } |x_t| > c \end{cases} \tag{4.36}$$

This majorizer is plotted in Figure 4.3.

Note $\forall p_i \in \mathcal{P}$ that $e_i(\mathbf{r}) = \|[\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathbf{r})]_{p_i}\|_F$ and $e_i(\mathbf{r}_t)$ is the same quantity but with $\mathbf{r}_t$, the variable at the previous MM iteration. By the definition of G just above, we can write:

$$\arg\min_{\mathbf{r}} G_c(e_i(\mathbf{r})|e_i(\mathbf{r}_t)) = \arg\min_{\mathbf{r}} \frac{1}{2}w_i^2(\mathbf{r}_t)e_i^2(\mathbf{r}) \tag{4.37}$$

where $w_i^2(\mathbf{r}_t) = 1$ if $e_i(\mathbf{r}_t) \leq c$ or else $w_i^2(\mathbf{r}_t) = \frac{c}{e_i(\mathbf{r}_t)}$. Also note that we can sum the majorizers over all blocks to get a global one. Then, it follows that by adding the remaining quadratic term of the objective function, we get the following majorizer at point $\mathbf{r}_t$ to the objective function (4.32):

$$\mathcal{G}_c(\mathbf{r}|\mathbf{r}_t) = \frac{\mu}{2}\sum_{p_i \in \mathcal{P}} G_c(e_i(\mathbf{r})|e_i(\mathbf{r}_t)) + \frac{\eta}{2}\left\|\mathbf{r} - (\text{vec}\,\mathbf{S} + \frac{1}{\eta}\,\text{vec}\,\tilde{\mathbf{\Gamma}})\right\|_F^2 \tag{4.38}$$

So that, via the MM framework, we are left with finding:

$$\begin{aligned}\mathbf{r}_{t+1} &= \arg\min_{\mathbf{r}} \mathcal{G}_c(\mathbf{r}|\mathbf{r}_t) \\ &= \arg\min_{\mathbf{r}} \quad \frac{\mu}{4}\sum_{p_i \in \mathcal{P}} w_i^2(\mathbf{r}_t)e_i^2(\mathbf{r}) + \frac{\eta}{2}\left\|\mathbf{r} - (\text{vec}\,\mathbf{S} + \frac{1}{\eta}\,\text{vec}\,\tilde{\mathbf{\Gamma}})\right\|_F^2 \\ &= \arg\min_{\mathbf{r}} \quad \frac{\mu}{4}\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathbf{r}))\|_F^2 \\ &\qquad + \frac{\eta}{2}\left\|\mathbf{r} - (\text{vec}\,\mathbf{S} + \frac{1}{\eta}\,\text{vec}\,\tilde{\mathbf{\Gamma}})\right\|_F^2\end{aligned} \tag{4.39}$$

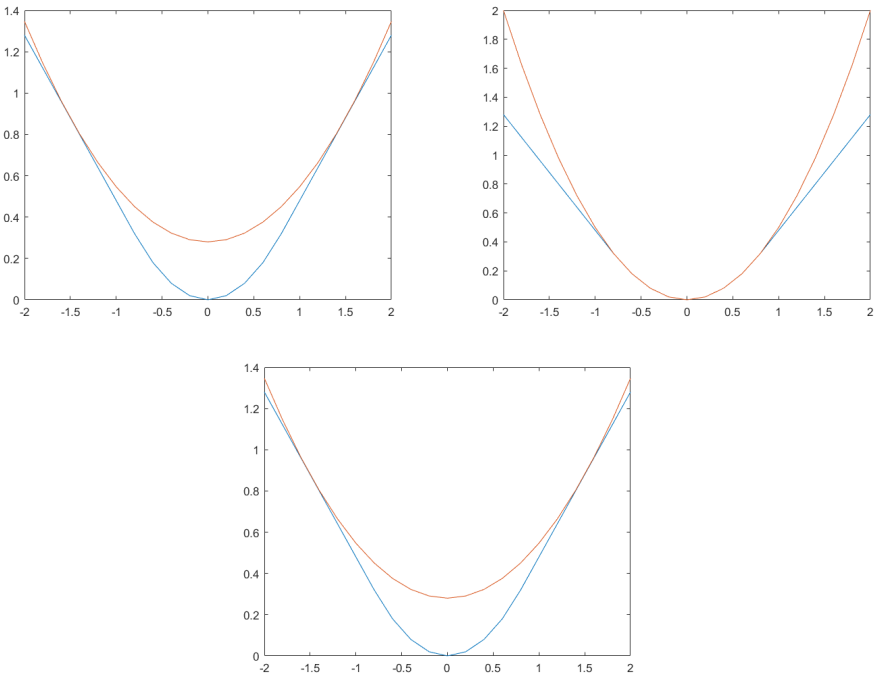


Figure 4.3: majorizing function (in orange) at $x_t = -1.5, x_t = 0.5, x_t = 1.5$ with $c = 0.8$

where $\mathbf{W}$ is such that $[\mathbf{W}]_{j,k} = w_i(\mathbf{r}_t)$ where the $(j,k)^{th}$ entry is in the i^{th} patch. To find the minimizer in (4.39), we vectorize the first term since the Frobenius norm acts component-wise. Then:

$$\begin{aligned}\mathbf{r}_{t+1} = \arg\min_{\mathbf{r}} \quad & \frac{\mu}{4}\|\mathbf{\Psi}_{AW}\mathbf{r} - (\operatorname{vec}\mathbf{Y}_W - \operatorname{vec}\mathbf{L}_W)\|_F^2 \\ & + \frac{\eta}{2}\left\|\mathbf{r} - (\operatorname{vec}\mathbf{S} + \frac{1}{\eta}\operatorname{vec}\tilde{\mathbf{\Gamma}})\right\|_F^2\end{aligned} \tag{4.40}$$

where $\mathbf{L}_W = \mathbf{W} \odot \mathbf{L}$, $\mathbf{Y}_W = \mathbf{W} \odot \mathbf{Y}$ and $\mathbf{\Psi}_{AW} = \operatorname{vec}(\mathbf{W})\mathbf{1}^T \odot \mathbf{\Psi}_A$. Via the first-order optimality conditions, we get:

$$\begin{aligned}\mathbf{r}_{t+1} = & \left(\frac{\mu}{2}\mathbf{\Psi}_{AW(\mathbf{r}_t)}^H\mathbf{\Psi}_{AW(\mathbf{r}_t)} + \eta\mathbf{I}\right)^{-1} \times \\ & \left(\frac{\mu}{2}\mathbf{\Psi}_{AW(\mathbf{r}_t)}^H(\operatorname{vec}\mathbf{Y}_{W(\mathbf{r}_t)} - \operatorname{vec}\mathbf{L}_{W(\mathbf{r}_t)}) + (\eta\operatorname{vec}\mathbf{S} + \operatorname{vec}\tilde{\mathbf{\Gamma}})\right)\end{aligned} \tag{4.41}$$

□

Finally, the $\mathbf{S}$,$\tilde{\mathbf{\Gamma}}$ updates are found in closed form.

$\mathbf{S}$-update

The update for $\mathbf{S}$ can be expressed as:

$$\min_{\mathbf{S}} \quad \lambda\|\mathbf{M}\|_{2,1} + \frac{\eta}{2}\left\|\mathbf{S} - \mathbf{R} + \frac{1}{\eta}\tilde{\mathbf{\Gamma}}\right\|_F^2 \tag{4.42}$$

whose solution is a proximal of the $\ell_{2,1}$-norm:

$$\mathbf{S} = T_{\lambda/\eta}(\mathbf{R} - \frac{1}{\eta}\tilde{\mathbf{\Gamma}}) \tag{4.43}$$

where T is the row thresholding operator.

$\tilde{\mathbf{\Gamma}}$-update

The $\tilde{\mathbf{\Gamma}}$-update is a generic ADMM step of dual ascent:

$$\tilde{\mathbf{\Gamma}} = \tilde{\mathbf{\Gamma}} + \eta(\mathbf{S} - \mathbf{R}). \tag{4.44}$$

Moreover, the dual balancing scheme [Boyd et al., 2011] to adapt the dual hyper-parameters proved useful in practice. The method is summarized in Algorithm 13.

4.1.4 . Convergence analysis

Semi-splitting algorithm

We consider the semi-splitting algorithm for HKRPCA, which we can write in the following equivalent formulation to (4.9):

$$\begin{aligned} \min_{\mathbf{L},\mathbf{R},\mathbf{M}} \quad & \|\mathbf{M}\|_* + \lambda\|\mathbf{R}\|_{2,1} + \frac{\mu}{2}\sum_{p_i\in\mathcal{P}} H_c(\|\mathbf{S}_{p_i}(-\operatorname{vec}\mathbf{Y} + \operatorname{vec}\mathbf{L} + \mathbf{\Psi}_A \operatorname{vec}\mathbf{R})\|_F) \\ \text{s.t.} \quad & \operatorname{vec}\mathbf{M} - [\mathbf{I}_{MN}, \mathbf{0}_{MN\times N_x N_z R}] \begin{bmatrix} \operatorname{vec}\mathbf{L} \\ \operatorname{vec}\mathbf{R} \end{bmatrix} = \mathbf{0}_{MN} \end{aligned} \tag{4.45}$$

where $\mathbf{S}_{p_i}$ denotes the selection matrix associated to the i^{th} block, which has a unique or no unit entry in each column/row and zeros elsewhere. $\mathbf{0}_{M\times N}$ denotes a matrix of zeros of M rows by N columns. The above problem may be cast in a 2-block ADMM with one composite variable $[\operatorname{vec}(\mathbf{L})^T, \operatorname{vec}(\mathbf{R})^T]^T$ with coefficient matrix $[\mathbf{I}_{MN}, \mathbf{0}_{MN}\mathbf{\Psi}_A] = [\mathbf{I}_{MN}, \mathbf{0}_{MN\times N_x N_z R}]$. In practice, solving directly over the composite variable is difficult so we solve for its sub-variables separately in a pass of Block Coordinate Descent (BCD), which is inexact and not part of the standard ADMM framework. Some works denoted Generalized ADMM (GADMM) [Fang et al., 2015] have been developed for approximate minimization but involve the introduction of a relaxation factor that changes the problem to solve.

Algorithm 13 Algorithm for HKRPCA (full variable splitting)

Have: $\{\mathbf{y}_i\}_{i=1}^N, \{\mathbf{\Psi}_i\}_{i=1}^N$
Choose: $\lambda, \mu, \nu, \eta, c$ and $\mathcal{P}$
$\mathbf{Y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_N]$
$\mathbf{\Psi} \triangleq [\mathbf{\Psi}_1, \mathbf{\Psi}_2, \dots \mathbf{\Psi}_N]$
$\mathbf{\Psi}_A \triangleq [\mathbf{\Psi}_1^T \mathbf{\Psi}_2^T \dots \mathbf{\Psi}_N^T]^T$
Initialize: $\mathbf{L}, \mathbf{R}, \mathbf{M}, \mathbf{S}, \mathbf{\Gamma}, \tilde{\mathbf{\Gamma}}, \mathbf{W}$
repeat:
    $[\mathbf{L}]_{p_i} = \text{prox}_{(\mu/2\nu)H_c \circ \|\cdot\|_F} \left([\mathbf{M} + \frac{1}{\nu}\mathbf{\Gamma} - \mathbf{Y} + \mathbf{\Psi}(\mathbf{I}_N \otimes \text{vec}(\mathbf{R}))]_{p_i} \right)$
        $+ [\mathbf{Y} - \mathbf{\Psi}(\mathbf{I}_N \otimes \text{vec}(\mathbf{R}))]_{p_i} \quad \forall p_i \in \mathcal{P}$
    repeat:
        $\mathbf{E} = \mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \text{vec}(\mathbf{R}))$
        $[\mathbf{W}]_{j,k} = 1$ if $\|[\mathbf{E}]_{p_i}\|_F \leq c$ else $\sqrt{c / \|[\mathbf{E}]_{p_i}\|_F} \quad \forall (j,k) \in p_i$
        $\mathbf{\Psi}_{AW} = \text{vec}(\mathbf{W})\mathbf{1}^T \odot \mathbf{\Psi}_A$
        $\mathbf{\Psi}_{AWI} = \left(\frac{\mu}{2}\mathbf{\Psi}_{AW}^H \mathbf{\Psi}_{AW} + \eta \mathbf{I}\right)^{-1}$
        $\mathbf{r} = \mathbf{\Psi}_{AWI} \left(\frac{\mu}{2}\mathbf{\Psi}_{AW}^H (\text{vec}\, \mathbf{Y}_W - \text{vec}\, \mathbf{L}_W) + \frac{1}{\eta} \text{vec}\, \mathbf{S} + \text{vec}\, \tilde{\mathbf{\Gamma}} \right)$
    until stopping criterion is met
    $\mathbf{M} = D_{1/\nu}(\mathbf{L} - \frac{1}{\nu}\mathbf{\Gamma})$
    $\mathbf{S} = T_{\lambda/\eta}(\mathbf{R} - \frac{1}{\eta}\tilde{\mathbf{\Gamma}})$
    $\mathbf{\Gamma} = \mathbf{\Gamma} + \nu(\mathbf{M} - \mathbf{L})$
    $\tilde{\mathbf{\Gamma}} = \tilde{\mathbf{\Gamma}} + \eta(\mathbf{S} - \mathbf{R})$
until stopping criterion is met

We might think to cast the problem in a 3-block ADMM, which has been a topic of research the past few years [Han, 2022, Chen et al., 2016]: not necessarily convergent, a simple sufficient condition for its convergence is that any two coefficient matrices in the constraints must be orthogonal to each other. But, in our case, the objective function is not separable in the different components of the composite variable, so that we cannot apply the 3-block ADMM. Thus, to the best of our knowledge, the analysis of the convergence of such a BCD split in a 2-block ADMM remains an open question while our experiments in the following Section 4.2 show its good practical recovery of the seeked result. The alternative use of GADMM may be investigated but will necessitate to solve new subproblems and to verify some additional suboptimality conditions.

Full-splitting algorithm

In the case of a full split of variables i.e. splitting both $\mathbf{L}$ and $\mathbf{r}$, we can rewrite the problem in the equivalent formulation:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{R},\mathbf{M},\mathbf{S}} \quad & \|\mathbf{M}\|_* + \lambda\|\mathbf{S}\|_{2,1} + \frac{\mu}{2}\sum_{p_i\in\mathcal{P}} H_c(\|\mathbf{S}_{p_i}(-\operatorname{vec}\mathbf{Y} + \operatorname{vec}\mathbf{L} + \mathbf{\Psi}_A \operatorname{vec}\mathbf{R})\|_F) \\ \text{s.t.} \quad & \begin{bmatrix}\operatorname{vec}\mathbf{M}\\ \operatorname{vec}\mathbf{S}\end{bmatrix} - \begin{bmatrix}\operatorname{vec}\mathbf{L}\\ \operatorname{vec}\mathbf{R}\end{bmatrix} = \begin{bmatrix}\mathbf{0}_{MN}\\ \mathbf{0}_{N_xN_zR}\end{bmatrix} \end{aligned} \tag{4.46}$$

we see that it lies within the 2-block ADMM with two composite variables $[\operatorname{vec}(\mathbf{L})^T, \operatorname{vec}(\mathbf{R})^T]^T$ and $[\operatorname{vec}(\mathbf{M})^T, \operatorname{vec}(\mathbf{S})^T]^T$. Again, we only do inexact minimization over $[\operatorname{vec}(\mathbf{L})^T, \operatorname{vec}(\mathbf{R})^T]^T$ as well as for $[\operatorname{vec}(\mathbf{M})^T, \operatorname{vec}(\mathbf{S})^T]^T$ via Block Coordinate Descent (BCD). The question of its convergence is thus also open while experiments show good results.

4.1.5 . Computational complexity

Semi-splitting algorithm

Assuming the same ordering of dimensions as for KRPCA, i.e. that $D > M > N$ where $D = N_xN_zR$ and that $MN > D$. The proximal operator of the Huber function composed with the Frobenius norm (plus a translation) is not the most costly operation as it scales linearly with the input matrix dimensions (so it is $\mathcal{O}(MN)$). The evaluation of $\mathbf{\Psi}(\mathbf{I}_N \otimes \mathbf{r})$ is $\mathcal{O}(MND)$ as well as for the gradient evaluation in the PGD. Setting the number of PGD iterations to K, we have a computational complexity of $\mathcal{O}(KMND)$ for the algorithm.

Full-splitting algorithm

Via full splitting, thus via a MM step for $\mathbf{r}$, we have the task of inverting a matrix at each MM iteration (or solving the associated linear system of equations) of size D which will be $\mathcal{O}(D^3)$ via Gaussian elimination. However, the major cost is the computation of the matrix product $\mathbf{\Psi}_{AW}^H\mathbf{\Psi}_{AW}$ inside the inverse, which will be $\mathcal{O}(NMD^2)$ and cannot be cached. This time again, consider K iterations of MM. Then, the cost of the $\mathbf{r}$-update via MM is $\mathcal{O}(KMND^2)$, which will be the overall computational complexity of the full splitting algorithm. Table 4.1 recapitulates the complexities of all algorithms proposed in this chapter. We see the higher iteration cost of the full decoupling method compared to the semi-decoupling one.

Figure 4.4 presents a study of the convergence speed of the different methods. In the point-block method, $\forall p_i \in \mathcal{P}$, p_i is the i^{th} entry of $\operatorname{vec}(\mathbf{Y})$. We denote this setup for the semi-decoupling algorithm as HKRPCA SD-pt and HKRPCA FD-pt for the full-decoupling algorithm. In the column-block method, $\forall p_i \in \mathcal{P}$, p_i is the support of the i^{th} column $\mathbf{y}_i$. We denote this setup for the

Method	KRPCA	HKRPCA SD	HKRPCA FD
Complexity	$\mathcal{O}(KMND)$	$\mathcal{O}(KMND)$	$\mathcal{O}(KMND^2)$

Table 4.1: Computational complexity of the introduced methods

semi-decoupling algorithm as HKRPCA SD-col and HKRPCA FD-col for the full-decoupling algorithm. It should be kept in mind that the different methods have different objective functions. Nevertheless, we see that their convergence in terms of iterations, except SRCS, behaves similarly. Over time, we see that the point-wise HKRPCA methods (HKRPCA SD-pt and HKRPCA FD-pt) perform similarly albeit a bit slower than KRPCA, whereas their column-wise counterparts are noticeably slower (HKRPCA SD-col and HKRPCA FD-col). This is explained by the implementation: the point-wise application of the Huber function can be vectorized over the matrix, whereas the column-wise case necessitates the slicing of the matrix along the columns before applying the Huber function, which is computationally more demanding.

4.2 . Experiments

4.2.1 . Simulation setup

FDTD data

We test our methods on electromagnetic simulations via Finite-Difference Time-Domain (FDTD) with GprMax [Warren et al., 2016]. We detail this method in the Appendix B. The scene, as described in Figure 1.8, is 4.9×5.4 m in crossrange (x-axis) vs downrange (z-axis) with a discretization step of 3mm. The front wall (parallel to the SAR movement) is at a standoff distance to the radar of 1.2 m. It is homogeneous and non-conductive, of thickness 20cm and relative permittivity $\epsilon = 4.5$. One target is behind the wall, a perfect electric conductor (PEC) cylinder of radius 3mm situated at coordinates $(2.6, 4)$. The radar moves 2cm along the x-axis between each acquisition, starting from $x = 1.824$m, with 67 different positions overall. As GrpMax works by sending pulses, we use a ricker wavelet centered at 2 GHz on which we apply a FFT to extract the frequency spectrum within the range of the bandwidth ($1 - 3$ GHz).

Noise generation

To simulate different data acquisitions, we add random heterogeneous noise drawn from student-t noise. We will consider both pointwise and columnwise noise. The column-wise noise heterogeneity may arise as a result of the wall structure, e.g., drywall. The pointwise case may arise by adding the possibility of a frequency-dependant relative permittivity of the wall. Additionally, we

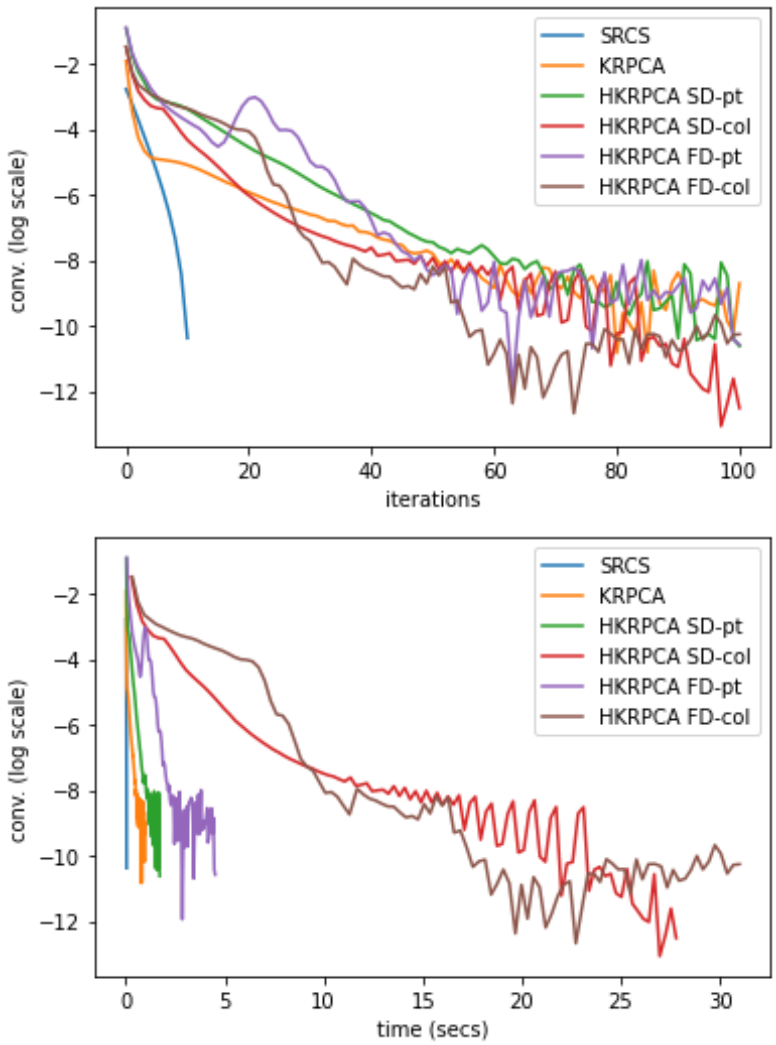


Figure 4.4: Convergence (log scale) vs iterations (top) and time (bottom)

consider the possibility of outliers coming from a different random process, which can be interpreted as mishandling in the acquisition process, etc. We first consider two pointwise cases.

- pointwise noise only: $[\mathbf{Y}]_{i,j} = [\mathbf{L} + \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R})]_{i,j} + [\mathbf{T}]_{i,j}$
- pointwise noise + outliers: $[\mathbf{Y}]_{i,j} = [\mathbf{L} + \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R})]_{i,j} + [\mathbf{T}]_{i,j} + [\mathbf{O}]_{i,j}$

with $\mathbf{T}_{i,j}$ being i.i.d. centered univariate complex t-random variables with $f > 2$ degrees of freedom (d.f.). [1] $\mathbf{O}$ is a matrix of outliers, whose number is set

[1] i.e. $\mathbf{T}_{i,j} \sim \mathcal{C}t_\nu(0, \sigma)$ where the standard deviation σ is ajusted to get the desired SNR level. This can be written in the stochastic compound Gaussian form as $\mathbf{T}_{i,j} =_d \sqrt{\tau}\mathbf{N}_{i,j}$ where $\tau =_d \frac{\nu}{x}$ is a positive and real valued random variable with $x \sim \chi^2(\nu)$ multiplying the normally distributed $\mathbf{N}_{i,j} \sim \mathcal{CN}(\mathbf{0}, \sigma\mathbf{I})$.

by the user and whose support Ω is randomly selected at uniform among all entries. The outliers are then drawn from a standard gaussian i.e. $\mathbf{O}_\Omega \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$. Entries of $\mathbf{O}$ not in Ω are then set to zero. Second, we consider two column-wise cases.

- column-wise noise only: $\mathbf{y}_i = [\mathbf{L} + \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R})]_{:,i} + [\mathbf{T}]_{:,i}$
- column wise noise + outliers: $\mathbf{y}_i = [\mathbf{L} + \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R})]_{:,i} + [\mathbf{T}]_{:,i} + [\mathbf{O}]_{:,i}$

where columns of $\mathbf{T}$ are i.i.d. random variables drawn from a m-variate t-distribution. [2] The outlying columns are selected uniformly at random among all columns, with their support denoted Ω. The entries of $\mathbf{O}$ on those columns then follow a standard gaussian distribution i.e. $\mathbf{O}_\Omega \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$ while entries not supported on Ω are set to zero.

Hyperparameter tuning

The hyperparameters have been tuned by hand in the following study. For fair comparisons, all algorithms are used with hyperparameters (when applicable): $\lambda = 1, \mu = 10, \nu = 1, c = 0.1, \eta = 1\mathrm{e}10$, which have given good results for all methods. The tuning of λ, μ follow the same reasoning as for KRPCA. Then ν, η are set in order for the auxiliary matrices to converge to the primary ones i.e. so as to enforce the equality constraints. Finally, the Huber threshold c is typically set to the median of the data matrix entries value.

All methods are run the same number of iterations, as all algorithms iterations cycle through every variable, and a comparison in terms of convergence is not possible, the methods converging based on different functionals. In order to avoid this tedious process, one may alternatively tune the hyperparameters using Bayesian optimization (see, e.g., [Snoek et al., 2012] and references therein). It uses a Gaussian Process (GP) prior over the f1-score of the detection map of the algorithm to tune. It is then possible to get an analytical formula for the posterior GP and to find sample hyperparameters to evaluate next based on some metric such as Expected Improvement. This can be readily implemented with the package BayesianOptimization [Nogueira, 2014]. This may make less calls to the algorithm, which is the proeminent cost of finding good hyperparameters, compared with a brute force grid-search where past iterations provide no information for the next chosen hyperpameters to probe.

[2] i.e. the i^{th} column $\mathbf{T}_{:,i} \sim \mathcal{C}t_{m,\nu}(\mathbf{0}, \sigma\mathbf{I})$ with $\nu > 2$. Denote the complex m-dimensional t-distribution with ν degrees of freedom (d.f.) parametrized by $(\boldsymbol{\mu}, \mathbf{\Sigma})$ as $\mathcal{C}t_{m,\nu}(\boldsymbol{\mu}, \mathbf{\Sigma})$. We then set $\boldsymbol{\mu} = \mathbf{0}$,$\mathbf{\Sigma} = \mathbf{I}$. If $\mathbf{z} \sim \mathcal{C}t_{m,\nu}(\mathbf{0}, \mathbf{I})$ then it admits the Coumpound Gaussian stochastic representation: $\mathbf{z} \overset{d}{=} \tau\mathbf{n}$ where $\tau \overset{d}{=} \frac{\nu}{x}$ is a positive and real valued random variable with $x \sim \chi^2(\nu)$ and $\mathbf{n} \sim \mathcal{CN}_m(\mathbf{0}, \mathbf{I})$.

The influence of the hyperparameters (λ, μ) on the performance of HKRPCA has been studied in Figure 4.5. There, each point's Area Under the Curve (AUC) is averaged over 30 draws. We see that there is a fairly large range of values $\lambda \in [0, 20], \mu \in [1, 100]$ where the AUC is high. Additionally, we observed empirically that the Bayesian hyperparameter tuning method does propose values in this area (e.g. $\lambda = 14, \mu = 99$ here).

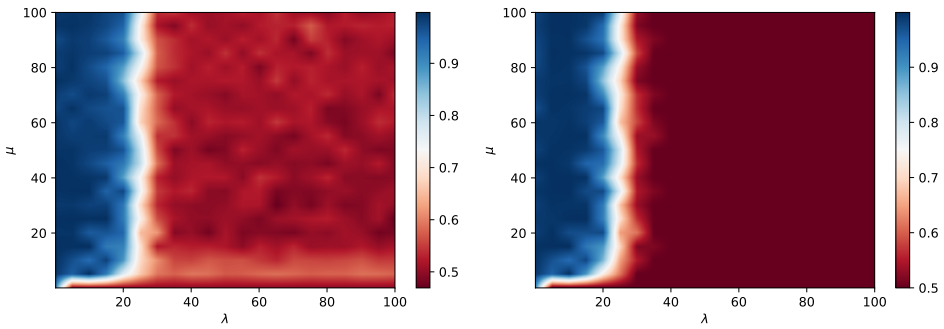


Figure 4.5: AUC over a grid of hyperparameters for HKRPCA FD-pt (left) and HKRPCA SD-pt (right) with pointwise noise

4.2.2 . Performance evaluation

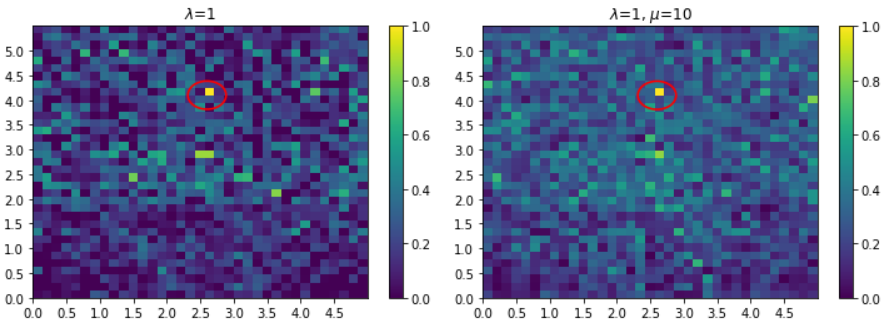
Sample results are shown for the different methods in Figure 4.6 with pointwise noise only. The target location is indicated with a red circle. We evaluate quantitatively the performance of the methods based on their Receiver Operator Characteristic (ROC) averaged over 100 draws at each point of the curve.

Pointwise noise only

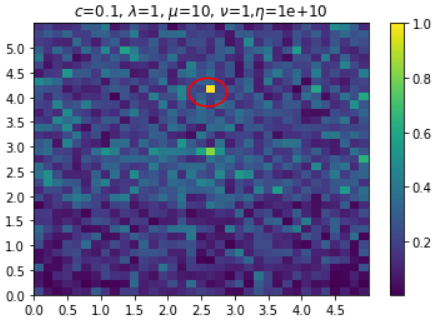
We begin with a setup consisting of only pointwise heterogeneous noise, that follow a centered multivariate student-t distribution. We chose the setup of degrees of freedom: d.f. $= 2.01$ and Signal to Noise Ratio: SNR $= 10$dB to visualize at best the difference in performance of the different methods. In Figure 4.7a, we plotted the resulting ROC. We observe that all methods with the Huber cost perform in a similar fashion. KRPCA performs worse and finally SRCS is the worst performing method.

Pointwise noise and point outliers

Next, we are interested in a setup with pointwise heterogeneous noise plus 100 point outliers, i.e. with perturbations coming from a different random process. Here the outlying entries have pointwise noise generated from a univariate standard Gaussian distribution. On Figure 4.7b, we have the resulting



(a) SRCS (b) KRPCA



(c) HKRPCA FD-pt

Figure 4.6: Sample detection maps (one target with location circled in red)

ROC. We see that both HKRPCA SD-pt and HKRPCA FD-pt perform similarly and better than HKRPCA SD-col and HKRPCA FD-col. KRPCA and SRCS are the least well performing again.

Column wise noise only

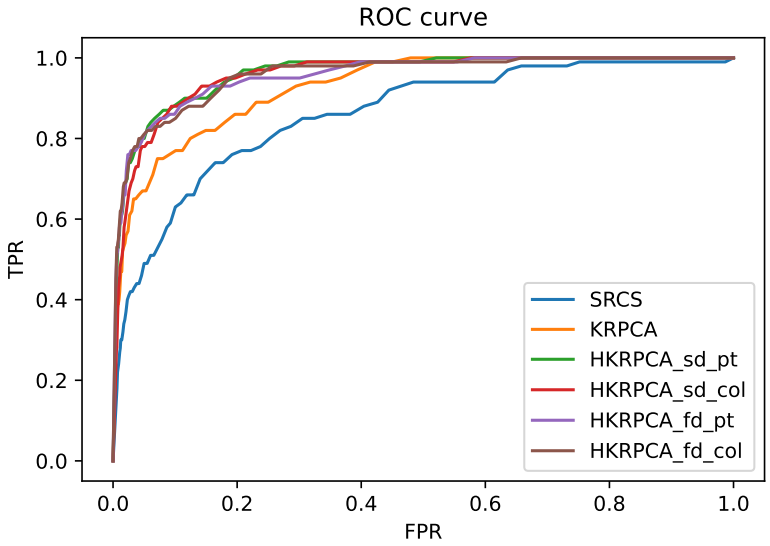
To evaluate the effect of the block-wise methods, we thus generate block-wise noise to see its effects and the resulting discrepancy in the performance of the different methods. In Figure 4.8a we have the resulting ROC with column-wise heterogeneous noise. In our setup, this means that the noise is considered radar position per radar position, and may change in power over radar acquisitions. We see here, with a bit more degraded setup than previous ones, that HKRPCA FD-col performs the best. Other methods except SRCS are a bit below the graph, and SRCS is last.

Column wise noise and column outliers

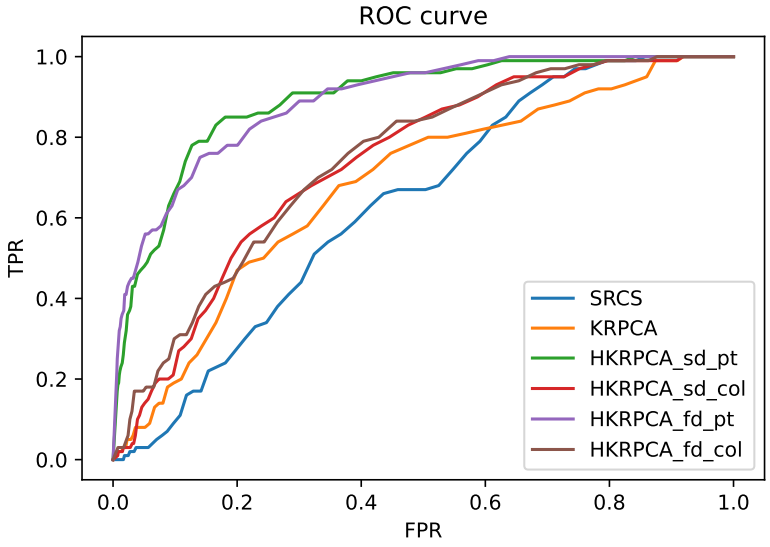
For one last setup, we add outliers to the column-wise setup. To the column-wise heterogeneous noise, we add 25 column outliers i.e. with column-wise noise generated from a standard multivariate Gaussian. In Figure 4.8b we

have the corresponding ROC. We see a clearer separation of performance between all methods. HRKPCA FD-col performs better than HKRPCA SD-col which in turn performs better than HRKPCA FD-pt. The method HRKPCA SD-pt comes after and KRPCA and SRCS are last.

On the whole, we have seen that the robust cost methods do perform better in heterogeneous noise scenarios, and that the correct block structure does impact the performance of those robust methods, especially and more clearly with outliers. Finally, the full decoupling method with an MM step performs better than the semi-decoupling method for blockwise setups.

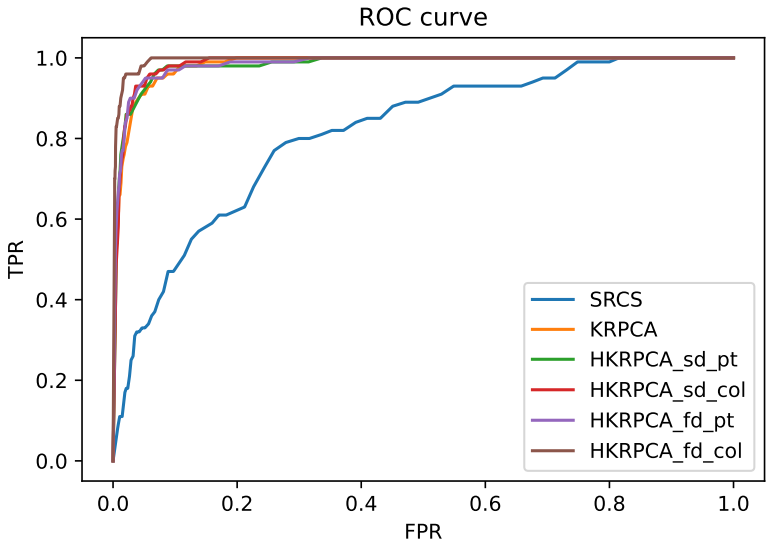


(a) ROC with only pointwise heterogeneous noise (student pointwise noise with d.f. = 2.01 and SNR= 10 dB)

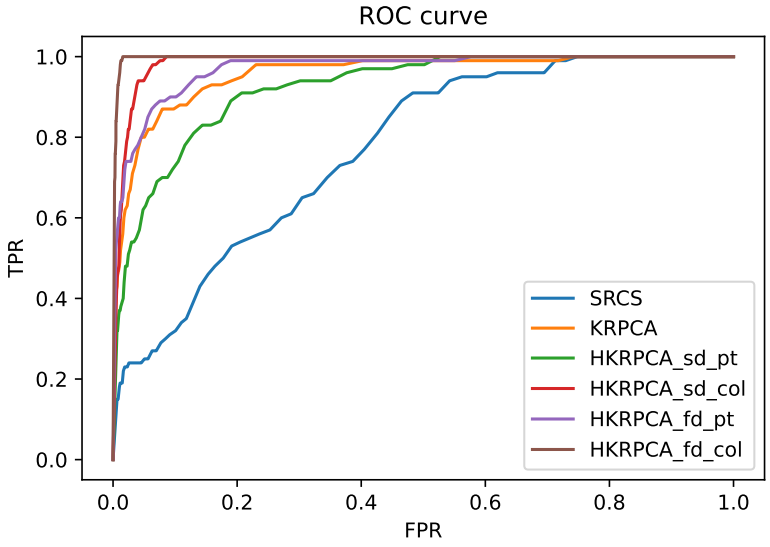


(b) ROC with pointwise noise and point outliers (student noise with d.f. = 2.1, SNR= 12 dB and 100 outliers)

Figure 4.7: ROC with pointwise corruptions



(a) ROC with only column-wise heterogeneous noise (student column-wise noise with d.f. = 2.01 and SNR= 6 dB)



(b) ROC with column noise and column outliers (student noise with d.f. = 2.1, SNR= 12 dB and 25 outliers)

Figure 4.8: ROC with column-wise corruptions

4.3 . HBCD: via Riemannian optimization

The previous section displayed a method for robust and one-step wall mitigation and target detection for TWRI through a robust data fitting in a decoupled convex relaxation. This has the advantage of having closed-form updates for $\mathbf{L}, \mathbf{M}, \mathbf{S}$ via proximal operators [Parikh and Boyd, 2014]. For $\mathbf{R}$, we can tailor a Majorization-Minimization [Sun et al., 2016] scheme which removes the need for a gradient descent and a step-size to tune.

However, this may slightly degrade performance due to the convex relaxation and the decoupling of variables. We can bypass the need for those elements by considering directly the non-convex optimization of the rank constraint. We have a good a priori of the true rank of the low-rank matrix $\mathbf{L}$ from the knowledge of the physical setup. We may then fix the rank constraint to a certain value and use Riemannian optimization. This does not require a convex relaxation of the rank nor a decoupling variable. Moreover, using a factorized representation of the low-rank component, the optimization may be done without computing a Singular Value Decomposition (SVD) at each iteration. The following work was published in [Brehier et al., 2024b].

We thus consider the following optimization program where the rank and cardinality constraint have not been convexely relaxed:

$$\boxed{\begin{aligned} &\min_{\mathbf{L},\mathbf{R}} \sum_{ij} H_c([\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I} \otimes \mathrm{vec}(\mathbf{R}))]_{ij}) \\ &\text{s.t. } \mathrm{rk}(\mathbf{L}) = k \ , \ \|\mathbf{R}\|_{2,0} \leq l \end{aligned}} \tag{4.47}$$

which can be tackled via a Block Coordinate Descent (BCD) over $\mathbf{L}$ and $\mathbf{R}$. We first study the optimization over $\mathbf{L}$ in a non-convex manner.

4.3.1 . Wall mitigation: Riemannian estimation of $\mathbf{L}$

The problem we are interested in, over $\mathbf{L}$, is then:

$$\min_{\mathbf{L} \in \mathbb{C}_k^{M \times N}} f(\mathbf{L}) = \sum_{i,j} H_c([\mathbf{Y} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R})) - \mathbf{L}]_{ij}) \tag{4.48}$$

where $\mathbb{C}_k^{M \times N} = \{\mathbf{X} \in \mathbb{C}^{M \times N} : \mathrm{rk}\,(\mathbf{X}) = k\}$. Notice that we went from a non-fixed low rank optimization to a fixed-rank constraint. A way to directly tackle the fixed-rank constraint is via Riemannian optimization [Boumal, 2023]. Such geometrical consideration allows for elegant algorithmic solutions, as the space $\mathbb{C}_k^{M \times N}$ forms a Riemannian manifold.

The manifold of fixed-rank matrices

Via the truncated SVD of rank $k \leq n$, we can parameterize a fixed-rank matrix as:

$$\mathbf{L} \overset{\text{TSVD}}{=} \mathbf{U}(\mathbf{\Sigma}\mathbf{W}^H) = \mathbf{U}\mathbf{V}^H \tag{4.49}$$

with $\mathbf{U} \in \mathbb{C}^{M\times k}, \mathbf{\Sigma} \in \mathbb{C}^{k\times k}, \mathbf{V} \in \mathbb{C}^{N\times k}$. Note that (4.49) leads to the subspace projection parameterization of the fixed-rank matrix manifold, described in [Mishra et al., 2012, Absil et al., 2012] whereas another possibility is the embedded one [Boumal, 2023, Vandereycken, 2013]. We shortly introduce the tools needed for optimization via Riemannian gradient Descent (RGD). Since (4.49) is invariant under an orthogonal factor, it gives rise to a quotient manifold:

$$\mathrm{St}(m,k) \times \mathbb{C}_*^{n\times k}/\mathcal{O}(k) \tag{4.50}$$

with the Stiefel manifold $\mathrm{St}(m,k) = \{\mathbf{X} \in \mathbb{C}^{m\times k} : \mathbf{X}^H\mathbf{X} = \mathbf{I}_k\}$, the manifold of full rank matrices $\mathbb{C}_*^{n\times k} = \{\mathbf{C} \in \mathbb{C}^{n\times k} : \mathrm{rk}(\mathbf{C}) = k\}$ and the orthogonal group $\mathcal{O}(k) = \{\mathbf{X} \in \mathbb{C}^{k\times k} : \mathbf{X}^H\mathbf{X} = \mathbf{I}_k\}$. Its tangent space can be decomposed into:

$$\mathrm{T}_{(\mathbf{U},\mathbf{V})}(\mathrm{St}(m,k) \times \mathbb{C}_*^{n\times k}) = \mathrm{T}_{\mathbf{U}}\,\mathrm{St}(m,k) \times \mathbb{C}^{n\times k} \tag{4.51}$$

with the tangent space of the Stiefel manifold $\mathrm{T}_{\mathbf{U}}\,\mathrm{St}(m,k) = \{\mathbf{U}\mathbf{\Omega} + \mathbf{U}_\perp\mathbf{W} : \mathbf{\Omega} \in \mathcal{A}(k), \mathbf{W} \in \mathbb{C}^{(m-k)\times k}\}$ and the tangent space of the manifold of full rank matrices $\mathrm{T}_{\mathbf{U}}\,\mathbb{C}_*^{n\times k} = \mathbb{C}^{n\times k}$ where $\mathcal{A}(k) = \{\mathbf{X} \in \mathbb{C}^{k\times k} : \mathbf{X}^H = -\mathbf{X}\}$ is the set of skew-symmetric matrices of size $k \times k$ which is the orthogonal complement of $\mathcal{O}(k)$. Projection onto the tangent space is then:

$$P^t_{(\mathbf{U},\mathbf{V})}(\dot{\mathbf{U}}, \dot{\mathbf{V}}) = (\dot{\mathbf{U}} - \mathbf{U}\,\mathrm{sym}(\mathbf{U}^H\dot{\mathbf{U}}), \dot{\mathbf{V}}) \tag{4.52}$$

where $\mathrm{sym}(\mathbf{A}) = \frac{1}{2}(\mathbf{A}^H + \mathbf{A})$.

A fiber of this quotient manifold is $\{(\mathbf{UO}, \mathbf{VO}) : \mathbf{O} \in \mathcal{O}(k)\}$ and the vertical space $\mathcal{V}_{(\mathbf{U},\mathbf{V})}$ is the space tangent to the fiber: $\mathcal{V}_{(\mathbf{U},\mathbf{V})} = \{(\mathbf{U\Omega}, \mathbf{V\Omega}) : \mathbf{\Omega} \in \mathcal{A}(k)\}$. This parametrization may be endowed with the metric:

$$\bar{g}_{(\mathbf{U},\mathbf{V})}((\dot{\mathbf{U}}, \dot{\mathbf{V}}), (\tilde{\mathbf{U}}, \tilde{\mathbf{V}})) = \mathrm{tr}(\dot{\mathbf{U}}^H\tilde{\mathbf{U}}) + \mathrm{tr}((\mathbf{V}^H\mathbf{V})^{-1}\dot{\mathbf{V}}^H\tilde{\mathbf{V}}) \tag{4.53}$$

where the first term is the standard Euclidean metric and the second term the natural metric on full rank matrices which renders it invariant to a change of basis. The horizontal space $[h]_{(\mathbf{U},\mathbf{V})}$ which we want to work in, is then the orthogonal complement to $\mathcal{V}_{(\mathbf{U},\mathbf{V})}$ w.r.t. the metric, which gives:

$$\begin{aligned}[h]_{(\mathbf{U},\mathbf{V})} = \{(\dot{\mathbf{U}}, \dot{\mathbf{V}}) \in \mathbb{C}^{m\times k} \times \mathbb{C}^{n\times k} :\\ \mathbf{U}^H\dot{\mathbf{U}} \in \mathcal{A}(k), \mathbf{U}^H\dot{\mathbf{U}} + \mathbf{V}^H\dot{\mathbf{V}} \in \mathcal{O}(k)\}\end{aligned} \tag{4.54}$$

Note that we can write the projection onto the horizontal space and along the vertical space, for some $\mathbf{\Omega} \in \mathcal{A}(k)$ as:

$$P^h_{(\mathbf{U},\mathbf{V})}(\dot{\mathbf{U}}, \dot{\mathbf{V}}) = (\dot{\mathbf{U}} - \mathbf{U\Omega}, \dot{\mathbf{V}} - \mathbf{V\Omega}) \tag{4.55}$$

Using the property that $P^h_{(\mathbf{U},\mathbf{V})}(\dot{\mathbf{U}}, \dot{\mathbf{V}}) \in [h]_{(\mathbf{U},\mathbf{V})}$ it follows after some rearrangement that we may obtain $\mathbf{\Omega}$ by solving a nested symmetric Lyapunov equation:

$$(\mathbf{V}^H\mathbf{V})\tilde{\mathbf{\Omega}} + \tilde{\mathbf{\Omega}}(\mathbf{V}^H\mathbf{V}) = 2\,\mathrm{skew}((\mathbf{V}^H\mathbf{V})(\dot{\mathbf{U}}^H\mathbf{U})(\mathbf{V}^H\mathbf{V})) - 2\,\mathrm{skew}((\dot{\mathbf{V}}^H\mathbf{V})(\mathbf{V}^H\mathbf{V})) \tag{4.56a}$$

$$\tilde{\mathbf{\Omega}} = (\mathbf{V}^H\mathbf{V})\mathbf{\Omega} + \mathbf{\Omega}(\mathbf{V}^H\mathbf{V}) \tag{4.56b}$$

where $\mathrm{skew}(\mathbf{A}) = \frac{1}{2}(\mathbf{A}^H - \mathbf{A})$. Finally, we introduce a retraction of horizontal vectors onto the manifold. In our case, it can be decomposed in terms of the retractions of the components $\mathrm{St}(m,k)$ and $\mathbb{C}_*^{n\times k}$ which can be found in [Absil et al., 2008, section 4.1.2]. For $(\bar{\mathbf{U}}, \bar{\mathbf{V}}) \in [h]_{(\mathbf{U},\mathbf{V})}$, it is:

$$\mathrm{R}_{(\mathbf{U},\mathbf{V})}(\bar{\mathbf{U}}, \bar{\mathbf{V}}) = (\mathrm{uf}(\mathbf{U} + \bar{\mathbf{U}}), \mathbf{V} + \bar{\mathbf{V}}) \tag{4.57}$$

where uf extracts the unitary factor (of the polar decomposition) of a full column rank matrix.

Algorithmic solution

We may use RGD with the addition of quotient space considerations. RGD for quotient manifold has j^{th} iteration:

$$(\mathbf{U},\mathbf{V})_{j+1} = \mathrm{R}_{(\mathbf{U},\mathbf{V})_j}(-\alpha_j P^h_{(\mathbf{U},\mathbf{V})_j}(P^t_{(\mathbf{U},\mathbf{V})_j}(\nabla f((\mathbf{U},\mathbf{V})_j)))) \tag{4.58}$$

with ∇f denoting the *Euclidean* gradient of f and α_k a step size found by line-search. $P^t_{(\mathbf{U},\mathbf{V})}$ is the projection from ambient space to tangent space while $P^h_{(\mathbf{U},\mathbf{V})}$ is the projection from the tangent space to the horizontal space and $\mathrm{R}_{(\mathbf{U},\mathbf{V})}$ denotes the retraction of a horizontal vector to the manifold (notions we expand on in the next section) at the point $(\mathbf{U},\mathbf{V})$. Indeed, the horizontal space is the 'interesting' part of the quotient manifold as horizontal vectors may represent the underlying abstract tangent vectors of the quotient manifold at some point.

Proposition 8. *The Euclidean gradient is found via Wirtinger calculus as* $\nabla f = (\frac{\partial f}{\partial \mathbf{U}^*}, \frac{\partial f}{\partial \mathbf{V}^*})$ *with:*

$$\frac{\partial f}{\partial \mathbf{U}^*} = \sum_{ij} H'_c([\mathbf{U}\mathbf{V}^H - \tilde{\mathbf{Y}}]_{ij})([\mathbf{J}^{mn}\mathbf{V}^T]_{ij})_{mn} \tag{4.59}$$

$$\frac{\partial f}{\partial \mathbf{V}^*} = \sum_{ij} H'_c([\mathbf{U}\mathbf{V}^H - \tilde{\mathbf{Y}}]^*_{ij})([\mathbf{J}^{mn}\mathbf{U}^T]_{ji})_{mn} \tag{4.60}$$

where $\tilde{\mathbf{Y}} = \mathbf{Y} - \mathbf{\Psi}(\mathbf{I} \otimes \mathbf{r})$ *and* $\mathbf{J}^{mn}$ *is the single-entry matrix which has* 1 *at the* $(m,n)^{th}$ *entry and* 0 *elsewhere. Moreover* $[\mathbf{A}]_{ij}$ *extracts the* $(i,j)^{th}$ *entry of* $\mathbf{A}$ *whereas* $(\mathbf{A})_{mn}$ *constructs a matrix entry by entry.*

Proof. Let the problem be:

$$\min_{\mathbf{U},\mathbf{V}} \; f(\mathbf{U},\mathbf{V}) = \sum_{ij} H_c([\mathbf{U}\mathbf{V}^H - \tilde{\mathbf{Y}}]_{ij}) \tag{4.61}$$

Denote $z = [\mathbf{U}\mathbf{V}^H - \tilde{\mathbf{Y}}]_{ij}$. Then via Wirtinger calculus and its chain rule, we find the steepest ascent direction:

$$\frac{\partial f}{\partial \mathbf{U}^*} = \sum_{ij} \left(\frac{\partial H_c(z,z^*)}{\partial z} \frac{\partial z}{\partial \mathbf{U}^*} + \frac{\partial H_c(z,z^*)}{\partial z^*} \frac{\partial z^*}{\partial \mathbf{U}^*} \right) \tag{4.62}$$

$\frac{\partial H_c(z,z^*)}{\partial z^*}$ is equivalent to $H_c'(z)$, the derivative of the real Huber function, while $\frac{\partial H_c(z,z^*)}{\partial z}$ is equivalent to $H_c'(z^*)$. Note that $\frac{\partial z}{\partial \mathbf{U}^*} = \frac{\partial [\mathbf{U}\mathbf{V}^H - \tilde{\mathbf{Y}}]_{ij}}{\partial \mathbf{U}^*} = \mathbf{0}$. Moreover $\frac{\partial z^*}{\partial [\mathbf{U}^*]_{mn}} = \frac{\partial [\mathbf{U}^*\mathbf{V}^T - \tilde{\mathbf{Y}}]_{ij}}{\partial [\mathbf{U}^*]_{mn}} = [\mathbf{J}^{mn}\mathbf{V}^T]_{ij}$ We can then construct the whole matrix $\frac{\partial z^*}{\partial \mathbf{U}^*}$ elementwise. Thus:

$$\frac{\partial f}{\partial \mathbf{U}^*} = \sum_{ij} H_c'([\mathbf{U}\mathbf{V}^H - \tilde{\mathbf{Y}}]_{ij})([\mathbf{J}^{mn}\mathbf{V}^T]_{ij})_{mn} \tag{4.63}$$

And similarly:

$$\frac{\partial f}{\partial \mathbf{V}^*} = \sum_{ij} H_c'([\mathbf{U}\mathbf{V}^H - \tilde{\mathbf{Y}}]_{ij}^*)([\mathbf{J}^{mn}\mathbf{U}^T]_{ji})_{mn} \tag{4.64}$$

□

4.3.2 . Target detection: Sparse r-step via PGD

The target detection is achieved via convex relaxation. No sparse Riemannian manifold exist, we therefore cannot use RGD or other riemannian methods for this step. We could use a row-wise hard-thresholding [Min et al., 2023, Definition 2] but it showed to underperform while other non-convex methods [Zhang et al., 2023] use a least squares data fitting. We thus resort to the classical convex relaxation via the $\ell_{2,1}$-norm. It is then possible to use proximal gradient descent (PGD) for the minimization over this variable. We will denote this method as HBCD for Huber-BCD. The minimization problem over $\mathbf{R}$ in regularized form is:

$$\min_{\mathbf{R}} \quad \sum_{p_i \in \mathcal{P}} H_c(\|[\mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathrm{vec}(\mathbf{R}))]_{p_i}\|_F) + \lambda \|\mathbf{R}\|_{2,1} \tag{4.65}$$

We consider the vectorized variable $\mathbf{r} = \mathrm{vec}(\mathbf{R})$ in order to compute the gradient which we then unvectorize in order to apply the proximal step. At iteration $t+1$, we have:

$$\mathbf{R}_{t+1} = T_{\lambda s}\left(\mathrm{vec}^{-1}\left(\mathbf{r}_t - s\mathbf{g}_t\right)\right) \tag{4.66}$$

where T is the proximal of the $\ell_{2,1}$-norm, s is a step-size that can be found by line-search (which does not vary over iterations in practice so that it can be fixed) and $\mathbf{g}$ is the needed gradient of the robust fitting term that we already derived:

$$\mathbf{g} = -\sum_{p_i \in \mathcal{P}} \frac{H_c'(\|[\mathbf{E}]_{p_i}\|_F)}{\|[\mathbf{E}]_{p_i}\|_F} \left(\sum_{(j,k)\in p_i} [\mathbf{E}]_{j,k} (\mathbf{\Psi}_k)^H_{j,:} \right) \tag{4.67}$$

where $\mathbf{E} = \mathbf{Y} - \mathbf{L} - \mathbf{\Psi}(\mathbf{I}_N \otimes \mathbf{r})$ and $(\mathbf{\Psi}_k)_{j,:}$ denotes the j^{th} line of $\mathbf{\Psi}_k$.

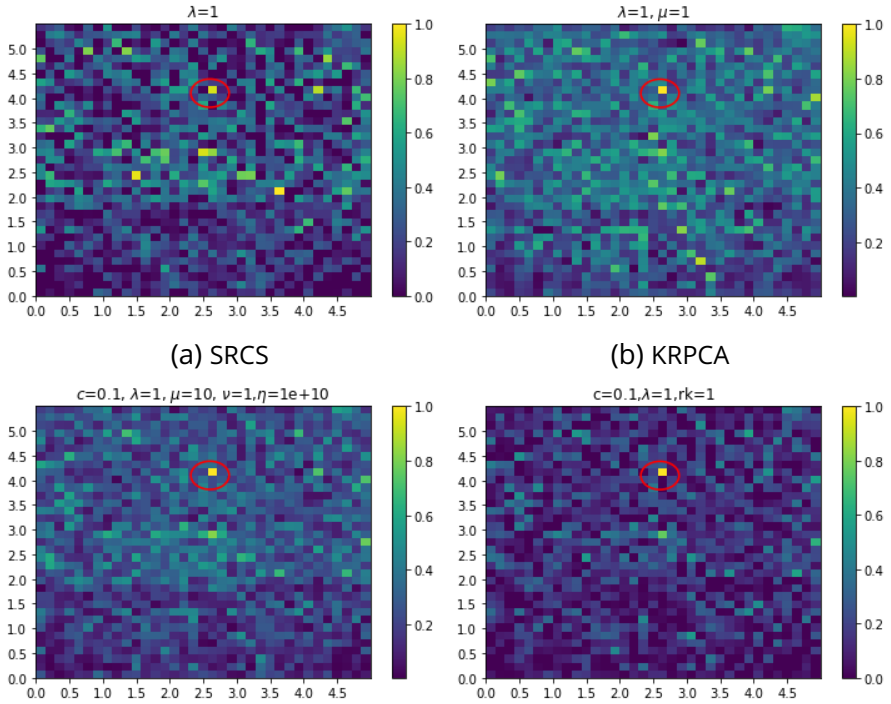
Algorithm 14 HBCD

1: Have: $\{\mathbf{y}_i\}_{i=1}^N, \{\mathbf{\Psi}_i\}_{i=1}^N$
2: Choose: λ
3: $\mathbf{Y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_N]$
4: $\mathbf{\Psi} \triangleq [\mathbf{\Psi}_1, \mathbf{\Psi}_2, \dots \mathbf{\Psi}_N]$
5: Initialize: $\mathbf{L}, \mathbf{R}$
6: **repeat**:
7: Update $\mathbf{L}$ via RGD (4.58)
8: Update $\mathbf{R}$ via PGD (4.66)
9: **until** stopping criterion is met

4.4 . Simulations

We test the method on Finite-Difference Time-Domain [Yee, 1966] (FDTD) simulated data obtained via GprMax [Warren et al., 2016] while the Riemannian optimization is carried out with Pymanopt [Townsend et al., 2016] which we completed by transposing the real manifolds to the complex case. We compare SRCS [Amin and Ahmad, 2013] as well as KRPCA [Brehier et al., 2022a] and HKRPCA to the method of this section (HBCD) with rank fixed to either 1 or 2 (as denoted by the suffixes rk1 or rk2). We generate heterogeneous noise following a student-t distribution with 2.1 degrees of freedom (d.f.) which is a renowned distribution having heavier tails [Ollila et al., 2012] than the normal distribution (for finite d.f.). We consider a point-wise structure of the noise, whereas column-wise (by radar position) may be alternatively considered. Sample detection maps are displayed in Figure 4.9 which show promising results: it appears that the method HBCD proposed here better handles the heterogeneous noise (which follows here a student-t distribution with 2.1 degrees of freedom (d.f.) which is a renowned distribution having heavier tails than the normal distribution for finite d.f.). We perform a quantitative study by constructing the Receiver Operator Characteristic (ROC) of the different methods. Each point on the curve is averaged over 60 Monte-Carlo trials. At

a specific SNR we observe the better performance of the method proposed here.



(a) SRCS (b) KRPCA

(c) HKRPCA FD-pt (d) HBCD

Figure 4.9: Sample detection maps with student-t noise with 2.1 d.f. and SNR of 10 dB (one target with location circled in red)

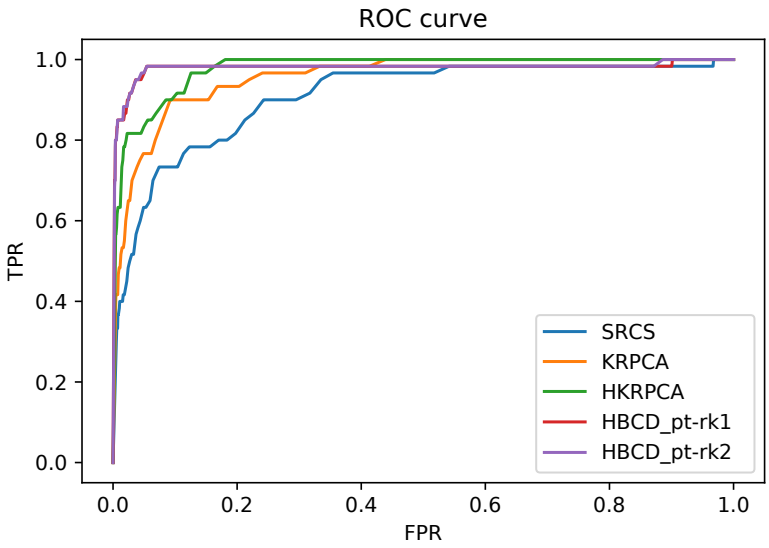


Figure 4.10: ROC with student-t noise with 2.1 d.f. and 60 Monte-Carlo samples and SNR = 10 dB

4.5 . Conclusion

We studied a new method of one-step localization of targets in the context of TWRI called HKRPCA. It is designed to be robust to heterogeneous noise and outliers. The proposed resolution relies on the ADMM framework with two distinct algorithms tailored. On the one hand, a single split of the variable comprising the wall returns results in a closed form proximal evaluation. On the other hand, an additional split of the variable comprising the target returns lends itself to tailored MM step. We show on FDTD simulated data, in more complex scenarios where the noise is heterogeneous or outliers are present, that our method achieves better performance than the state-of-the-art. We then developed an extension which leverages the performance of Riemannian optimization over fixed-rank matrices. Detection results achieved in a standard detection step show its advantage in a context of heterogeneous noise.

We have shown that methods belonging to the framework of low rank and sparse decomposition, i.e. Robust PCA (RPCA), are effective for TWRI detection and localization. However, several issues are raised:

- reliance on a frequency-domain dictionary $\mathbf{\Psi}$ that is very large (and grows in size with the image resolution) which makes it unusable in real-time and requires a lot of memory. To relieve this problem, we may use to a non-dense dictionary, e.g. a convolutional one, which naturally lies in the image domain.

- the underlying model yielding the dictionary is limited as it does not include such things as dispersity (e.g. frequency-dependant permittivity) leading to a model misspecification.

All in all, we may aim for a convolutional dictionary over an image formed from the radar returns. However, it is unknown to us. We may resolve this issue by learning it using a data-driven approach, i.e. learning from data. However, we are limited in the amount of data we have. Obtaining it from FDTD simulations is not so cheap. We may explore the field of hybrid model-based/data-driven approaches to tackle this problem. This is the problem we explore in the next chapter.

5 - Inversion via unrolling

Contents

We underlined at the end of the previous chapter the limitations of the model (associated with a dictionary). We presented the interest of other methods such as Deep Learning (DL) methods, which have shown their effectiveness in Computer Vision (CV). However, these are black-box methods with heavy data usage. We aim to solve these issue by introducing a hybrid model-based and data-driven method. We retain the underlying physical model of TWRI used in RPCA which we mix with Convolutional Sparse Coding (CSC). This allows us to transition to the image domain while restraining the number of parameters to learn, thus reducing the need in data. This is important for our application where data is scarce. In turn, this allows the use of a non-dense convolutional dictionary, lighter but unknown in practice. The learning process is then specifically aimed at learning this dictionary. This chapter's work is associated with [Brehier et al., 2024a].

5.1 . Towards image domain processing and deep methods

5.1.1 . CSC: from optimization...

CSC [Wohlberg, 2016] can be seen as the analog to the standard sparse recovery/coding (i.e. the problem LASSO tackles) but with local structure. On

some image $\mathbf{Y} \in \mathbb{R}^{P_x \times P_z}$, it may be written as:

$$\begin{aligned} &\min_{\{\mathbf{R}_k\}_{k=1}^K} \lambda \sum_{k=1}^K \|\mathbf{R}_k\|_1 \\ &\text{s.t. } \mathbf{Y} = \sum_{k=1}^K \mathbf{\Psi}_k * \mathbf{R}_k \end{aligned} \tag{5.1}$$

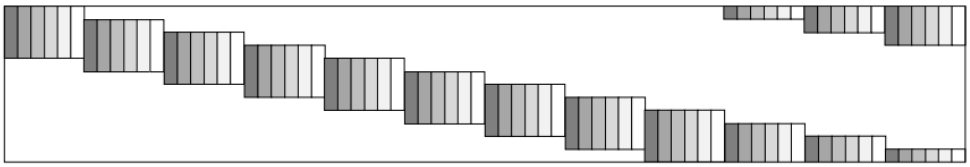
where $*$ denotes a convolution and with $\{\mathbf{\Psi}_k\}_{k=1}^K$ a collection of K (small) convolution filters while $\{\mathbf{R}_k\}_{k=1}^K$ is a collection of sparse activation maps (the same size as the input image $\mathbf{Y}$) which we aim to retrieve. This can be solved using ADMM as described in [Wohlberg, 2016]. CSC can be augmented with a low rank term (e.g. [Gallet et al., 2023]) :

$$\begin{aligned} &\min_{\mathbf{L},\{\mathbf{R}_k\}_{k=1}^K} \mu\|\mathbf{L}\|_* + \lambda \sum_{k=1}^K \|\mathbf{R}_k\|_1 \\ &\text{s.t. } \mathbf{Y} = \mathbf{L} + \sum_{k=1}^K \mathbf{\Psi}_k * \mathbf{R}_k \end{aligned} \tag{5.2}$$

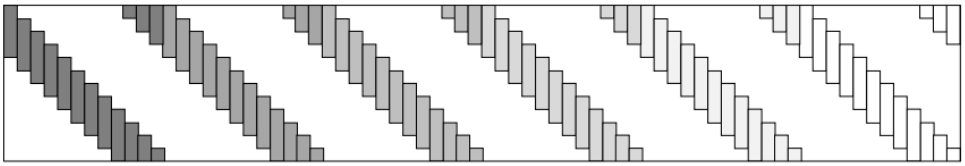
This is an analog to our low rank and sparse model transposed to the image domain: the low rank term will capture the wall contribution. One issue is that the set of convolution kernels is not known as we have not constructed them explicitly. In the literature, this may be tackled with dictionary learning which boils down to adding a step in ADMM over the filters (this method is called Constrained Convolutional Method of Moment (CCMOD) in [Wohlberg, 2016]). However, we will lean towards using a data-driven approach such as DL, specifically those used in CV. Indeed, the main takeaway is that using DL, the filters can be tailored to the objective of target detection in mind, by minimizing a detection loss instead of just unsupervised data reconstruction.

5.1.2 . ...to learning

The low rank plus sparse signal model remains a simplified approximation of the actual undergoing physics: dispersive walls, anisotropic targets, clutter, etc., are not considered. With this in mind, Deep Learning (DL) methods have recently been successfully used for TWRI [Li et al., 2021a, Qu et al., 2022] by learning from the data i.e. changing from a model-driven method to a data-driven one. In order to transition to Convolutional Neural Networks (CNN), we first recall that the convolution structure may be equivalently written in linear form using convolutional matrices (equivalent to a concatenation of banded and circulant matrices) applied on vectorized images [Papyan et al., 2016], which is shown in Figure 5.1). Next, CSC may be modified by cascading the sparse component: we get a multi-layered CSC (ML-CSC)[Papyan et al.,



(a) A convolutional matrix.



(b) A concatenation of banded and Circulant matrices.

Figure 5.1: CSC dictionary structure in matrix form [Papyan et al., 2016]

2016]. For the 2-layer case:

$$\begin{aligned} \mathbf{y} &= \mathbf{\Psi}_1 \mathbf{r}_1 \\ \mathbf{r}_1 &= \mathbf{\Psi}_2 \mathbf{r}_2 \end{aligned} \tag{5.3}$$

where $\mathbf{\Psi}_1, \mathbf{\Psi}_2$ represent CSC filters (in linear form) for two layers while $\mathbf{r}_1, \mathbf{r}_2$ are the associated sparse activations. Then, we propose the following algorithm:

$$\begin{aligned} \hat{\mathbf{r}}_1 &= S_\beta(\mathbf{\Psi}_1^T \mathbf{y}) \\ \hat{\mathbf{r}}_2 &= S_\beta(\mathbf{\Psi}_2^T \hat{\mathbf{r}}_1) \end{aligned} \tag{5.4}$$

where we recall that S is the soft-thresholding operator i.e. the proximal of the ℓ_1 norm (c.f. Proposition 1).

The equation (5.4) is a projection of the measurements on the atoms of the dictionary and threshold them in order to keep the most influential ones and enforce sparsity. This procedure is similar to OMP (recall Algorithm 2). In short:

$$\hat{\mathbf{r}}_2 = S_\beta(\mathbf{\Psi}_2^T S_\beta(\mathbf{\Psi}_1^T \mathbf{y})) \tag{5.5}$$

This is very close to the architecture of a CNN, although the optimization is very different. Indeed we may write the output $\mathbf{r}_2$ of a generic 2-layer CNN as:

$$\mathbf{r}_2 = \mathrm{ReLU}(\mathbf{\Psi}_2^T \, \mathrm{ReLU}(\mathbf{\Psi}_1^T \mathbf{y} + \mathbf{b}_1) + \mathbf{b}_2) \tag{5.6}$$

with $\mathrm{ReLU}(x) = \max(x, 0)$ the so-called Rectified Linear Unit operator. For analysis purposes, the max-pooling (i.e. subsampling to the maximal value of a neighbourhood) can be seen as convolutional layer with increased stride while a fully connected layer can be seen as a special case of a convolutional

layer with filter size being the size of its input. Notice how restricting the soft thresholding operator $S_\beta(x) = \text{sgn}(x)(|x| - \beta)_+$ to be non-negative yields (for a real argument x):

$$S_\beta^+(x) = \begin{cases} x - \beta & \text{if } x > \beta \\ 0 & \text{if } x \leq \beta \end{cases} = \text{ReLU}(x - \beta) \tag{5.7}$$

Thus, changing the thresholding operator, ML-CSC rewrites as :

$$\begin{aligned} \hat{\mathbf{r}}_2 &= S_\beta^+(\mathbf{\Psi}_2^T S_\beta^+(\mathbf{\Psi}_1^T \mathbf{y})) \\ &= \text{ReLU}(\mathbf{\Psi}_2^T \,\text{ReLU}(\mathbf{\Psi}_1^T \mathbf{y} - \mathbf{b}_1) - \mathbf{b}_2) \end{aligned} \tag{5.8}$$

which is similar to the aforementioned 2-layers CNN. However, this is only similar to a forward model (the forward pass of the CNN). In the backward pass, a CNN is typically optimized w.r.t. to some objective function (called the loss) that models a task based on the output of the convolution layers (the innermost representation). This can be a detection task hence a detection loss such as binary cross-entropy may be used. Hence, The convolutional structure acts a feature extractor to serve for some task. Typically, this backward pass is achieved by the so-called Back-Propagation (not to be confused with Back-Projection...) which updates each parameter by stochastic gradient descent on the loss using the chain rule. There exist some variants: with additional momentum/extrapolation or with adaptive weights/ weight decay (e.g. the popular Adam optimizer [Kingma and Ba, 2015]).

5.1.3 . U-Net

A variant that we will compare to is the U-Net [Ronneberger et al., 2015], a form of Fully Convolutional Network (FCN) as used in [Li et al., 2021a] in the context of TWRI. It is very popular to achieve image to image transformation. It is characterized by a symmetric, U-shaped structure consisting of two main parts: the contracting path and the expansive path as shown in Figure 5.2. Unlike autoencoders, U-net contains shortcut connections from the contracting path to the expansive path which makes it efficient for image segmentation tasks. Typically, the contracting path consists of convolutions followed by max-pooling. It may be modified by using strided convolutions instead of max-pooling which has the same effect of reducing the output size. The expansive path consists of transposed convolutions. Their output is concatenated with the result of the contracting path of the same size to maintain both spatial and semantic information. This may also be enhanced by adding an attention mechanism [Vaswani et al., 2017].

The attention gate in [Li et al., 2021a] is shown in Figure 5.3: it consists in processing both inputs by pointwise convolutions (1×1 kernels), added elementwise. A ReLU activation is used and a further pointwise convolution is followed by a sigmoid to form weights that represent the attention at each

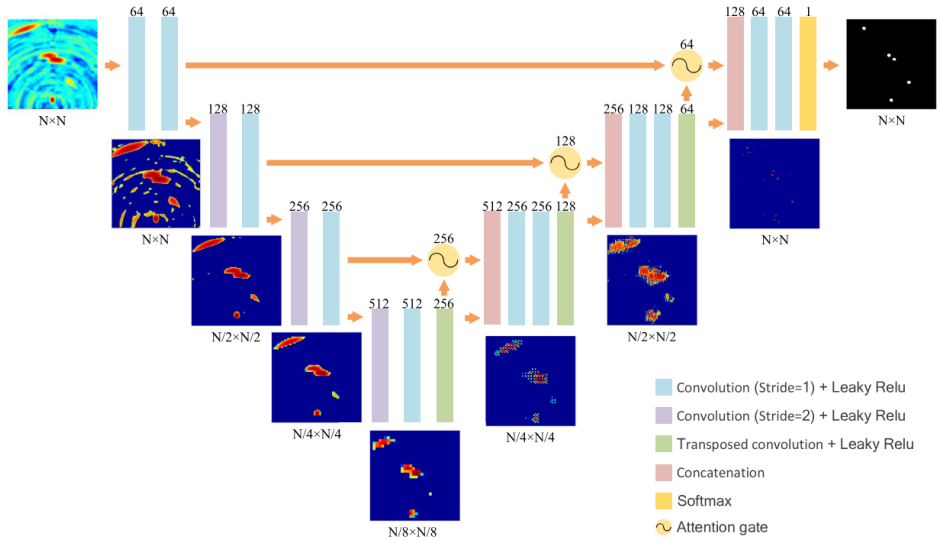


Figure 5.2: Special case of a U-Net [Ronneberger et al., 2015]: the Attention U-Net [Li et al., 2021a]

pixel. Residual connections such as this one have the known effect of allowing for deeper networks by stabilizing the gradient through the weights. Here, it also enhances the network by combining low-level and high-level information from the different resolutions more effectively.

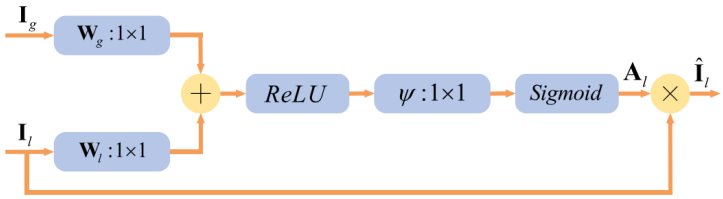


Figure 5.3: The attention gate in[Li et al., 2021a] (inspired by [Vaswani et al., 2017])

The concatenated elements will be the result of the transposed convolution (the expansive path) as well as the result of the contracting path of the same size multiplied by a weight matrix corresponding to attention coefficients.

5.1.4 . Hybrid model-based and data-driven methods: unrolling

One concern we have with these black box model is that their architecture may too generic, not tailored enough for a specific purpose. In turn, this means that they require a large amount of training data samples and computing power in order to fit to the task at hand. An area of study to overcome

this problem is to mix the newer data driven paradigm of deep learning with the traditional model based approach. This has given rise to several methods such as Plug-and-Play (PnP) [Kamilov et al., 2023] or unrolling frameworks [Monga et al., 2021].

PnP methods arise from traditional model based methods with regularization. From a Bayesian perspective, a method such as LASSO effectively retrieves a Maximum A Posteriori (MAP) estimate in the presence of Gaussian noise using a sparsity prior (which might be in a transformed space e.g. Fourier or Wavelet). We have seen that the optimization on the sparsity constraint is achieved using a proximal operator in PGD (Algorithm 3) or ADMM (Algorithm 5). Interpreted as a denoising step, the idea is then to replace the proximal operator (e.g. the soft-thresholding operator) induced by some prior (e.g. sparsity), with a black-box one with unknown prior (e.g. a U-Net pretrained on other data to perform denoising). However, it retains the optimization based procedure for CSC, whereas we want to learn the filters.

We will instead be interested in unrolling, whose fundamental procedure is to truncate a fixed number of iterations of a reference iterative optimization method and convert each operation in these iterations into learnable components of a network. This process maintains the original processing structure where each iteration is now translated into the layer of a network. All layers together then form a valid neural network which can be optimized in standard manner. We delve into this further as we will use this framework.

LISTA

The precursor to unrolling methods is the Learned Iterative Soft Thresholding Algorithm (LISTA) [Gregor and LeCun, 2010]. It is based on ISTA, which is simply PGD (Algorithm 3) applied to the LASSO. Recall the LASSO:

$$\min_{\mathbf{r}} \lambda \|\mathbf{r}\|_1 + \frac{1}{2}\|\mathbf{y} - \mathbf{\Psi}_A \mathbf{r}\|_2^2 \tag{5.9}$$

Which can be solved via the PGD iteration:

$$\begin{aligned} \mathbf{r}_{i+1} &= S_{\lambda t}\left(\mathbf{r}_i + t\mathbf{\Psi}_A^H(\mathbf{y} - \mathbf{\Psi}_A \mathbf{r}_i)\right) \\ &= S_{\lambda t}\left((\mathbf{I} - t\mathbf{\Psi}_A^H \mathbf{\Psi}_A)\mathbf{r}_i + t\mathbf{\Psi}_A^H \mathbf{y}\right) \end{aligned} \tag{5.10}$$

with t a step-size equal to the inverse of the largest eigenvalue of the above gram matrix. The last equation show its closeness to a fully connected (FC) /Multi-Layer Perceptron (MLP) layer. Indeed, we may consider casting this iteration to a FC layer with weights $\mathbf{W}_1, \mathbf{W}_2$ and non-linear activation S_λ (with learnable threshold):

$$\mathbf{r}_{i+1} = S_\lambda\left(\mathbf{W}_1 \mathbf{r}_i + \mathbf{W}_2 \mathbf{y}\right) \tag{5.11}$$

Then, we can stack a certain number of such layers to form a MLP that mimics the source algorithm (ISTA/PGD) for a fixed number of iterations. No-

tice that if the parameters are shared across layers, this resembles a Recurrent Neural Network (RNN) as seen in Figure 5.4.

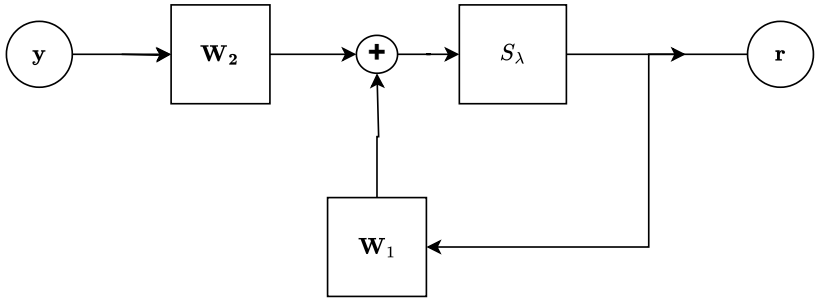


Figure 5.4: LISTA flowchart

The authors in [Gregor and LeCun, 2010] consider it as a way to obtain fast approximation of the iterative method ISTA. In doing so, the loss of training the network is a distance from the results of the iterative methods. Thus, if ISTA yields so-called codes $\mathbf{r}_{\text{ISTA}}$ and the network output is $\hat{\mathbf{r}}$, then the loss may be the Mean Squared Error (MSE) over N samples $\{\mathbf{y}_n\}_{n=1}^N$:

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}\|\mathbf{r}_{\text{ISTA},n} - \hat{\mathbf{r}}_n\|_2^2 \tag{5.12}$$

which is used in Back-Propagation to update the weights $\mathbf{W}_1$ and $\mathbf{W}_2$ as well as the threshold λ of the non-linear activation.

LCSC

An extension to a convolutional structure was proposed and coined Learned CSC (LCSC) in [Sreter and Giryes, 2018] by using the convolutional matrix structure we saw previously. Recall the CSC problem:

$$\min_{\{\mathbf{R}_k\}_{k=1}^K} \lambda\sum_{k=1}^{K}\|\mathbf{R}_k\|_1 + \frac{1}{2}\left\|\mathbf{Y} - \sum_{k=1}^{K}\mathbf{\Psi}_k * \mathbf{R}_k\right\|_2^2 \tag{5.13}$$

which may be rewritten as suggested in standard linear form, giving a PGD iteration:

$$\mathbf{r}_{i+1} = S_{\lambda t}\left(\mathbf{r}_i + t\mathbf{\Psi}_C^H(\mathbf{y} - \mathbf{\Psi}_C\mathbf{r}_i)\right) \tag{5.14}$$

We may observe that the dictionary $\mathbf{\Psi}_C$ (which is of convolutional form here) is applied once on the codes $\mathbf{r}_k$ and its adjoint $\mathbf{\Psi}_C^H$ is applied once on the residuals $\mathbf{y} - \mathbf{\Psi}_C\mathbf{r}_k$. This implies that we may rewrite the dictionary in convolutional form as well as its adjoint:

$$\mathbf{R}_{i+1} = S_{\lambda t}\left(\mathbf{R}_i + t\mathbf{\Psi}_{\text{adj}} \star \left(\mathbf{Y} - \sum_{k=1}^{K}\mathbf{\Psi}_k * \mathbf{R}_{k,i}\right)\right) \tag{5.15}$$

where $\mathbf{\Psi}_{\text{adj}} \star \mathbf{Y} \triangleq [\text{flip}(\mathbf{\Psi}_1) * \mathbf{Y}, \dots, \text{flip}(\mathbf{\Psi}_K) * \mathbf{Y}]$ with flip having the effect of reversing entries in both directions and $\mathbf{R}_i = [\mathbf{R}_{1,i}, \dots, \mathbf{R}_{K,i}]$ is the concatenation of the K sparse activations maps at iteration i. To transition to a neural network, we may learn the filters:

$$\mathbf{R}_{i+1} = S_\lambda \left(\mathbf{R}_i + \{\mathbf{W}_{2,k}\}_{k=1}^K * \left(\mathbf{Y} - \sum_{k=1}^K \mathbf{W}_{1,k} * \mathbf{R}_{k,i} \right) \right) \tag{5.16}$$

Notice that we relaxed the adjoint constraint by decoupling the set of learned filters $\{\mathbf{W}_{1,k}\}_{k=1}^K$ and $\{\mathbf{W}_{2,k}\}_{k=1}^K$. Also note that we implicitly define the outputs of the set of filters convoluted one-by-one with an input to be concatenated together. LCSC finally adds a third set of filters $\{\mathbf{W}_{3,k}\}_{k=1}^K$ in order to reconstruct the data from the sparse activation maps $\mathbf{R}_I$ of the last iteration I:

$$\hat{\mathbf{Y}} = \sum_{k=1}^K \mathbf{W}_{3,k} * \mathbf{R}_{k,I} \tag{5.17}$$

The whole network is then trained as an auto-encoder:

$$\mathcal{L} = \text{dist}(\mathbf{Y}, \hat{\mathbf{Y}}) \tag{5.18}$$

with $\mathbf{Y}$ denoting the ground truth. Moreover, dist denotes a distance which can be the Euclidean ℓ_2 distance or the ℓ_1 one, although it is suggested to mix them with the Multi-Scale Structural Similarity Index Measure (MS-SSIM) loss [Wang et al., 2003] which specifically measures visual similarity between images. The unsupervised training scheme means that the sparsity is enforced implicitly through the operator S_λ (which is assumed to have $\lambda > 0$).

CORONA

The final method we will mention is Convolutional rObust pRincipal cOmpoNent Analysis (CORONA)[Solomon et al., 2020] which unrolls RPCA for the application of clutter suppression in Ultrasound imaging. It does so by using the PGD approach mentioned in Algorithm 6. Indeed, recall the PGD iteration:

$$\begin{aligned} \mathbf{L} &= D_{t\mu}(\mathbf{L} + t(\mathbf{Y} - \mathbf{L} - \mathbf{S})) \\ \mathbf{S} &= S_{\lambda\mu t}(\mathbf{S} + t(\mathbf{Y} - \mathbf{L} - \mathbf{S})) \end{aligned} \tag{5.19}$$

CORONA extends it with dictionaries/compression matrices $\mathbf{\Psi}_1, \mathbf{\Psi}_2$ which are typically measurement operators in imaging applications. Indeed, the forward model is:

$$\mathbf{Y} = \mathbf{\Psi}_1 \mathbf{L} + \mathbf{\Psi}_2 \mathbf{S} \tag{5.20}$$

And the PGD iteration can be written as:

$$\begin{aligned} \mathbf{L} &= D_{t\mu}\left((\mathbf{I} - t\mathbf{\Psi}_1^H \mathbf{\Psi}_1)\mathbf{L} - \mathbf{\Psi}_1^H \mathbf{\Psi}_2 \mathbf{S} - \mathbf{\Psi}_1^H \mathbf{D} \right) \\ \mathbf{S} &= S_{\lambda\mu t}\left((\mathbf{I} - t\mathbf{\Psi}_2^H \mathbf{\Psi}_2)\mathbf{L} - \mathbf{\Psi}_2^H \mathbf{\Psi}_1 \mathbf{S} - \mathbf{\Psi}_2^H \mathbf{D} \right) \end{aligned} \tag{5.21}$$

A layer of CORONA is then:

$$\begin{aligned}\mathbf{L} &= D_\mu\left(\mathbf{W}_1 * \mathbf{L} + \mathbf{W}_2 * \mathbf{S} + \mathbf{W}_3 * \mathbf{D}\right)\\ \mathbf{S} &= S_\lambda\left(\mathbf{W}_4 * \mathbf{L} + \mathbf{W}_5 * \mathbf{S} + \mathbf{W}_6 * \mathbf{D}\right)\end{aligned} \tag{5.22}$$

with learnable parameters the weights $\{\mathbf{W}_i\}_{i=1}^6$ as well as λ, μ. The training is similar to LISTA with a loss w.r.t. $(\mathbf{L}, \mathbf{S})$ precomputed from an iterative algorithm such as RPCA or from ground truth acquired via simulations. However, it allows the parameters to vary from layer to layer contrary to LISTA. We thus investigate a new method as the aforementioned ones lack in some aspects:

- LCSC was only develop for sparse coding, it doesn't involve a low rank term making it unproprer for our use.
- Corona does not properly source itself from a convolutional model. It simply translate the dense dictionary to a single convolution. It also increases the number of parameters by decoupling every learned operator $\{\mathbf{W}_i\}_{i=1}^6$ without preserving the relations among them (such as being transposes of each other).
- In contrast to both, we aim to leverage the true target position of the training data samples (known in simulations) to make the method truly robust to artefacts from complex radar effects, which may defocus the input BP image.

5.2 . LCRPCA: hybrid model/learning method for TWRI

Recall our problem concerning TWRI: the model with a dense linear dictionary is very heavy, and it does not model some physical effects which can impact the imaging process. We proposed a convolutional structure on the image formed from the signal matrix (also called B-Scan matrix) by some naive method such as BP. In turn, the hybrid model/learning method we aim at developing will clean that image: erase multipath ghosts, refocus the target from dispersion effects, etc. which the BP cannot achieve. This is shown on the flowchart in Figure 5.5.

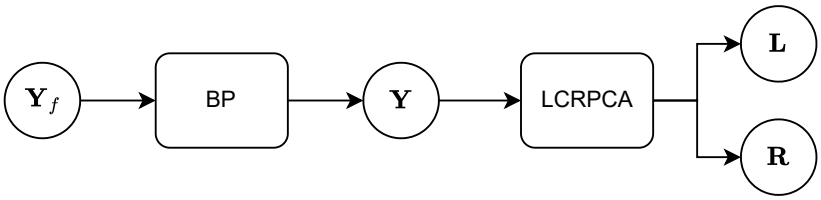


Figure 5.5: LCRPCA pipeline

We consider the CSC model mixed with a low rank component on the image $\mathbf{Y} \in \mathbb{R}^{P_x \times P_z}$, resulting from BP (parametrized by the permittivity ϵ and thickness d of the wall to penetrate) on the B-Scan $\mathbf{Y}_f$ i.e. $\mathrm{BP}_{\epsilon,d}(\mathbf{Y}_f) \triangleq \mathbf{Y}$. This may then be written as:

$$\boxed{\begin{aligned} &\min_{\mathbf{L},\{\mathbf{R}_k\}_{k=1}^K} \mu\|\mathbf{L}\|_* + \lambda \sum_{k=1}^K \|\mathbf{R}_k\|_1 \\ &\text{s.t.} \quad \mathbf{Y} = \mathbf{L} + \sum_{k=1}^K \mathbf{\Psi}_k * \mathbf{R}_k \end{aligned}} \tag{5.23}$$

with $\{\mathbf{\Psi}_k\}_{k=1}^K$ a collection of K (small) convolutional filters and $\{\mathbf{R}_k\}_{k=1}^K$ a collection of sparse activation maps (the same size as the input BP image $\mathbf{Y}$) to retrieve.

5.2.1 . Source algorithm: a composite PGD for CSC/RPCA

We can rewrite this by considering the linear form of CSC, which uses a concatenation of Toeplitz matrices $\mathbf{\Psi}_C$ containing the filters $\{\mathbf{\Psi}_k\}_{k=1}^K$, plus a low rank component as in RPCA:

$$\min_{\mathbf{L},\mathbf{R}} \frac{1}{2}\|\mathrm{vec}(\mathbf{Y}) - \mathrm{vec}(\mathbf{L}) - \mathbf{\Psi}_C \,\mathrm{vec}(\mathbf{R})\|_F^2 + \lambda\|\mathbf{R}\|_1 + \mu\|\mathbf{L}\|_* \tag{5.24}$$

with $\mathbf{L}$ low rank and $\mathbf{R} = [\mathbf{R}_1, \ldots, \mathbf{R}_K]$ the concatenation of the K sparse activation maps. We may use PGD over a composite variable as in Algorithm

6 and CORONA, and return to the convolutional form as in LCSC. We recapitulate this for readers' comprehension. The optimization program may be compacted as:

$$\min_{\mathbf{z}} f(\mathbf{z}) + \frac{1}{2}\|\mathrm{vec}(\mathbf{Y}) - \mathbf{K}\mathbf{z}\|_F^2 \tag{5.25}$$

where :

$$\mathbf{z} = [\mathrm{vec}(\mathbf{L}), \mathrm{vec}(\mathbf{R})]^T \quad \mathbf{K} = [\mathbf{I}, \mathbf{\Psi}_c] \quad f(\mathbf{z}) = \mu\|\mathbf{L}\|_* + \lambda\|\mathbf{R}\|_1 \tag{5.26}$$

The proximal of $f(\mathbf{z})$ is separable in its components and thus computable as:

$$\mathrm{prox}\, f(\mathbf{z}) = [\mathrm{vec}(D_\mu(\mathbf{L})), \mathrm{vec}(S_\lambda(\mathbf{R}))]^T \tag{5.27}$$

The gradient of the differentiable part is readily known as:

$$\nabla_{\mathbf{z}} \frac{1}{2}\|\mathrm{vec}(\mathbf{Y}) - \mathbf{K}\mathbf{z}\|_F^2 = -\mathbf{K}^H(\mathrm{vec}(\mathbf{Y}) - \mathbf{K}\mathbf{z}) \tag{5.28}$$

Giving a PGD iteration that we separate in the components of $\mathbf{z}$:

$$\begin{aligned} \mathrm{vec}(\mathbf{L}) &= \mathrm{vprox}_{\mu t\|\cdot\|_*}(\mathrm{vec}(\mathbf{L}) + t(\mathrm{vec}(\mathbf{Y} - \mathbf{L}) - \mathbf{\Psi}_c\,\mathrm{vec}(\mathbf{R}))) \\ \mathrm{vec}(\mathbf{R}) &= \mathrm{vprox}_{\lambda t\|\cdot\|_1}(\mathrm{vec}(\mathbf{R}) + t(\mathbf{\Psi}_c^H(\mathrm{vec}(\mathbf{Y} - \mathbf{L}) - \mathbf{\Psi}_c\,\mathrm{vec}(\mathbf{R})))) \end{aligned} \tag{5.29}$$

where $\mathrm{vprox}_{\lambda f}(\mathbf{x}) = \mathrm{vec}(\mathrm{prox}_{\lambda f}(\mathrm{vec}^{-1}(\mathbf{x})))$ and t is some stepsize. Then, returning to the original convolutional form yields:

$$\begin{aligned} \mathbf{L} &= D_{\mu t}(\mathbf{L} + t(\mathbf{Y} - \mathbf{L} - \sum_k \mathbf{\Psi}_k * \mathbf{R}_k)) \\ \mathbf{R} &= S_{\lambda t}(\mathbf{R} + t(\{\mathbf{\Psi}_k\} \star (\mathbf{Y} - \mathbf{L} - \sum_k \mathbf{\Psi}_k * \mathbf{R}_k))) \end{aligned} \tag{5.30}$$

5.2.2 . Proposed network: LCRPCA

We are ready to propose an unrolling of the optimization scheme outlined in (5.30) which we will call Learned Convolutional Robust PCA (LCRPCA). This is similar in spirit to CORONA which considers unrolling RPCA algorithms. However, we include a genuine collection of filters instead of replacing one matrix multiplication by a convolution thanks to sourcing the method from CSC as in LCSC, as well as staying closer to the source algorithm by not decoupling all weight matrices. We also train our network differently. A layer of our proposed method is:

$$\begin{aligned} \mathbf{L} &= D_{\mu}(\mathbf{L} + w_0(\mathbf{Y} - \mathbf{L} - \sum_k \mathbf{W}_{1,k} * \mathbf{R}_k)) \\ \mathbf{R} &= S_{\lambda}(\mathbf{R} + \{\mathbf{W}_{2,k}\} * (\mathbf{Y} - \mathbf{L} - \sum_k \mathbf{W}_{1,k} * \mathbf{R}_k)) \end{aligned} \tag{5.31}$$

This is summed up graphically in Figure 5.6.

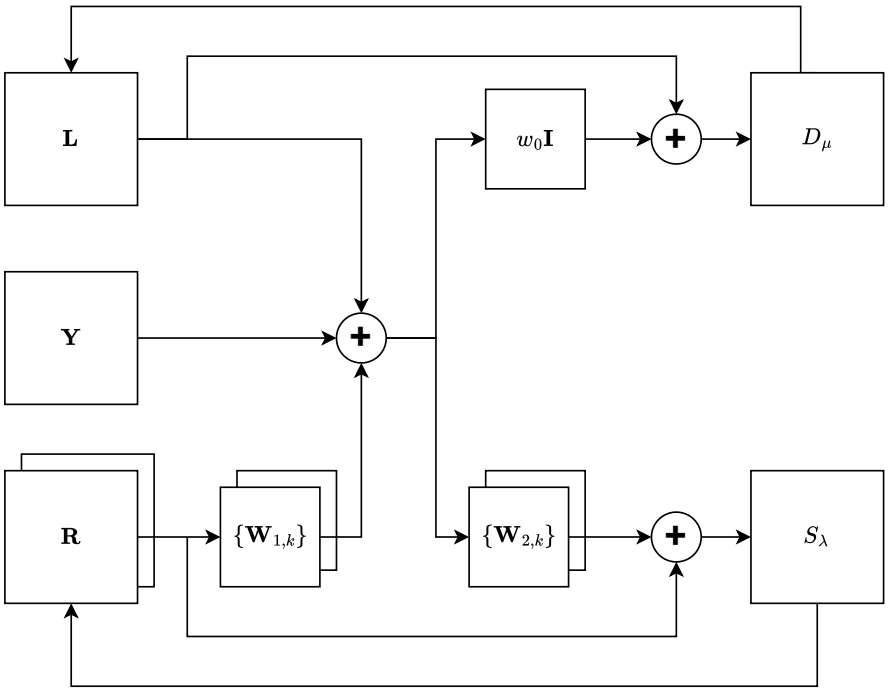


Figure 5.6: LCRPCA flowchart

At the end of the network, an image $\hat{\mathbf{Y}}$ is formed as well as a detection map $\hat{\mathbf{R}}_d$ obtained using the sigmoid operator (denoted σ) on the overall sparse component (instead of taking a pixel-wise average of the sparse activation maps $\{\mathbf{R}_k\}$). This allows us to capture more precisely the shape of the targets via the pattern inscribed in the filters. Accordingly:

$$\hat{\mathbf{Y}} = \mathbf{L} + \sum_k \mathbf{W}_{1,k} * \mathbf{R}_k, \quad \hat{\mathbf{R}}_d = \sigma(\sum_k \mathbf{W}_{1,k} * \mathbf{R}_k) \tag{5.32}$$

The network will learn on training data the set of filters $\{\mathbf{W}_{1,k}\}$ and $\{\mathbf{W}_{2,k}\}$ as well as the scalars w_0 and μ, λ. As for all RPCA methods, we aim at finding components $\mathbf{L}$ and $\mathbf{R}$ that faithfully reconstruct the data. We do not consider ground truths for $\mathbf{L}$ which would necessitate an empty scene for TWRI. However, we may consider having access to the ground truth of the detection map as we control the measurement setup, which we leverage during training. Indeed, the loss $\mathcal{L}$ used during training is a weighted sum of two components: a reconstruction loss in the form of an Euclidean distance and a detection loss as seen in [Li et al., 2021a] in the form of the Cross-Entropy plus the Dice Coefficient (also known as F1-score). It is used to cope with class imbalance as the

target class contains much fewer pixels than the background one:

$$\mathcal{L}(\hat{\mathbf{Y}}, \hat{\mathbf{R}}_d) = \frac{\left\| \mathbf{Y} - \hat{\mathbf{Y}} \right\|_F^2 + \mathrm{CE}(\hat{\mathbf{R}}_d, \mathbf{R}_d)}{P_x P_z} + (1 - \mathrm{DC}(\hat{\mathbf{R}}_d, \mathbf{R}_d)) \tag{5.33}$$

where CE is the cross-entropy loss and DC is the Dice coefficient. The Dice loss is used in addition to the cross-entropy as detection here consists in a binary classification of pixels (target or background) which are unbalanced: there are far more background pixels. the Dice coefficient helps alleviate this. Here, $(\hat{\mathbf{Y}}, \hat{\mathbf{R}}_d)$ is the network output while $(\mathbf{Y}, \mathbf{R}_d)$ is the reference obtained from FDTD simulations: the detection map $\mathbf{R}_d$ is the simulated scene (that we thus know) while $\mathbf{Y}$ is the output returns passed through BP. This loss without explicit references for $\mathbf{L}, \mathbf{R}$ (such as the ones of RPCA) allows the algorithm to learn new representations instead of mimicking pre-obtained ones, as does CORONA.

5.3 . Simulation study

5.3.1 . Setting

The dataset for training is collected via GprMax [Warren et al., 2016] simulations. We consider metallic cylinders varying in radius ($5 - 10$ cm) in number ($1 - 3$ targets in one scene) and in position. We generate 330 different scenes with the same dispersive wall via the multi-pole Debye model [Zhekov et al., 2020].

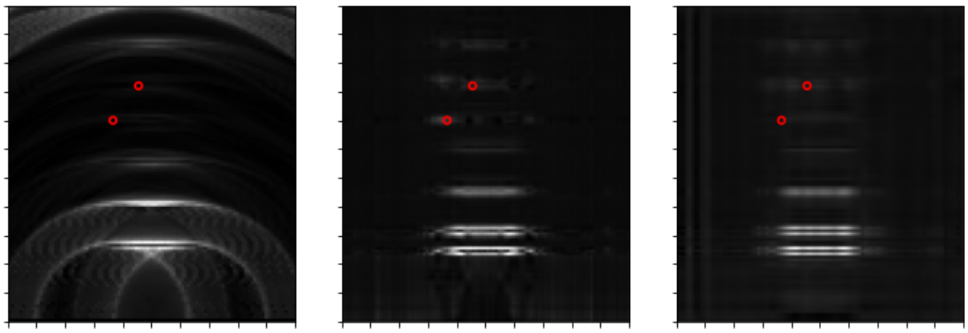
Each scene is then added with 10 different draws of a heterogeneous student-t noise (10 - 30 dB and 2.5-5 d.f.). We then have a dataset of 3300 noised returns. The Train/Validation/Test dataset sizes are respectively 2400, 800 and 100. The implementation of the network was carried out using PyTorch. The proximal of the nuclear uses a SVD whose gradient during backpropagation is unstable when singular values are near zero (something which we try to enforce). We thus use the randomized SVD implementation instead of the standard one. This necessacitates fixing some upper bound on the low rank dimension, which we choose to be 5.

We use a learning rate of 0.001 for 30 epochs and the Adam optimizer [Kingma and Ba, 2015]. We initialize (ie. we feed the first layer of the network) with $\hat{\mathbf{L}} = \mathbf{Y}$ and $\hat{\mathbf{R}}_d = \{\mathbf{0}\}^{64}$. Our method LCRPCA is used with $K = 64$ filters, 6 layers of size $2 \times (7 \times 7), 2 \times (5 \times 5), 2 \times (3 \times 3)$ which implies that filters are not shared across layers. For comparison, we use HKRPCA and CORONA which we adapted to our 2D setup as well as the method of [Li et al., 2021a] that relies on a U-Net with attention.

5.3.2 . Visualization

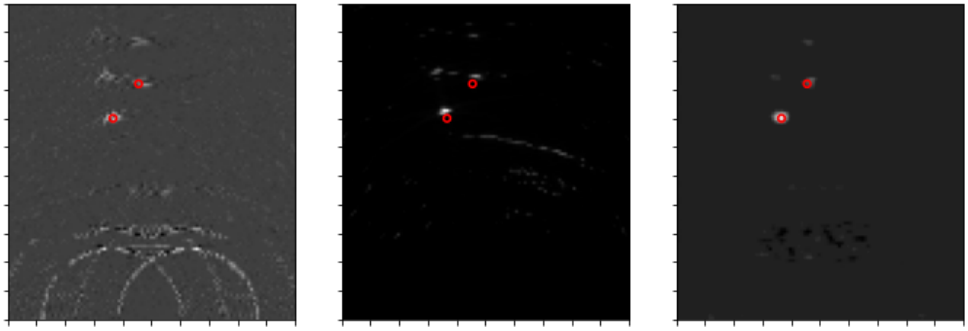
In Figures 5.7,5.8 and 5.9 we show the low rank components $\mathbf{L}$ which are supposed to capture wall returns (the U-Net model does not have a low rank component and is thus not shown). We see that the one of HKRPCA (5.7a) is more complex, with some sprawl on the side. The one of CORONA (5.7b) has more ghosts while the one of LCRPCA (5.7c) is cleaner. We then have sparse components, which are supposed to capture target returns. The one resulting of RPCA on the BP image, which is the input of the DL methods, is in Figure 5.8a. The one of HKRPCA may be seen in Figure 5.8b, where we see some ghosts behind the true targets. The dispersive wall also has the effect of defocusing the detection. The sparse component of CORONA (5.8c) is cleaner but retains a small ghost. Finally, we have the detection maps (Figures 5.9a and 5.9b) of LCPRCA and U-Net which are quite similar.

Otherwise, we may look at some components of the network such as the scalars learned which are displayed in Figure 5.10a. We then see the sparse component evolution across layers in Figure 5.10b. In Figure 5.11 we can observe the decomposition of the sparse component in the feature maps of each filter learned.



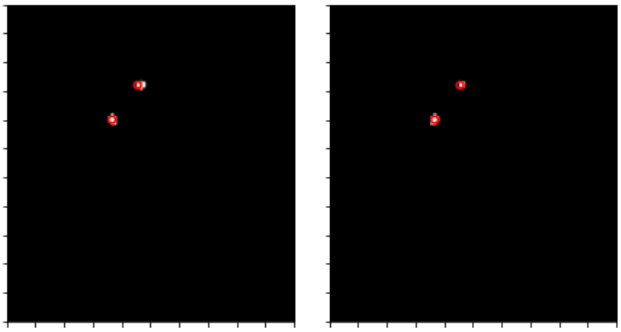
(a) HKRPCA $\hat{\mathbf{L}}$ (b) CORONA $\hat{\mathbf{L}}$ (c) LCRPCA $\hat{\mathbf{L}}$

Figure 5.7: Sample results: low rank components.



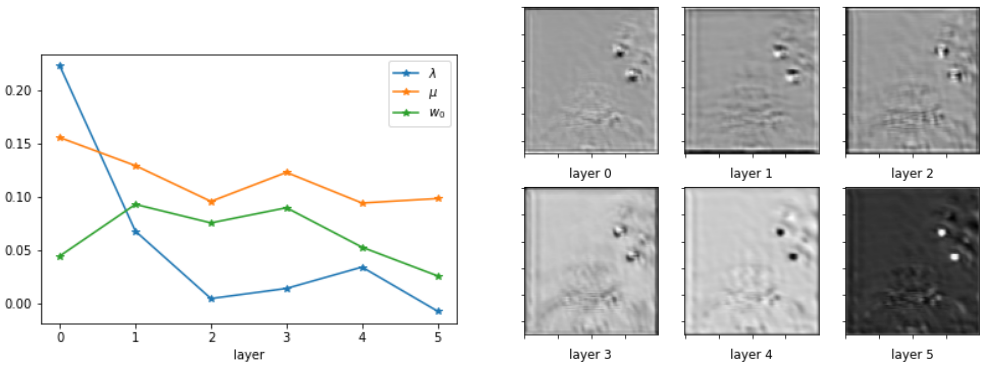
(a) BP-RPCA $\hat{\mathbf{R}}$ (b) HKRPCA $\hat{\mathbf{R}}$ (c) CORONA $\hat{\mathbf{R}}$

Figure 5.8: Sample results: sparse components.



(a) U-Net $\hat{\mathbf{R}}_d$ (b) LCRPCA $\hat{\mathbf{R}}_d$

Figure 5.9: Sample results: detection maps



(a) λ,μ,w_0 across layers (b) Sparse component across layers

Figure 5.10: Comparison of λ, μ, w_0 and sparse component across layers

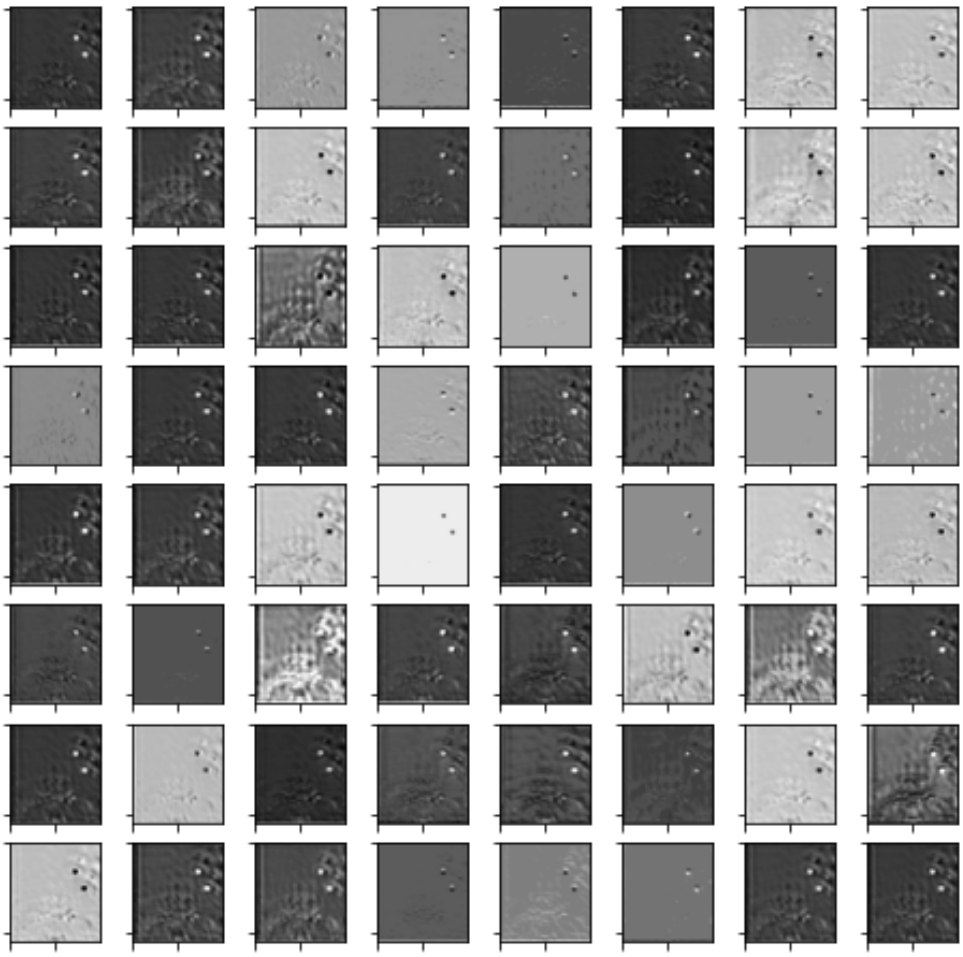


Figure 5.11: Feature map

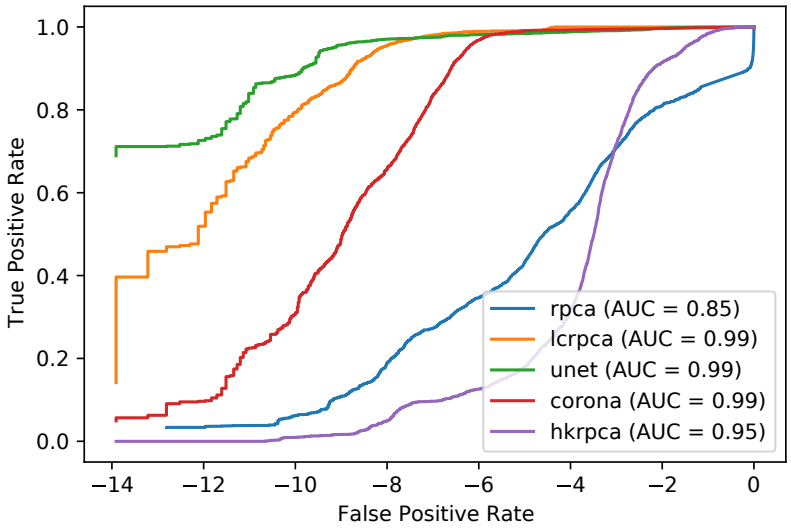
5.3.3 . Performance comparison

We move on to the quantitative evaluation with Receiver Operating Characteristic (ROC) and Precision-Recall curves. The legends are enhanced with the Area Under the Curve (AUC) of the ROC and the Average Precision (AP). We see the same ordering of the methods in the two graphs 5.12a and 5.12b, with the U-Net followed by LCPRCA then CORONA and RPCA methods last. We then compute the same metrics but with 10 % of the training data in Figures 5.13a and 5.13b. The U-Net performance backs down to the level of LCRPCA and even falls off at the tail end of the curves. LCRPCA is better both in AUC and AP. This highlights the better efficiency of our method under a restricted training regime which is an interesting property in our application where data is scarce. Note especially that the U-Net has $8,650,474$ trainable parameters while LCRPCA has $21,656$.

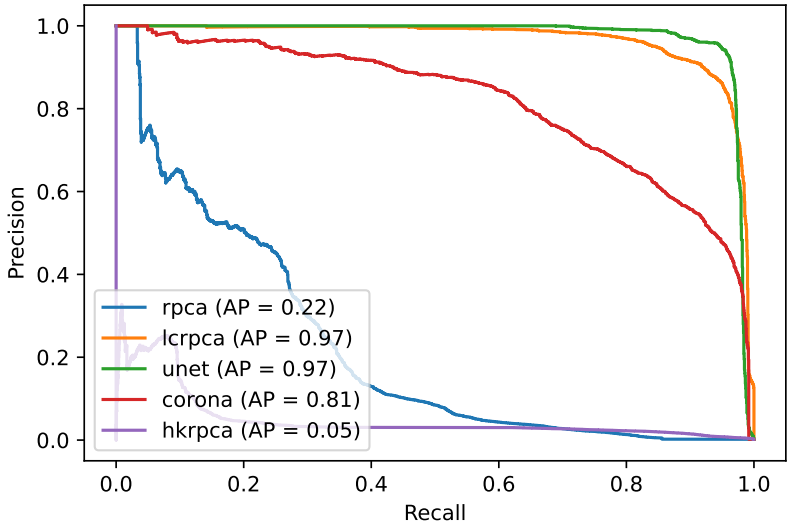
We also show in Table 5.1 the Target to Clutter Ratio (TCR) of the methods. On the first row, we see when the DL methods are trained on 100% of the data where we see the same ordering. The next row is with 10% of the data and shows how LCRPCA and the U-Net get closer. LCRPCA has lost 6dB in TCR while the U-Net has lost 8dB. CORONA, the other DL method generating a low-rank component, totally collapses with a 20dB loss in TCR.

TCR (dB)	HKRPCA	BP-RPCA	Corona	U-Net	LCRPCA
full training data	18.46	19.52	32.01	41.40	39.93
scarce training data	18.46	19.52	12.60	33.78	33.64

Table 5.1: TCR with different trainings

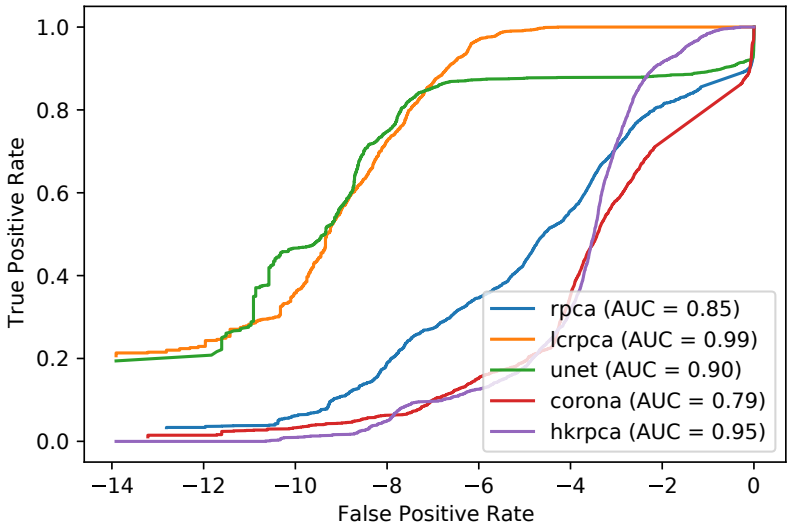


(a) ROC (FPR in log scale) with full training data

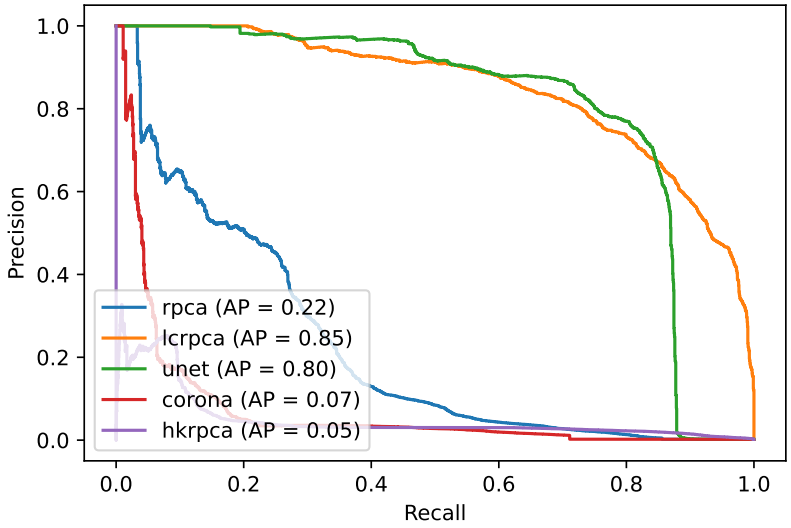


(b) Precision-Recall curve with full training data

Figure 5.12: ROC and PR curves with full training (100% training data)



(a) ROC (FPR in log scale) with scarce training data (10% training data)



(b) Precision-Recall curve with scarce training data (10% training data)

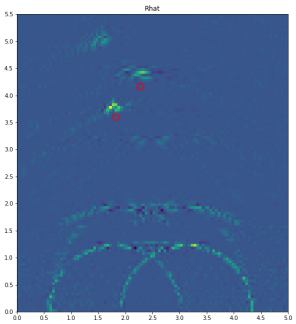
Figure 5.13: ROC and PR curves with scarce training (10% training data)

5.3.4 . Limitations

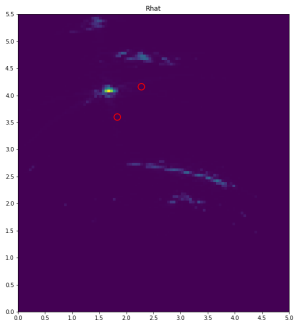
Until now, the scene setting has been the same for the training and test data. Only the target position and number could vary. In practical applications, we would be interested in using the trained network on one setting to another setting. This could mean a wall with different parameters: permittivity, thickness, structure. This is readily understandable to be a complicated undertaking: the returns delay is heavily dependent on the wall characteristics. If those change, without further considerations, this may lead the network to wrong localizations.

Test wall differing from the train wall

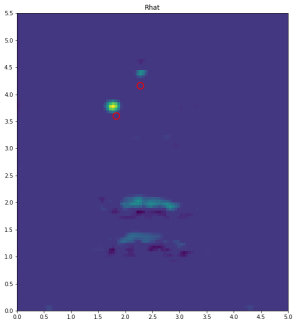
We first evaluate how a different wall would impact the imaging, if we naively used a network trained on another wall. We change the wall thickness from 20cm to 40cm and relative permittivity from 8 to 6 as well as the dispersivity profile corresponding to concrete with large gravels rather than concrete with small gravels. The results are presented in Figure 5.14. We can observe a shift in target localization as well as ghosts target in all the considered methods. The original BP is clearly perturbed which impacts all the data-driven methods which have it as input image and cannot correct it, having seen only a different wall in training.



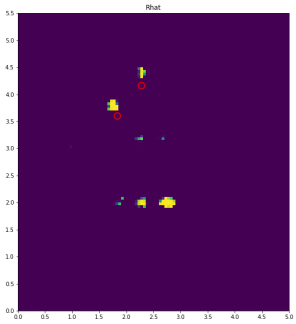
(a) BP RPCA



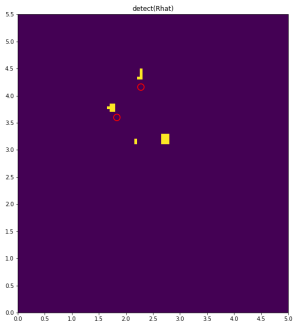
(b) HKRPCA



(c) Corona



(d) UNet



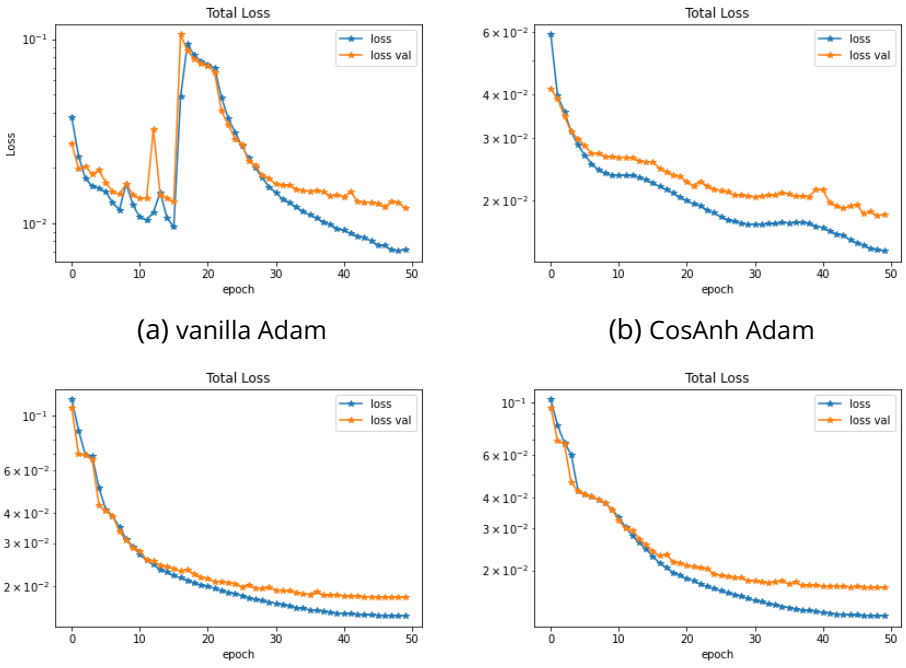
(e) LCRPCA

Figure 5.14: Algorithms with test wall different than the training wall

Two different walls in the training dataset

We then evaluate if we can train LCRPCA on a training dataset composed of two walls. The question is whether the network has the capacity to separate the two walls via a standard training procedure. We noticed that the loss necessitates an adjustment in training procedure to decrease smoothly. Indeed, we adjusted the optimizer with a learning rate scheduler, which modifies the learning rate over iterations. We used the one cycle learning rate scheduler [Smith and Topin, 2018] over the Cosine Annealing learning rate scheduler [Loshchilov and Hutter, 2017]. Moreover, we selected the AdamW optimizer [Loshchilov and Hutter, 2019] over the original Adam [Kingma and Ba, 2015] which is a slight correction on the former. This is empirically described in [Bilogur, 2021].

The evolution of the loss over epoch is shown in Figure 5.15. Sample results are shown in Figure 5.16 and Figure 5.17. We see that the wall returns clearly indicate two different walls while the detection are achieved at correct locations.



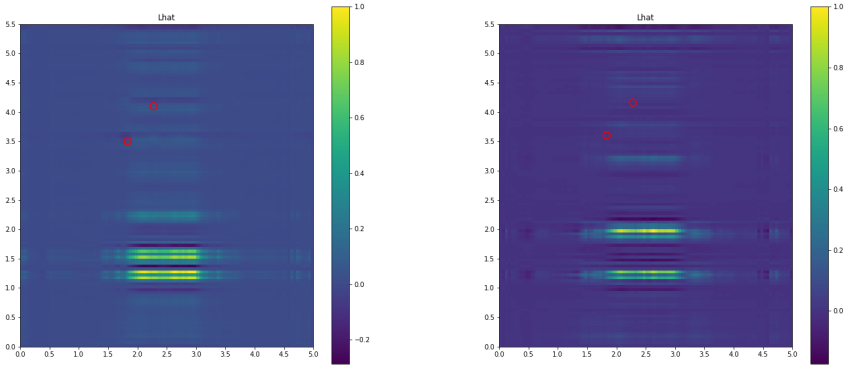
(a) vanilla Adam

(b) CosAnh Adam

(c) OneCycle Adam

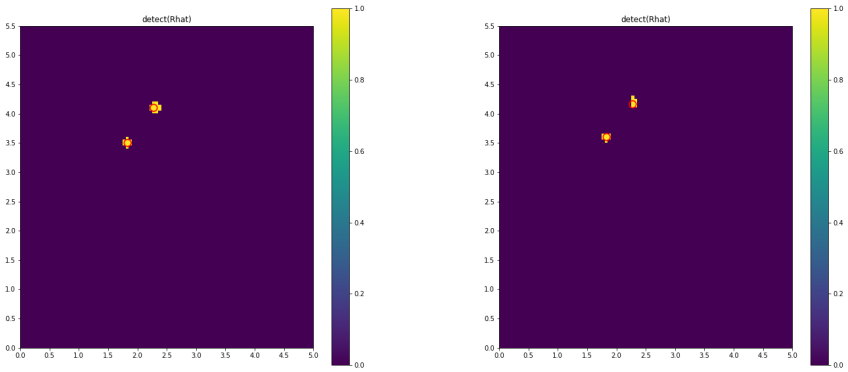
(d) OneCycle AdamW

Figure 5.15: Training with different learning rate schedules



(a) $\mathbf{L}$ associated with wall 1 (b) $\mathbf{L}$ associated with wall 2

Figure 5.16: wall returns with both walls in the training dataset



(a) detection map associated with wall 1 (b) detection map associated with wall 2

Figure 5.17: detection maps with both walls in the training dataset

5.4 . Conclusion

This chapter presented a deep unrolling of RPCA mixed with CSC for TWRI localization of targets. Thanks to its data-driven framework, it proved to outperformed our previous purely model-based methods for TWRI detection. Under limited training schemes, i.e. with less training samples, it showed to be competitive with a U-Net with attention gates. There remain open questions about handling several unknown walls as well as cluttered environments, which we may hope to tackle by extending this method in the future.

Conclusion and perspectives

5.5 . Conclusion

This manuscript explored new imaging methods for Through Wall Radar Imaging, specifically for imaging stationary targets. We started the manuscript by describing a classical acquisition setup consisting in stepped-frequency SAR returns collected along to the wall to penetrate. The classical methodology then consists in mitigating the returns of the wall to penetrate followed by a detection of the targets. The wall mitigation is typically achieved by assuming the returns from the wall span a subspace different than the targets. Effectively, we choose the subspace formed via a subset of singular vectors and project onto its orthogonal complement. Imaging the scene in order to retrieve the position of targets can be done by standard beamforming method, such as Back-Projection, which can be seen as the application of the Point Spread Function of the Radar system. Otherwise, to handle multipath effects, the number of targets can be assumed to be few with respect to the scene dimension. Then, imaging can be done via sparse reconstruction methods. The question raised is that of the optimality of such sequential methods. Indeed, the wall returns that are erased may in fact contain target information.

This started our first work, where we applied low rank and sparse decomposition methods, also known under the name of Robust PCA, to Through Wall Radar Imaging. We reformulated the original vectorized model in a matrix form and tailored the ADMM, a convex optimization framework, to our needs. The ADMM has proven convergence and is widely used for its practical speed when functions are separables, as is our case. We showed on ray-tracing simulations that the method performs better, when considering detection metrics, than the classical sequential method.

We then investigated the possibility of making the method robust to complex noise, specifically when the noise level is not homogeneous over the field of view of the SAR: it is then heterogeneous. In our application, this can happen as SAR acquisitions are made at several positions such that the wall returns are not homogeneous as well as the scene environment behind. We tackled this problematic by introducing a robust distance, which down-weight the contribution of positions and frequencies that are considered as outliers. We developed algorithms for the resolution of this problem and studied their practical properties. We also explored the use of Riemannian optimization to tackle the low rank constraint by using the fixed-rank matrix manifold, as opposed to the more widely used nuclear norm relaxation. Afterwards, we showed on FDTD simulations the performance gain of these methods under heterogeneous noise.

However, these optimization based method suffer from a heavy computation cost. The modelization of the radar returns also do not consider some physical, such as dispersion that distort the signal. We explored a transition of this model based approach to a data driven one. We specifically proposed a hybrid between the two, by considering unrolling networks. We proposed a new network trained in a supervised fashion, with roots in optimization via Convolutional Sparse Coding. This method advantages include its significantly faster execution (once training is achieved), while being able to handle dispersion effects which are hard to model. We underlined how the training on one wall is hard to transfer to another wall. Although the method can be trained on a dataset comprising several walls, the extrapolation capabilities on a never seen wall are underwhelming and should be subject to future research.

5.6 . Perspectives

5.6.1 . Generalization problematic of data driven methods

When using a data driven unrolling method such as LCRPCA, the training data returns are tied to a setting such as a specific wall. However, we would typically like our method to work on another set of returns that are handed to us without associated train samples. This can also represent going from synthetic to real data. Among the domains of research that aim at addressing this generalization problematic, one is called Domain Adaptation (DA). In DA, we have source samples with labels as well as unlabelled target samples. The goal is to predict the target labels.

The first type of methods consist in **aligning the distributions of the features** extracted from samples before the last step (classification, detection...). This is done considering all samples together, as there is no one-to-one correspondence between target and source samples. One popular distance between distributions is the Maximum Mean Discrepancy (MMD) [Gretton et al., 2006] which measures the distance between the samples mean embedded in a feature space. Several methods exist based on this: Transfer Component Analysis (TCA) [Pan et al., 2011] or Deep Adaptation Networks (DAN) [Long et al., 2015]. Other methods exist based on other means of feature alignement: Optimal Transport (OT) methods [Courty et al., 2017b, Courty et al., 2017a] or Domain-Adversarial Neural Networks (DANN) [Ganin et al., 2016].

Other methods use an **alignment in the pixel space** such as CycleGAN [Zhu et al., 2017], Cycada [Hoffman et al., 2018] or Fourier Domain Adaptation (FDA) [Yang and Soatto, 2020]. Interestingly, those last three were evaluated on the task of semantic segmentation, which is the task of classyfing each pixel of an image. In fact, this is what we are interested in since we classify each pixel of the BP image as target or not (binary case).

We rapidly tested FDA to see if it can make LCRPCA transferable to a different test wall. However, the experiment was not successful and FDA did not seem to be able to improve on the simple training (using only source data) while the mixed training (both walls in the training dataset) would be the objective to attain. Altough this may not be tractable for testing on a different wall, the subject of DA remains interesting to optimize performance on real data after training on simulated samples. For example, to handle the real antenna compared to the one simulated which is often simpler. For a large variety of walls, it remains to be studied if a single network has the capacity to handle them simultaneously and obtain interpolation capabilities on never seen walls.

5.6.2 . Other perspectives

It should be emphasized that these methods cover a broad range of methodologies: from sequential classical imaging to alternating optimization methods, from recent convex optimization frameworks to Riemannian optimization, from model-based approaches to data driven approaches. The work done here may be useful in the comparison it allows over a diversity of methods on a single common application. The underlying frameworks can also be transferred to other Radar applications, such as Ground Penetrating Radar to image the subsurface. Some aspects are also shared with other Computational Imagery applications such as Magnetic Resonance Imaging.

We point out that this work may be continued in different paths. This concerns the applied mathematics/processing domain and not the broader research that may be led in physics, antenna for TWRI. Concerning the optimization methods, they might be extended to a three dimensional spatial setting by considering tensor decompositions. The model might be extended and rewritten to a non-uniform dielectric constant across the frequency band to capture important effects. The dictionary may also be enhanced thanks to a standard dictionary learning procedure. Another perspective for LCRPCA is its ability to discriminate target which may make it an useful tool for classification of targets: for example based on their material composition and shape. This would allow some clutter from inside the scene to be ignored. All in all, the general objective to move up from the simplified setting usually considered should be a driving force in steering future research towards a useful direction.

A - Wirtinger Calculus

Contents

For the most part, our application is concerned with complex variables. We are therefore interested in the optimization of functions defined on such complex variables. We will see that complex derivability is too restrictive and that we can apply another type of calculus called Wirtinger calculus [Hunger, 2007].

A.1 . Complex differentiability

Definition 12. *Let $\mathcal{A} \subset \mathbb{C}$ be an open set. Then, a function $f : \mathcal{C} \to \mathbb{C}$ is complex differentiable at $z_0 \in \mathcal{A}$ if there exist a limit:*

$$f'(z_0) = \lim_{z \to z_0} \frac{f(z) - f(z_0)}{z - z_0} \tag{A.1}$$

which is independent of the path for which $z \to z_0$.

A function that is complex differentiable on every point of its domain is called holomorphic. It can be shown that the derivative of an holomorphic function is also holomorphic. By induction, it is then also analytic i.e. of class $\mathcal{C}^\infty$ which is a stronger requirement than for a real function to be derivable. Moreover, the basic properties of the differential of a sum, product and composition of two functions arising in the real case remain valid in the complex case. In order to check for complex differentiability, we may check the validity of the necessary conditions called Cauchy-Riemann equations.

Theorem 1. *Let $f(z) = u(z) + jv(z)$ with $u(z), v(z) \in \mathbb{R}$ where $z = x + jy$ with $x, y \in \mathbb{R}$. Then we can express $f(z)$ as $F(x, y) = u(x, y) + jv(x, y)$ where $u(x, y), v(x, y) \in \mathbb{R}$. A necessary condition for $f(z)$ to be holomorphic is that the following system of partial differential equations, called Cauchy-Riemann equations, holds for every point in the domain of f:*

$$\frac{\partial u(x,y)}{\partial x} = \frac{\partial v(x,y)}{\partial y} \text{ and } \frac{\partial u(x,y)}{\partial y} = -\frac{\partial v(x,y)}{\partial x} \tag{A.2}$$

This can be seen directly via the definition of complex differentiability with paths for $z \to z_0$ along both axes i.e. along x or y and equating them.

Additionally, the Cauchy-Riemann equations are sufficient for f being holomorphic if the partial derivatives $u(x,y), v(x,y)$ are continuous.

A.2 . Extension to non-holomorphic functions

A.2.1 . Wirtinger derivatives

The total differential of the function $F(x,y) = f(z)|_{z=x+jy}$ is:

$$\begin{aligned} dF &= \frac{\partial F(x,y)}{\partial x}dx + \frac{\partial F(x,y)}{\partial y}dy \\ &= \frac{\partial u(x,y)}{\partial x}dx + j\frac{\partial v(x,y)}{\partial x}dx + \frac{\partial u(x,y)}{\partial y}dy + j\frac{\partial v(x,y)}{\partial y}dy \end{aligned} \tag{A.3}$$

Note also that:

$$dz = dx + jdy \quad dz^* = dx - jdy \tag{A.4}$$

So that:

$$dx = \frac{1}{2}(dz + dz^*) \quad dy = \frac{1}{2}(dz - dz^*) \tag{A.5}$$

Then, the total differential is:

$$\begin{aligned} dF = &\frac{1}{2}\left(\frac{\partial u(x,y)}{\partial x} + \frac{\partial v(x,y)}{\partial y} + j\left(\frac{\partial v(x,y)}{\partial x} - \frac{\partial u(x,y)}{\partial y}\right)\right)dz \\ &+ \frac{1}{2}\left(\frac{\partial u(x,y)}{\partial x} - \frac{\partial v(x,y)}{\partial y} + j\left(\frac{\partial v(x,y)}{\partial x} + \frac{\partial u(x,y)}{\partial y}\right)\right)dz^* \end{aligned} \tag{A.6}$$

Note that for a holomorphic function (satisfying the Cauchy-Riemann equations), the second term is zero and thus its differential does not depend on dz^*. Anyhow, the total differential can be rearranged:

$$\begin{aligned} dF &= \frac{1}{2}\left(\frac{\partial}{\partial x}(u(x,y) + jv(x,y)) - j\frac{\partial}{\partial y}(u(x,y) + jv(x,y))\right)dz \\ &+ \frac{1}{2}\left(\frac{\partial}{\partial x}(u(x,y) + jv(x,y)) + j\frac{\partial}{\partial y}(u(x,y) + jv(x,y))\right)dz^* \\ &= \frac{1}{2}\left(\frac{\partial}{\partial x} - j\frac{\partial}{\partial y}\right)F(x,y)dz + \frac{1}{2}\left(\frac{\partial}{\partial x} + j\frac{\partial}{\partial y}\right)F(x,y)dz^* \end{aligned} \tag{A.7}$$

In analogy with the total differential for bivariate real functions, this leads us to the following.

Definition 13. *Let the two partial differential operators $\frac{\partial}{\partial z}, \frac{\partial}{\partial z^*}$ be:*

$$\frac{\partial}{\partial z} = \frac{1}{2}\left(\frac{\partial}{\partial x} - j\frac{\partial}{\partial y}\right) \quad \frac{\partial}{\partial z^*} = \frac{1}{2}\left(\frac{\partial}{\partial x} + j\frac{\partial}{\partial y}\right) \tag{A.8}$$

So that we get the following.

Theorem 2. *The differential of a complex valued function f is:*

$$df(z) = \frac{\partial f}{\partial z}dz + \frac{\partial f}{\partial z^*}dz^* \tag{A.9}$$

The two operators $\frac{\partial}{\partial z}, \frac{\partial}{\partial z^*}$ are also called Wirtinger derivatives. Their expressions also imply the following.

Theorem 3. *z and z^* can be regarded as constant when differentiating w.r.t. the other one, as:*

$$\frac{\partial z^*}{\partial z} = \frac{\partial z}{\partial z^*} = 0 \tag{A.10}$$

which is implied by the expression of the Wirtinger derivatives with $z = x + jy$ and $z^* = x - jy$.

For example, the function $|z|^2 = zz^*$ is not holomorphic. Indeed, it does not satisfy the Cauchy-Riemann equations as it maps to $\mathbb{R}$, thus having a null imaginary part. Then, via Wirtinger calculus:

$$\frac{\partial |z|^2}{\partial z} = \frac{\partial zz^*}{\partial z} = z^* \tag{A.11}$$

A.2.2 . Real valued functions

Since we are interested in optimizing complex values function mapping to the positive real numbers, we will face non-holomorphic functions.

Theorem 4. *for all functions $f : \mathbb{C} \to \mathbb{R}$ we have:*

$$df = 2\Re\left(\frac{\partial f}{\partial z}dz\right) = 2\Re\left(\frac{\partial f}{\partial z^*}dz^*\right) \tag{A.12}$$

We need a way to study stationary points of such functions.

Theorem 5. *for all functions $f : \mathbb{C} \to \mathbb{R}$, we have the relation:*

$$df = 0 \iff \frac{\partial f}{\partial z} = 0 \tag{A.13}$$

Which leads to the ascent direction.

Theorem 6. *for all functions $f : \mathbb{C} \to \mathbb{R}$, the steepest ascent direction is:*

$$dz = \frac{\partial f}{\partial z^*}ds \tag{A.14}$$

where ds is a real valued differential (i.e. a stepsize for optimization purposes). Indeed, from Cauchy Schwartz inequality, $\frac{\partial f}{\partial z^*}dz^*$ is maximized when dz^* and $\left(\frac{\partial f}{\partial z^*}\right)^*$ are colinear. Removing the conjugate, this means $dz = \frac{\partial f}{\partial z^*}ds$ for some real valued ds.

This suggests the gradient descent method:

$$z \leftarrow z - 2\frac{\partial f(z)}{\partial z*}ds \tag{A.15}$$

A.2.3 . Multivariate case

The extension to multivariate cases is achieved by summing the differentials overall dimensions.

Theorem 7. *The differential of a multivariate function $f : \mathbb{C}^N \to \mathbf{R}$ is:*

$$\begin{aligned} df(\mathbf{z}) &= \sum_{k=1}^{N} \frac{\partial f(\mathbf{z})}{\partial z_k} dz_k + \sum_{k=1}^{N} \frac{\partial f(\mathbf{z})}{\partial z_k^*} dz_k^* \\ &= \frac{\partial f(\mathbf{z})}{\partial \mathbf{z}}^T d\mathbf{z} + \frac{\partial f(\mathbf{z})}{\partial \mathbf{z}^*}^T d\mathbf{z}^* \end{aligned} \tag{A.16}$$

where $\frac{\partial}{\partial \mathbf{z}} = [\frac{\partial}{\partial z_1}, \ldots, \frac{\partial}{\partial z_N}]^T$ acts a gradient operator.

Similarly to the univariate case, a gradient descent method can be implemented as:

$$\mathbf{z} \leftarrow \mathbf{z} - 2\frac{\partial f(\mathbf{z})}{\partial \mathbf{z}*} ds \tag{A.17}$$

for some real-valued scalar ds acting as a step-size.

B - Finite Difference Time Domain methods

Contents

We introduce the Finite-Difference Time-Domain (FDTD) which allows us to create more realistic synthetic radar data. It is useful for testing purposes as well as creating datasets for data-driven methods, as real radar acquisitions are costly.

B.1 . General elements

B.1.1 . Introduction

The FDTD [Yee, 1966] method is one of the most popular techniques for solving electromagnetic problems today. It has been effectively applied to a wide range of issues, including scattering from metal objects and dielectrics, antennas, microstrip circuits, and electromagnetic absorption in the human body exposed to radiation. The primary reason for the success of the FDTD method is its simplicity. All electromagnetic phenomena, on a macroscopic scale, are described by the well-known Maxwell's equations, which are first order partial differential equations expressing the relations between the fundamental electromagnetic field quantities and their dependence on their sources:

$$\begin{aligned} \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \times \mathbf{B} &= \mu_0 \left(\epsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J} \right) \\ \nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon_0} \\ \nabla \cdot \mathbf{B} &= 0 \end{aligned} \tag{B.1}$$

where $\mathbf{E}, \mathbf{B}$ are the electric and magnetic fields while ϵ_0, μ_0 are free space permittivity and permeability. $\mathbf{J}$ is the current density and ρ the electric charge density.

B.1.2 . Algorithm

In order to simulate the returns from a pre-designed scene, the above equations have to be solved subject to the geometry of the problem and the initial conditions. Any material can be used as long as the permeability, permittivity, and conductivity are specified in the cells locations. The forward problem can be classified as an initial value with an open boundary problem. The initial value consists of the emission of a signal by the excitation of the transmitting antenna, while there is no boundary on which the fields take a predetermined value, which means they reach zero at infinity. This is tackled in practice by using an absorbing boundary around the scene of interest which absorbs incoming waves to truncate the computation space. The FDTD gives an approximated numerical solution to Maxwell's equations by discretization both in space and time domains, giving rise to Yee's cell (as shown in Figure B.1).

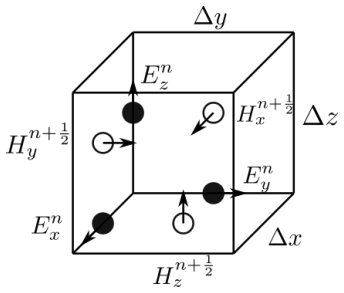


Figure B.1: Yee cell (with magnetic $\mathbf{H}$ field)

More precisely, it considers the electric and magnetic fields shifted in space by half a (time and spatial) step and uses a central difference approximation of the derivatives. They are then alternatively updated based on the other one. By assigning appropriate constitutive parameters to the locations of the electromagnetic field components, complex shaped targets can be included easily in the models.

B.1.3 . Constraints

Note that the time and space discretization steps are dependent on each other. Indeed, energy should not be able to propagate any further than one spatial step for each temporal step, because in the FDTD algorithm each node only affects its nearest neighbors. This implies the following.

Definition 14. *The Courant–Friedrichs–Lewy (CFL) condition is a necessary condition for the convergence of numerical solvers for some partial differential equations, including FDTD. For our particular problem, it states that the discretization spatial steps* $(\Delta x, \Delta y, \Delta z)$ *and the time step* Δt *must be such that:*

$$c\Delta t\sqrt{\frac{1}{(\Delta x)^2}+\frac{1}{(\Delta y)^2}+\frac{1}{(\Delta z)^2}}\leq 1 \tag{B.2}$$

To avoid errors associated with numerically induced dispersion, a rule of thumb is that the discretization step $\Delta l \in (\Delta x, \Delta y, \Delta z)$ be smaller than a tenth of the smallest wavelength of the propagating electromagnetic fields:

$$\Delta l \leq \frac{\lambda}{10} \tag{B.3}$$

For our purposes, a source can be generated by a current density term at some electric field location i.e. a Hertzian dipole with a (isotropic) Gaussian waveform.

B.2 . Complex media

B.2.1 . Dispersion

When we subject a dielectric material to an electric field, it reacts by polarizing. This reaction is not instantaneous and depends on the frequency of the applied field. Recall that the permittivity is modeled as a complex variable for lossy media (non-zero conductivity):

$$\epsilon_c = \epsilon' - j\epsilon'' \tag{B.4}$$

where $\epsilon'' = \frac{\sigma}{\omega}$ with σ, ω the material conductivity and the applied field angular frequency. The imaginary part is non-zero in lossy media, where dissipation of the propagating wave energy into current and then heat occurs. The factor $\frac{\epsilon''}{\epsilon'} = \tan\delta$ is also called the loss tangent. This can then be rewritten as:

$$\epsilon_c = \epsilon'(1 - j\tan\delta) \tag{B.5}$$

Dielectric relaxation is the momentary lag in the permittivity of a material caused by the delay in molecular polarisation with respect to a changing electric field. Debye relaxation is the dielectric relaxation response of an ideal, non-interacting population of dipoles to an alternating external electric field. It can be written as:

$$\epsilon_c(\omega) = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{1 + j\omega\tau} + \frac{\sigma_s}{j\omega\epsilon_0} \tag{B.6}$$

where ϵ_∞ is the complex permittivity limit in high frequencies, ϵ_s the static one (for low frequencies), σ_s the static conductivity while τ is the characteristic relaxation time of the medium. For common wall materials, it may not be described accurately with the Debye model [Zhekov et al., 2020]. A multi-pole extension may be used:

$$\epsilon_c(\omega) = \epsilon_\infty + \sum_{n=1}^{N} \frac{\Delta\epsilon_{d,n}}{1 + j\omega\tau_{d,n}} + \frac{\sigma_s}{j\omega\epsilon_0} \tag{B.7}$$

where $\epsilon_{d,n}$ and $\tau_{d,n}$ constitute Debye models parameters to be fitted for N poles.

B.2.2 . Other effects

Another effect we can model but that we will not delve into is anisotropy, by modeling the constitutive parameters of the materials as second rank tensors, for example as a diagonal matrix over the spatial dimension:

$$\epsilon = \begin{bmatrix} \epsilon_{xx} & 0 & 0 \\ 0 & \epsilon_{yy} & 0 \\ 0 & 0 & \epsilon_{zz} \end{bmatrix} \tag{B.8}$$

Other elements include surface roughness or more realistic antennas.

Bibliography

[Absil et al., 2012] Absil, P., Amodei, L., and Meyer, G. (2012). Two newton methods on the manifold of fixed-rank matrices endowed with riemannian quotient geometries. *Computational Statistics*, 29.

[Absil et al., 2008] Absil, P.-A., Mahony, R., and Sepulchre, R. (2008). *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ.

[Ahmad, 2008] Ahmad, F. (2008). Multi-location wideband through-the-wall beamforming. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5193–5196.

[Ahmad et al., 2007] Ahmad, F., Amin, M. G., and Mandapati, G. (2007). Autofocusing of through-the-wall radar imagery under unknown wall characteristics. *IEEE Transactions on Image Processing*, 16(7):1785–1795.

[Amin, 2017] Amin, M. (2017). *Through-the-Wall Radar Imaging*. CRC Press.

[Amin and Ahmad, 2013] Amin, M. G. and Ahmad, F. (2013). Compressive sensing for through-the-wall radar imaging. *Journal of Electronic Imaging*, 22(3):1 – 22.

[Beck, 2017] Beck, A. (2017). *First-Order Methods in Optimization*. SIAM-Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

[Beck and Teboulle, 2009] Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.

[Bilogur, 2021] Bilogur, A. (2021). LR schedulers, adaptive optimizers. https://residentmario.github.io/pytorch-training-performance-guide/lr-sched-and-optim.html. Accessed: 23/08/2024.

[Boumal, 2023] Boumal, N. (2023). *An introduction to optimization on smooth manifolds*. Cambridge University Press.

[Boyd et al., 2011] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122.

[Brehier et al., 2023a] Brehier, H., Breloy, A., Ren, C., and Ginolhac, G. (2023a). Atténuation robuste du fouillis mural en imagerie radar à travers murs par optimisation riemannienne. In *XXIXème Colloque Francophone de Traitement du Signal et des Images, GRETSI*.

[Brehier et al., 2023b] Brehier, H., Breloy, A., Ren, C., and Ginolhac, G. (2023b). Through the wall radar imaging via Kronecker-structured Huber-type RPCA. *Signal Processing*, page 109228.

[Brehier et al., 2024a] Brehier, H., Breloy, A., Ren, C., and Ginolhac, G. (2024a). Deep unrolling of Robust PCA and convolutional sparse coding for stationary target localization in through wall radar imaging. In *2024 32th European Signal Processing Conference (EUSIPCO)*.

[Brehier et al., 2024b] Brehier, H., Breloy, A., Ren, C., and Ginolhac, G. (2024b). Through-the-wall radar imaging with wall clutter removal via riemannian optimization on the fixed-rank manifold. In *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8596–8600.

[Brehier et al., 2022a] Brehier, H., Breloy, A., Ren, C., Hinostroza, I., and Ginolhac, G. (2022a). Robust PCA for through-the-wall radar imaging. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 2246–2250.

[Brehier et al., 2022b] Brehier, H., Breloy, A., Ren, C., Hinostroza, I., and Ginolhac, G. (2022b). Robust PCA pour l'imagerie radar à travers les murs. In *XXVIIème Colloque Francophone de Traitement du Signal et des Images, GRETSI*.

[Breloy et al., 2018] Breloy, A., El Korso, M. N., Panahi, A., and Krim, H. (2018). Robust subspace clustering for radar detection. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1602–1606.

[Candès et al., 2011] Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37.

[Chambolle and Pock, 2011] Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145.

[Chandrasekaran et al., 2011] Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. (2011). Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596.

[Chen et al., 2016] Chen, C., He, B., Ye, Y., and Yuan, X. (2016). The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Math. Program.*, 155(1–2):57–79.

[Clemente et al., 2013] Clemente, C., Balleri, A., Woodbridge, K., and Soraghan, J. J. (2013). Developments in target micro-doppler signatures analysis: radar imaging, ultrasound and through-the-wall radar. *EURASIP Journal on Advances in Signal Processing*, 2013(1):1–18.

[Courty et al., 2017a] Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. (2017a). Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Courty et al., 2017b] Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2017b). Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865.

[de Leeuw and Lange, 2009] de Leeuw, J. and Lange, K. (2009). Sharp quadratic majorization in one dimension. *Computational Statistics and Data Analysis*, 53(7):2471–2484.

[Debes et al., 2011] Debes, C., Hahn, J., Zoubir, A. M., and Amin, M. G. (2011). Target discrimination and classification in through-the-wall radar imaging. *IEEE Transactions on Signal Processing*, 59(10):4664–4676.

[Dehmollaian and Sarabandi, 2008] Dehmollaian, M. and Sarabandi, K. (2008). Refocusing through building walls using synthetic aperture radar. *IEEE Transactions on Geoscience and Remote Sensing*, 46(6):1589–1599.

[Donoho, 2006] Donoho, D. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306.

[Durand, 2007] Durand, R. (2007). *Processeurs SAR basés sur des détecteurs de sous-espaces*. PhD thesis. Thèse de doctorat dirigée par Forster, Philippe Traitement du signal Paris 10 2007.

[Fang et al., 2015] Fang, E., He, B.-S., Liu, H., and Yuan, X. (2015). Generalized alternating direction method of multipliers: New theoretical insights and applications. *Mathematical Programming Computation*, 7.

[Fazel, 2002] Fazel, M. (2002). *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University.

[Gallet et al., 2023] Gallet, M., Mian, A., Ginolhac, G., Ollila, E., and Stelzenmuller, N. (2023). New robust sparse convolutional coding inversion algorithm for ground penetrating radar images. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–14.

[Ganin et al., 2016] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030.

[Garnier and Papanicolaou, 2016] Garnier, J. and Papanicolaou, G. (2016). *Passive Imaging with Ambient Noise*. Cambridge University Press.

[Gennarelli et al., 2015] Gennarelli, G., Vivone, G., Braca, P., Soldovieri, F., and Amin, M. G. (2015). Multiple extended target tracking for through-wall radars. *IEEE Transactions on Geoscience and Remote Sensing*, 53(12):6482–6494.

[Gregor and LeCun, 2010] Gregor, K. and LeCun, Y. (2010). Learning fast approximations of sparse coding. In *International Conference on Machine Learning*.

[Gretton et al., 2006] Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. (2006). A kernel method for the two-sample-problem. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

[Han, 2022] Han, D.-R. (2022). A survey on some recent developments of alternating direction method of multipliers. *Journal of the Operations Research Society of China*, pages 1–52.

[Hestenes, 1969] Hestenes, M. R. (1969). Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320.

[Hoffman et al., 2018] Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. (2018). CyCADA: Cycle-consistent adversarial domain adaptation. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1989–1998. PMLR.

[Huang et al., 2010] Huang, Q., Qu, L., Wu, B., and Fang, G. (2010). Uwb through-wall imaging based on compressive sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 48(3):1408–1415.

[Huber, 1964] Huber, P. J. (1964). Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101.

[Hunger, 2007] Hunger, R. (2007). An introduction to complex differentials and complex differentiability. Technical Report TUM-LNS-TR-07-06.

[Jin et al., 2013] Jin, T., Chen, B., and Zhou, Z. (2013). Image-domain estimation of wall parameters for autofocusing of through-the-wall sar imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 51(3):1836–1843.

[Kamilov et al., 2023] Kamilov, U. S., Bouman, C. A., Buzzard, G. T., and Wohlberg, B. (2023). Plug-and-play methods for integrating physical and learned models in computational imaging: Theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 40(1):85–97.

[Kebe et al., 2020] Kebe, M., Gadhafi, R., Mohammad, B., Sanduleanu, M., Saleh, H., and Al-Qutayri, M. (2020). Human vital signs detection methods and potential using radars: A review. *Sensors*, 20:1454.

[Kingma and Ba, 2015] Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

[Kowalski, 2009] Kowalski, M. (2009). Sparse regression using mixed norms. *Applied and Computational Harmonic Analysis*, 27(3):303–324.

[Leigsnering et al., 2014] Leigsnering, M., Ahmad, F., Amin, M., and Zoubir, A. (2014). Multipath exploitation in through-the-wall radar imaging using sparse reconstruction. *IEEE Transactions on Aerospace and Electronic Systems*, 50(2):920–939.

[Li et al., 2021a] Li, H., Cui, G., Guo, S., Kong, L., and Yang, X. (2021a). Human target detection based on FCN for through-the-wall radar imaging. *IEEE Geoscience and Remote Sensing Letters*, 18(9):1565–1569.

[Li et al., 2019] Li, H., Cui, G., Kong, L., Chen, G., Wang, M., and Guo, S. (2019). Robust human targets tracking for mimo through-wall radar via multi-algorithm fusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(4):1154–1164.

[Li et al., 2021b] Li, Z., Jin, T., Dai, Y., and Song, Y. (2021b). Through-wall multi-subject localization and vital signs monitoring using uwb mimo imaging radar. *Remote Sensing*, 13(15).

[Lions and Mercier, 1979] Lions, P. L. and Mercier, B. (1979). Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979.

[Long et al., 2015] Long, M., Cao, Y., Wang, J., and Jordan, M. (2015). Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 97–105, Lille, France. PMLR.

[Loshchilov and Hutter, 2017] Loshchilov, I. and Hutter, F. (2017). SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

[Loshchilov and Hutter, 2019] Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

[Mardani et al., 2013] Mardani, M., Mateos, G., and Giannakis, G. B. (2013). Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies. *IEEE Transactions on Information Theory*, 59(8):5186–5205.

[Maronna et al., 2019] Maronna, R., Martin, R., Yohai, V., and Salibián-Barrera, M. (2019). *Robust Statistics: Theory and Methods (with R)*. Wiley Series in Probability and Statistics. Wiley.

[Min et al., 2023] Min, W., Xu, T., Wan, X., and Chang, T.-H. (2023). Structured sparse non-negative matrix factorization with $\ell_{2,0}$-norm. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8584–8595.

[Mishra et al., 2012] Mishra, B., Meyer, G., Bonnabel, S., and Sepulchre, R. (2012). Fixed-rank matrix factorizations and riemannian low-rank optimization. *Computational Statistics*, 29.

[Monga et al., 2021] Monga, V., Li, Y., and Eldar, Y. C. (2021). Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44.

[Mériaux et al., 2019] Mériaux, B., Breloy, A., Ren, C., El Korso, M. N., and Forster, P. (2019). Modified sparse subspace clustering for radar detection in non-stationary clutter. In *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 669–673.

[Nogueira, 2014] Nogueira, F. (2014). Bayesian Optimization: Open source constrained global optimization tool for Python.

[Ollila et al., 2012] Ollila, E., Tyler, D. E., Koivunen, V., and Poor, H. V. (2012). Complex elliptically symmetric distributions: Survey, new results and applications. *IEEE Transactions on Signal Processing*, 60(11):5597–5625.

[Pan et al., 2011] Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210.

[Papyan et al., 2016] Papyan, V., Romano, Y., and Elad, M. (2016). Convolutional neural networks analyzed via convolutional sparse coding. *Journal of Machine Learning Research*, 18.

[Parikh and Boyd, 2014] Parikh, N. and Boyd, S. (2014). Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239.

[Pati et al., 1993] Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *in Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pages 1–3.

[Protiva et al., 2011] Protiva, P., Mrkvica, J., and Machac, J. (2011). Estimation of wall parameters from time-delay-only through-wall radar measurements. *IEEE Transactions on Antennas and Propagation*, 59(11):4268–4278.

[Qu et al., 2022] Qu, L., Wang, C., Yang, T., Zhang, L., and Sun, Y. (2022). Enhanced through-the-wall radar imaging based on deep layer aggregation. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer.

[Smith and Topin, 2018] Smith, L. N. and Topin, N. (2018). Super-convergence: Very fast training of residual networks using large learning rates.

[Snoek et al., 2012] Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[Solomon et al., 2020] Solomon, O., Cohen, R., Zhang, Y., Yang, Y., He, Q., Luo, J., van Sloun, R. J. G., and Eldar, Y. C. (2020). Deep unfolded robust PCA with application to clutter suppression in ultrasound. *IEEE Transactions on Medical Imaging*, 39(4):1051–1063.

[Soumekh, 1999] Soumekh, M. (1999). Synthetic aperture radar signal processing with matlab algorithms.

[Sreter and Giryes, 2018] Sreter, H. and Giryes, R. (2018). Learned convolutional sparse coding. In *2018 ICASSP*, pages 2191–2195.

[Sun et al., 2016] Sun, Y., Babu, P., and Palomar, D. (2016). Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Trans. on Signal Process.*, PP(99):1–1.

[Tang et al., 2020] Tang, V. H., Bouzerdoum, A., and Phung, S. L. (2020). Compressive radar imaging of stationary indoor targets with low-rank plus jointly sparse and total variation regularizations. *IEEE Transactions on Image Processing*, 29:4598–4613.

[Tang et al., 2016] Tang, V. H., Bouzerdoum, A., Phung, S. L., and Tivive, F. H. C. (2016). Radar imaging of stationary indoor targets using joint low-rank and sparsity constraints. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1412–1416.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

[Tivive et al., 2011] Tivive, F. H. C., Amin, M. G., and Bouzerdoum, A. (2011). Wall clutter mitigation based on eigen-analysis in through-the-wall radar imaging. In *2011 17th International Conference on Digital Signal Processing (DSP)*, pages 1–8.

[Tivive et al., 2015] Tivive, F. H. C., Bouzerdoum, A., and Amin, M. G. (2015). A subspace projection approach for wall clutter mitigation in through-the-wall radar imaging. *IEEE Transactions on Geoscience and Remote Sensing*, 53(4):2108–2122.

[Townsend et al., 2016] Townsend, J., Koep, N., and Weichwald, S. (2016). Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research*, 17(137):1–5.

[Uzawa, 1958] Uzawa, H. (1958). Iterative methods for concave programming. *Studies in linear and nonlinear programming*, 6:154–165.

[Vandereycken, 2013] Vandereycken, B. (2013). Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Verma et al., 2009] Verma, P. K., Gaikwad, A. N., Singh, D., and Nigam, M. (2009). Analysis of clutter reduction techniques for through wall imaging in uwb range. *Progress In Electromagnetics Research B*, 17:29–48.

[Wang and Amin, 2006] Wang, G. and Amin, M. (2006). Imaging through unknown walls using different standoff distances. *IEEE Transactions on Signal Processing*, 54(10):4015–4025.

[Wang et al., 2003] Wang, Z., Simoncelli, E., and Bovik, A. (2003). Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems and Computers, 2003*, volume 2, pages 1398–1402 Vol.2.

[Warren et al., 2016] Warren, C., Giannopoulos, A., and Giannakis, I. (2016). gprMax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar. *Computer Physics Communications*, 209:163–170.

[Wax and Kailath, 1985] Wax, M. and Kailath, T. (1985). Detection of signals by information theoretic criteria. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(2):387–392.

[Wohlberg, 2016] Wohlberg, B. (2016). Efficient algorithms for convolutional sparse representations. *IEEE Transactions on Image Processing*, 25(1):301–315.

[Yang et al., 2021] Yang, D., Zhu, Z., Zhang, J., and Liang, B. (2021). The overview of human localization and vital sign signal measurement using handheld ir-uwb through-wall radar. *Sensors*, 21(2).

[Yang and Soatto, 2020] Yang, Y. and Soatto, S. (2020). Fda: Fourier domain adaptation for semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4084–4094, Los Alamitos, CA, USA. IEEE Computer Society.

[Yee, 1966] Yee, K. (1966). Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, 14(3):302–307.

[Zhang et al., 2023] Zhang, X., Zheng, J., Wang, D., Tang, G., Zhou, Z., and Lin, Z. (2023). Structured sparsity optimization with non-convex surrogates of $\ell_{2,0}$-norm: A unified algorithmic framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6386–6402.

[Zhekov et al., 2020] Zhekov, S. S., Franek, O., and Pedersen, G. F. (2020). Dielectric properties of common building materials for ultrawideband propagation studies. *IEEE Antennas and Propagation Magazine*, 62(1):72–81.

[Zhu et al., 2017] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251.