Università degli Studi di Padova

Department of Information Engineering
University of Padova

University of Padova

Department of Information Engineering

Master Thesis in

ICT for Internet and Multimedia

Deep Learning Techniques for Backscattering Vector Estimation in ToF Data

Supervisor
Pietro Zanuttigh
University of Padova

Co-supervisor
Henrik Schäfer
Sony Europe B.V.

Co-supervisor
Gianluca Agresti
University of Padova

Master Candidate
Enrico Buratto
University of Padova

Padova, 2 December, 2019
Academic Year 2018/2019

I dedicate this work to my family
who always supported me throughout my studies.

Abstract

In this work we propose a new approach to correct the *multi-path interference* phenomenon which occurs in *time-of-flight* cameras. The basic ToF assumption that the light is only reflected once in the scene is not met in reality. In the real-world, multiple light rays come at the camera sensor from multiple indirect paths with different amplitudes and time delays. This causes interference and affects the final depth estimation accuracy.

We introduce a deep learning approach to learn the typical reflection structure of the light in a real environment and use it as strong prior to estimate the shape of the time-dependent scene impulse response, called *backscattering vector*. We use a predictive model to perform the actual prediction starting from the raw ToF data acquired by the camera and a generative model to constrain the solution to resemble the characteristic structure of real backscattering vectors. We show also how spatial correlation on the input data can be efficiently exploited to improve the final prediction.

We develop the proposed approach under some simplifying assumptions. In particular, we assume that the MPI effect is generated only by specular reflections. Nevertheless, experimental results on real data demonstrate the effectiveness of the proposed approach showing performance comparable with other state-of-the-art algorithms.

Acknowledgments

Before starting, I would like to thank some people that helped me during my work and without whom I would not have been able to finish it.

I would like to express my sincere gratitude to Professor *Pietro Zanuttigh* for his helpfulness. First, he was the teacher of my Computer Vision and Machine Learning classes who inspired my interest in this field, then he was my main supervisor for this work who advised me throughout the entire duration of the thesis.

This research has been carried on during my six-months internship at the Sony European Technology Center (EuTEC) in Stuttgart. I would like to thank *Sony* for the supplied resources and all the members of the *Computational Imaging Group*. In particular, I would like to gratefully acknowledge my mentor *Henrik Schäfer* for his precious comments and insights.

I am particularly grateful for the assistance given by my co-supervisor *Gianluca Agresti* who supported me during my work with his experience and with valuable advice.

Finally, I must express my very profound gratitude to my *family* for the continuous support and encouragement throughout my years of study. This accomplishment would not have been possible without them.

Table of contents

List of acronyms

AMCW Amplitude Modulated Continuous Wave. vii, 4, 5, 11, 15, 27

CMD Cumulative Mass Distribution. 58

DoF Degree of Freedom. 27, 28, 64

EMD Earth Mover's Distance. viii, 58, 59, 61, 63, 64, 66, 72

LTI Linear Time-Invariant. 5, 9

MAE Mean Absolute Error. 60, 70–74

MPI Multi-Path Interference. iii, vii, 1, 3, 9, 11, 15–18, 21–24, 27, 32, 35–37, 39, 40, 43, 44, 47, 60, 61, 63, 69–75, 77, 78, 81, 83

PMD Probability Mass Distribution. 53, 57, 58

ToF Time-of-Flight. iii, vii, 1–6, 8, 10, 12, 13, 15, 17–19, 21–24, 27, 28, 30–33, 35, 36, 38–40, 44, 48, 49, 60, 65, 69–71, 77, 78, 80, 81, 83

The truly creative people tend to be outliers.

Nolan Bushnell

1 Introduction

A ToF camera is a range imaging system which is able to capture depth information in real-time. It works by illuminating the scene with a light signal and measuring the time taken by the light to travel the distance from the camera to the object and back again. However, the basic assumption that the light is only reflected once in the scene is not met in many real cases. Typically, it bounces multiple times inside the scene, generating multiple reflected signals that interference at the camera sensor. This is known as *multi-path interference effect* and it can lead to significant errors in the depth estimation [1].

The time-dependent scene impulse response, indicating the amount of light returned to the camera sensor after a particular time-of-flight, is called *backscattering vector*. It contains important information about the scene geometry, therefore its knowledge may be exploited in a wide range of applications. It can be used to see around the corners [28], detect object in highly-scattering media [29], infer material properties from a distance [30, 31], as well as perform optimal multi-path interference correction.

Many authors [2] in the literature propose methods to correct the multi-path effect starting from the raw data acquired by the ToF camera. This work focuses on the development of a completely new approach for backscattering estimation and its application to the multi-path interference correction task. Due to the reflection properties of the light in a real environment, typically the backscattering vector presents some very characteristic structures. The main idea is to employ a deep learning approach to learn the underlying reflection structure and use it as strong prior to optimize the estimation of the backscattering vector.

The proposed technique is based on two models, the first one is a *predictive* model which takes in input the raw ToF data and produces a compressed representation of the corresponding backscattering vector, while the second is a *generative* model that reconstructs the final backscattering vector starting from its compressed representation. We developed the proposed approach under some simplifying assumptions. In particular, we trained the deep leaning architecture in a supervised manner on simulated data, assuming that MPI is generated only by two specular reflections of the light inside the scene. In future works the idea can be generalized to a more realistic scenario, accounting for diffuse reflections and performing the optimization on real data through some unsupervised domain adaptation techniques.

Experimental results confirm the effectiveness of the approach, achieving performance comparable to other state-of-the-art algorithms for multi-path correction. We worked either at single pixel-level and at local level, showing how spatial correlation can be efficiently exploited to improve the final prediction. The method turns out to be robust against noise, and provides good correction capabilities on real-world ToF data.

Technology is nothing. What's important is that you have a faith in people, that they're basically good and smart, and if you give them tools, they'll do wonderful things with them.

Steve Jobs

2
ToF Technology and MPI Problem

The use of Time-of-Flight (ToF) technology for range imaging has become increasingly popular in the last years. Compared to other technologies to obtain scene depth, ToF cameras are more versatile and this makes them useful in a wide range of applications, such as pose estimation, coarse 3D reconstruction, human body parts recognition and tracking, augmented reality, scene understanding, autonomous driving, robotic navigation systems, and so on. Depth measurements are based on the well-known time-of-flight principle. It works by illuminating the scene with a light signal and estimating the depth by measuring the time Δt taken by the light to come back to the camera sensor. Since speed of light in vacuum is constant, *i.e.* $299\,792\,458\ \mathrm{m/s^2}$, the relationship between depth and time is given by:

$$d = \frac{ct}{2} \tag{2.1}$$

The factor $1/2$ is due to the fact that the light has to travel the distance twice, it propagates from the ToF projector to the scene and comes back. Note that ToF cameras acquires radial depth maps, this means that the distance value associated to each pixel refers to the radial distance of the corresponding 3D point from the camera projector. Radial depth can be converted to the standard z-depth knowing the intrinsic parameters of the camera.

The success of these systems is driven by their benefits, in particular they exhibit some interesting advantages compared to their competitors. Nowaday, the most widely used range imaging techniques are the stereo and the structured light cameras. In stereo, the depth is computed looking for pixel-wise correspondences into two images of the same scene. The main limitation in this context is that the success of the correspondence computation requires the presence of meaningful textures in the scene. Structured light cameras mitigate this problem actively projecting a light pattern to be used as reference. Both these two technologies suffer the occlusion problem, require an accurate extrinsic calibration and the correspondence computation prevents them from being used in real-time. Conversely, ToF systems are able to acquire dense depth and intensity images simultaneously at high frame rates,

	ToF	Stereo	Structured light
Correspondence computation	No	Yes	Yes
Occlusion problem	Limited	Yes	Yes
Extrinsic calibration	No	Yes	Yes
Auto illumination	Yes	No	Yes
Untextured surfaces	Good performance	Bad performance	Good performance
Depth range	Light-power and mod. freq. dependent	Baseline dependent	Light-power and baseline dependent
Spatial resolution	Low	High	High

Table 2.1: Comparison between range imaging techniques

have low weight and compact design without any moving parts and do not suffer from the occlusion problem. In table 2.1 a comparison between the most widely used range imaging techniques is reported.

There exist two main types of ToF technologies. Sensors based on discrete pulse modulation that measure the time-of-flight directly, or sensors based on Amplitude Modulated Continuous Wave (AMCW) that measure the time-of-flight indirectly. The AMCW sensors estimate the time from the phase displacement between the emitted and the received light signals. This work will focus only on AMCW systems, since currently they are the most available and widely used cameras on the market. Beside all these good aspects, ToF cameras are subjected to some limitations that need to be further analysed and improved. Between all, the multi-path interference problem is responsible for a significant error in the depth estimation that is highly scene dependent and thus difficult to be corrected.

2.1 AMCW Camera Principle

AMCW cameras illuminate the scene with an amplitude modulated continuous near-infrared light signal at frequency f_m and compute the phase displacement between the reflected signal and an internal reference clock for each pixel. Figure 2.1 shows a graphical illustration of the AMCW camera principle. Typically the frequency of the amplitude modulation is in the megahertz range. Knowing the phase shift $\varphi \in [0, 2\pi)$, it is possible to derive the corresponding time-of-flight and depth values according to:

$$\Delta t = \frac{\varphi}{2\pi f_m} \qquad d = \frac{c\,\Delta t}{2} = \frac{c\,\varphi}{4\pi f_m} \tag{2.2}$$

Note that the previous equations link the modulation frequency at the depth resolution, higher is the modulation frequency smaller is the minimum appreciable depth value. As we will see later in section 2.1.2, modulation frequency influences also the maximum measurable range. The higher the frequency the smaller the maximum acquirable range.

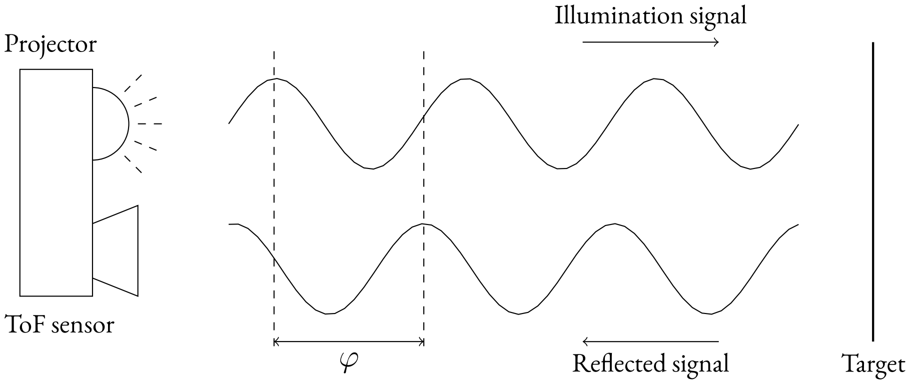


Figure 2.1: Illustration of AMCW camera principle. The ToF camera illuminates the scene with an amplitude modulated light signal and estimates the depth looking at the phase displacement between the illumination and the reflected signals.

The most common setting is to use a sinusoidal illumination signal and a rectangular clock at the same frequency, this combination leads to a closed-form solution for phase and amplitude reconstruction [3, 4]. Mathematically, the illumination signal is expressed as:

$$i_m(t) = 1 + sin\left(2\pi f_m t\right) = 1 + sin\left(\omega t\right) \tag{2.3}$$

For a particular acquisition time t, the scene can be modelled as a Linear Time-Invariant (LTI) system. Assuming that the light bounces only once in the scene, it introduces only an attenuation $\alpha < 1$ and a time delay $\Delta t > 0$ in the illumination signal, therefore its impulse response is given by:

$$h(t) = \alpha\, \delta\left(t - \Delta t\right) \tag{2.4}$$

Note that the delay $\Delta t = \frac{2d}{c}$ corresponds to the time-of-flight of the light. Under this ideal assumption, the reflected illumination signal presents only an attenuated $\alpha < 1$ and a phase displacement $\varphi = 2\pi f_m \Delta t$:

$$r(t) = \left(i_m * h\right)(t) = \alpha + \alpha\, sin\left(2\pi f_m (t - \Delta t)\right) = \alpha + \alpha\, sin\left(\omega t - \varphi\right) \tag{2.5}$$

The sensor sensitivity is modulated with a rectangular shutter signal at the same frequency f_m, represented by:

$$s(t) = \mathbb{1}\left(sin\left(2\pi f_m t\right)\right) = \mathbb{1}\left(sin\left(\omega t\right)\right) \tag{2.6}$$

The camera sensor is formed by an array of smart pixels, the so-called *lock-in pixels* [5]. Each pixel is able to sample the amount of light reflected by the scene, computing the correlation between the reflected illumination signal and the sensor sensitivity inside a given integration interval T_{int}, as illustrated in figure 2.2. The raw correlation measure recorded by the camera

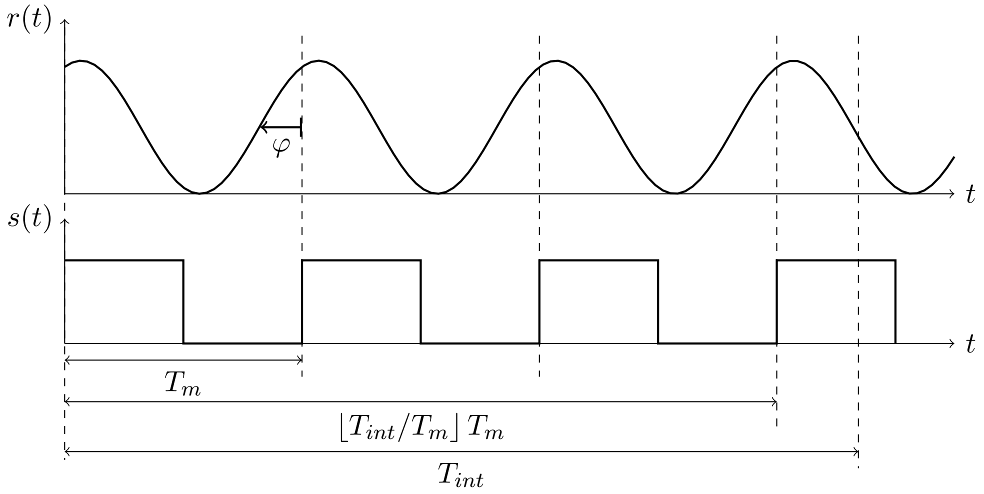


Figure 2.2: Correlation computation. The lock-in pixels of the ToF camera sensor record the correlation between the reflected light signal and the sensor sensitivity inside a given integration interval T_{int}.

is given by:

$$\begin{aligned} m &= \int_{T_{int}} r(t)\, s(t)\, dt \\ &= \underbrace{\left\lfloor \frac{T_{int}}{T_m} \right\rfloor}_{N} \int_0^{T_m} r(t)\, s(t)\, dt + \int_{NT_m}^{T_{int}} r(t)\, s(t)\, dt \end{aligned} \tag{2.7}$$

Under the reasonable assumption that $T_m = \frac{1}{f_m} \ll T_{int}$ the second term in (2.7) can be neglected. Moreover, since the sensitivity is zero for the second half of the modulation period, the equation reduces to:

$$\begin{aligned} m &= N \int_0^{T_m/2} \left[\alpha + \alpha\, sin\,(\omega t - \varphi)\right]\, dt \\ &= N\,\alpha \frac{T_m}{2} + N\,\alpha \int_0^{T_m/2} sin\,(\omega t - \varphi)\;dt \\ &= \underbrace{N\,\alpha \frac{T_m}{2}}_{I} + \underbrace{N\,\alpha \frac{T_m}{\pi}}_{A} \Big[- cos\,(\omega t - \varphi) \Big]_0^{T_m/2} \\ &= I + A\, cos\,(\varphi) \end{aligned} \tag{2.8}$$

The main result is that the raw correlation measurement captured by the ToF sensor turns out to be a sinusoidal function of the phase shift between illumination and reflected signal.

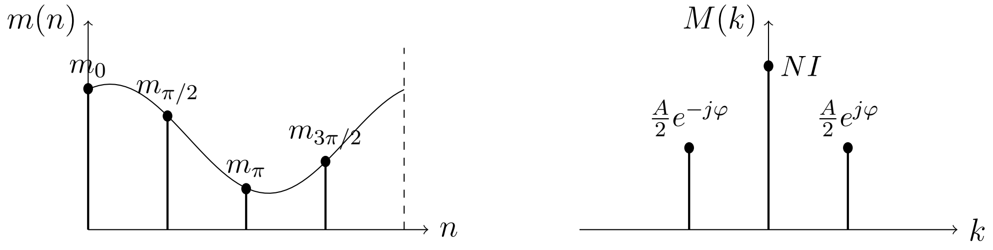


Figure 2.3: Correlation function reconstruction. In order to reconstruct completely amplitude, phase and intensity of the correlation function it has to be sampled several times. On the left it is reported the correlation function sampled for $\theta = 0, \pi/2, \pi$ and $3\pi/2$, while on the right the corresponding DFT. In case of perfect sinusoidal modulation the DFT contains only the fundamental harmonic and a DC component.

2.1.1 Correlation Function Reconstruction

The correlation function encodes useful information about the scene impulse response in its amplitude, phase and intensity, therefore we are interested in its reconstruction. To this end, the correlation function has to be sampled several times, figure 2.3. One simple way is introducing additional known internal phase displacements $\theta = n2\pi/K$, for $n = 0, ..., K-1$, in the sensor sensitivity and performing K subsequent correlation measurements. Since there are three unknowns, it must be $K \geq 3$. More measurements improve the precision but also incorporate additional errors due to the sequential sampling such as motion blur which will be discussed later on. Typically the system is solved in the optimal least-squares sense [1] using $K = 4$ samples. The recorded correlation measures are going to be dependent on the artificial phase displacements introduced in the sensor sensitivity:

$$m_{2\pi n/K} = I + A\,cos\left(\varphi + n\frac{2\pi}{K}\right) = I + \frac{A}{2}\left(e^{j\left(\varphi+n\frac{2\pi}{K}\right)} + e^{-j\left(\varphi+n\frac{2\pi}{K}\right)}\right) \tag{2.9}$$

Amplitude, phase and intensity can be easily derived looking at the DFT of the previous expression:

$$M(k) = \sum_{n=0}^{K-1} m_{2\pi n/K}\, e^{-jkn\frac{2\pi}{K}} \tag{2.10}$$

Due to linearity of equation (2.9), it results:

$$M(k) = \frac{A}{2}e^{-j\varphi}\,\delta\,(k+1) + NI\,\delta\,(k) + \frac{A}{2}e^{j\varphi}\,\delta\,(k-1) \tag{2.11}$$

Imposing the equality between equations (2.10) and (2.11), the information about intensity is contained on the DC tap $M(0)$, while the information about phase and amplitude can be extracted from the first tap $M(1)$. For $K = 4$ it results:

$$M(0) = 4I = m_0 + m_{\pi/2} + m_\pi + m_{3\pi/2} \tag{2.12}$$

$$\begin{aligned} M(1) = \frac{A}{2}e^{j\varphi} &= m_0\, e^0 + m_{\pi/2}\, e^{-\pi/2} + m_\pi\, e^{-\pi} + m_{3\pi/2}\, e^{-3\pi/2} \\ &= (m_0 - m_\pi) + j(m_{3\pi/2} - m_{\pi/2}) \end{aligned} \tag{2.13}$$

The inversion of the previous relationships leads to a close-form solution for the amplitude:

$$A = \frac{1}{2}\sqrt{(m_0 - m_\pi)^2 + (m_{3\pi/2} - m_{\pi/2})^2} \tag{2.14}$$

the phase:

$$\varphi = arctan\left(\frac{m_{3\pi/2} - m_{\pi/2}}{m_0 - m_\pi}\right) \tag{2.15}$$

and the intensity:

$$I = \frac{m_0 + m_{\pi/2} + m_\pi + m_{3\pi/2}}{4} \tag{2.16}$$

Note that, in order to reconstruct completely the correlation function, an interval of time $K\, T_{int}$ is required, since K subsequent correlation measures need to be acquired. The stretching of the overall acquisition time may lead to motion blurring effects in case of dynamic scenes.

2.1.2 Unambiguous Range

At this point it is important to notice that, since the correlation function (2.9) is periodic of period 2π, the reconstructed phase value φ will be necessary in the interval $[0, 2\pi)$. This implies that also the depth estimated through equation (2.2) will always falls into a specific interval, that interval is called *unambiguous range*:

$$d \in [0, D_a) \qquad D_a = \frac{c}{2f_m} = \frac{\lambda_{mod}}{2} \tag{2.17}$$

where λ_{mod} is the wavelength of the illumination signal. Distances to objects that differ of D_a appear undistinguishable. The maximum unambiguous distance D_a represents the maximum depth value that can be correctly estimated acquiring ToF data at a single modulation frequency f_m. Real depth values greater than the maximum unambiguous distance are going to wrap into the interval $[0, D_a)$, leading to an ambiguity in the depth estimation. Clearly, as the modulation frequency f_m increases, the depth resolution increases, but the unambiguous range decreases. This trade-off must be taken into consideration for the choice of the modulation frequency. As we will see later in section 2.6, acquiring data at multiple modulation frequencies allows to extend the unambiguous range.

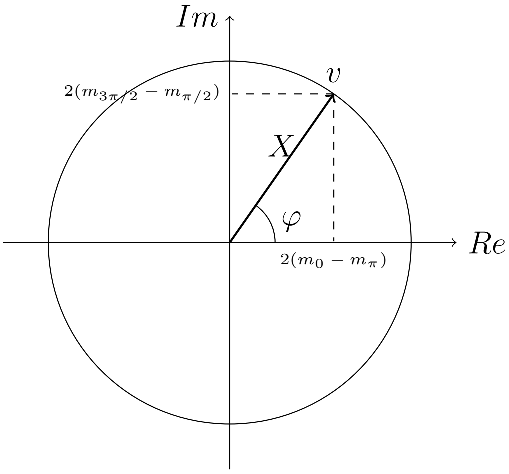


Figure 2.4: Phasor representation of the correlation function in a MPI-free case. Amplitude and phase of the recorded correlation function can be fully characterized by means of a complex phasor.

2.1.3 Phasor Notation

As stressed by Gupta *et al.* [6], a convenient representation for the sinusoidal correlation function is the phasor notation. Ignoring the intensity component that will require a separate treatment, the information about amplitude and phase of the recorder correlation function is fully contained on the first tap of its DFT (2.10). Without loss of generality, it can be fully characterize by means of the complex phasor, as illustred in figure 2.4:

$$v = 2M(1) = X\, e^{j\varphi} \in \mathbb{C} \qquad\qquad \varphi = 2\pi f_m \Delta t \tag{2.18}$$

This duality allows to perform operations between sinusoidal signals in the complex plane, simplifying the analysis of multiple interfering reflected signals in a MPI scenario.

2.2 Measurement Model and Backscattering

We have seen that, a light signal that flies inside the scene for a time Δt before coming back to the camera sensor produces a correlation measure fully characterized by the complex phasor:

$$v = X\, e^{j2\pi f_m \Delta t} \tag{2.19}$$

Due to the linearity of the scene modelled as a LTI system, in case of a more complex behaviour of the light, it is always possible to express the acquired correlation measure as an integral over all the infinitesimal light returns [7], that is:

$$v = \int_{t_{min}}^{t_{max}} x(t)\, e^{j2\pi f_m t}\, dt \tag{2.20}$$

where $[t_{min}, t_{max}]$ is the ToF interval of interest and $x(t)$ is referred to as "backscattering". The backscattering is proportional to the scene impulse response and indicates the amount of light returned to the camera sensor after a particular time-of-flight t:

$$x(t) \propto h(t) \tag{2.21}$$

Note how the measurement model above enforces a strict relationship between the correlation measure and the scene impulse response which is useful to derive information about the scene geometry from the acquired ToF data.

From a computational point of view, it is interesting to derive a discrete approximation of the previous model in order to be able to perform efficient computations using a finite precision machine. Discretizing the ToF interval of interest into N steps, the continuous model can be approximated with its discrete version:

$$v = \sum_{n=0}^{N-1} x_n \, e^{j2\pi f_m t_n} \qquad\qquad t_n = t_{min} + \frac{t_{max} - t_{min}}{N} n \tag{2.22}$$

Time and depth resolutions achievable using the discrete approximation depend on the number of steps N, in particular:

$$t_{res} = \frac{t_{max} - t_{min}}{N} \qquad\qquad d_{res} = \frac{d_{max} - d_{min}}{N} \tag{2.23}$$

Clearly, increasing the number of steps the approximation is going to be better and better, up to the limit for $N \rightarrow \infty$ where the discrete and the continuous models coincide. Note that, since the temporal variable represents the time taken by the light to cross the scene and the light moves at extremely high speed, the time resolution must be on the order of picoseconds.

In addition, the discrete model (2.22) can vectorized as a matrix multiplication between the measurement matrix $\Phi \in \mathbb{C}^{1\times N}$ and the so called "backscattering vector" $\boldsymbol{x} \in \mathbb{R}^{N\times 1}$:

$$v = \begin{bmatrix} e^{j2\pi f_m t_0} & \dots & e^{j2\pi f_m t_{N-1}} \end{bmatrix} \begin{bmatrix} x_0 \\ : \\ x_{N-1} \end{bmatrix} = \Phi \, \boldsymbol{x} \tag{2.24}$$

The backscattering vector $\boldsymbol{x}$ is the sampled version of the backscattering signal $x(t)$, and still remains proportional to the scene impulse response $h(t)$. Figure 2.5 reports the typical shape of the backscattering vector in the ideal case.

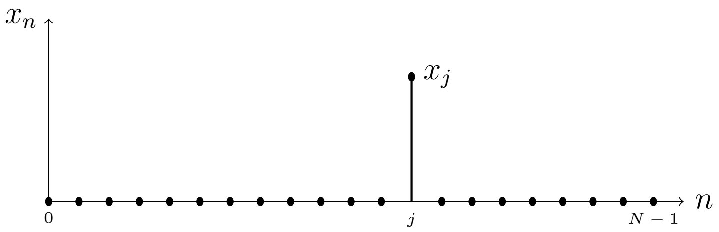


Figure 2.5: Backscattering vector in the MPI-free case. When the light bounces only once in the scene at some distance d^*, the backscattering vector is formed by a single non-zero element, *i.e.* $x_j > 0$ for $j = \lfloor (d^* - d_{min})/d_{res} \rfloor$, and $x_n = 0$ for $n \neq j$. The amplitude of the non-zero element mainly depends on distance and reflectance of the surface.

2.3 Depth Measurement Errors

With the evolution of time-of-flight technology, a lot of work has been devoted in order to understand the sources of errors [3, 4]. Depth errors can be classified into systematic errors and non-systematic errors. Generally, systematic errors can be managed by calibration while non-systematic are more difficult to be compensated because highly unpredictable. One common approach to deal with non-systematic errors is filtering.

2.3.1 Systematic Errors

Systematic errors occur when the formulas used for the reconstruction do not model all aspects of the actual physical layer. In AMCW cameras, a relevant error appears as consequence of the fact that in practice the emitted illumination signal does not follow exactly the theoretical one, due to the difficult generation of a perfectly sinusoidal signal. The actual modulation process introduces high order harmonics that induce a deviation from the perfect sine function. This error, typically refereed to as *wiggling* or *circular error*, produces an offset that depends only on the measured depth. The error plotted against the depth follows a sinusoidal shape. The actual form of this oscillation depends on the strength and frequencies of the higher order harmonics. This offset is typically compensated acquiring more samples of the correlation function and extending the formulas to incorporate higher order harmonics, or keeping the formulas are they are and estimating the residual error between the true and the measured depth during the calibration process.

In addition, it has been observed that the measured depth is greatly affected by the total amount of incident light received by the sensor. This error is know as *amplitude-related error*. The higher the reflected amplitude, the higher the depth accuracy. Low amplitude appears more often in the border of the image as the emitted power is lower than in the centre. Contrarily, when objects are too close to the camera saturation can appear and depth measures will be not valid. The origin of amplitude-related errors has been identified in the non-linearities of the semi-conductors forming. Discarding high and low amplitude pixels in the depth estimation avoids this kind of error but may lead to sparse depth maps. A different solution can be the combination of depth measurements from multiple range images with different exposure settings, or filtering.

Another phenomenon that introduces ambiguity in the depth estimation is its dependence on the chosen integration time. Acquiring the same scene with different integration times causes different depth values in the entire scene. The main reason behind this *integration-time-related error*, is sill a subject under investigation. The common strategy to solve this problem is to repeat the calibration procedure for all different integration times used.

The manufacturing process as well as the physical position of the pixels in the camera sensor produce an error characteristic for each pixel, namely *built-in pixel-related error*. Impurities in the semi-conductors introduce pixel-related depth offsets, leading to different depth measured by two neighbour pixels corresponding to the same real depth. On the other hand, the position of each pixel in the sensor array affects the capacitor charge time delay. The effect is a rotation of the image plane, *e.g.* a perpendicular flat surface appears with a different orientation. A common representation of this error is a fixed pattern noise obtained by comparing the measured depths with a reference distance.

A well-know source of error in all electronics devices is the temperature. *Temperature-related errors* happen because internal camera temperature affects the semi-conductors response introducing a depth drift in the whole image. To mitigate this error, new generation ToF cameras incorporate a fun to stabilize the internal temperature.

Other noise sources that introduce errors in the final depth estimation are the *shot noise*, originated from the discrete nature of the photon-to-electron conversion, and the *quantization noise*, characteristic of the analog-to-digital conversion process which transforms the collected electrons into a digital number. Typically these sources of noise are statistically characterized. Shot noise can be modelled by a Poisson process, while quantization noise has an approximately uniform distribution.

2.3.2 Non-Systematic Errors

Non-systematic errors are those who do not depend on the camera itself but by the actual acquired scene. These errors cannot be compensated through an accurate calibration process so more sophisticated techniques must be adopted. The typical low resolution of ToF cameras promotes *flying pixels* along depth discontinuities. In the case that the solid angle extent from a sensor pixel falls on the boundary between a foreground and a background, the recorded correlation measure is a mixture of the light returns from both regions. Due to the non-linearity of the depth on the raw data and to the phase wrapping effect, the resulting depth is not restricted to the range between the foreground and the background but may attain any value in the camera's depth range. In case of flying-pixels, local information from neighbouring pixels can be used to approximately reconstruct the true depth value.

Motion artefacts is another kind of non-systematic error that is present in traditional cameras used in dynamic environments, and that results even more relevant in ToF cameras. As already underlined, the reconstruction of the whole correlation function in the depth acquisition process requires a finite time interval. In particular, the correlation function needs to be sampled several times, stretching the overall acquisition time. In case of physical motion of the objects or of the camera in the scene during the acquisition time, moving parts appear displaced in subsequent acquisitions. This leads to inconsistent depth measurements that generate motion blurring effects, especially along depth discontinuities. Moreover, new generation ToF cameras acquire raw data at multiple modulation frequencies aggravating the effect of motion artefacts. The development of effective techniques to mitigate these artefacts is still a open research topic. From a physical point of view the use of special 4-tap lock-in pixels [5] allows to halve the required acquisition time and thus to reduce the impact of motion artefacts. Other proposals rely on an estimation of the optical flow in the scene and on a software compensation based on a theoretical movement model.

Other concerns include ambient light that may contain unwanted light of the same wavelength as that of the ToF projector which causes false depth measurements. Frequency-based filters can be used in order to minimize this effect.

Finally, the most significant source of error in ToF technology is the *multi-path interference effect*, produced by multiple returning light rays. The correction of this unpredictable error is the main objective of this work. Section 2.5 summarizes the main reasons behind this phenomenon and describes a mathematical model useful to characterize it.

2.4 Statistical Error Propagation Analysis

From the previous section it is clear that the raw data acquired by the ToF sensor are affected by noise. Consequently, the reconstructed correlation function is noisy as well. Here the objective is to analyse how the errors on the raw acquired samples propagate to the reconstructed correlation function. The *propagation of uncertainty* theory provides the mathematical framework to study the effect of variables' uncertainties on the uncertainty of a function based on them. Referring to equations (2.15), (2.14) and (2.16), the non-linear relationship between the raw data and the reconstructed amplitude, phase and intensity values can be rewritten as [1]:

$$\begin{aligned} f: \qquad & \mathbb{R}^4 & \mapsto \quad & \mathbb{R}^3 \\ \boldsymbol{m} = \left[m_0, m_{\pi/2}, m_{\pi}, m_{3\pi/2}\right]^T & & \mapsto \quad & f(\boldsymbol{m}) = [A, \varphi, I]^T \end{aligned} \tag{2.25}$$

where $f = \mathcal{X}_2 \circ \mathcal{X}_1$ with $\mathcal{X}_1$ being the linear mapping:

$$\mathcal{X}_1(\boldsymbol{m}) = \begin{bmatrix} \frac{1}{2} & 0 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \boldsymbol{m} \tag{2.26}$$

and:

$$\mathcal{X}_2([x,y,c]^T) = [\Phi(x,y)^{-1}, c]^T \qquad\qquad \Phi(A,\varphi) = [A\,cos\,(\varphi)\,, A\,sin\,(\varphi)]^T \tag{2.27}$$

Applying the error propagation analysis, the first order approximation of the noise on the output parameters is given by:

$$\Sigma_f = J_f \cdot \Sigma_m \cdot J_f^T \tag{2.28}$$

where Σ_m and Σ_f indicate the covariance matrices respectively of the input variable $\boldsymbol{m}$ and the output function value $f(\boldsymbol{m})$, while J_f is the Jacobian of f with entries $J_{f,ij} = \frac{\partial f_i}{\partial x_j}$.

The Jacobian, computed on the average recorded raw values, can be derived exploiting standard differentiation rules, yielding to:

$$J_{\mathcal{X}_1} = \begin{bmatrix} \frac{1}{2} & 0 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \tag{2.29}$$

$$J_{\Phi^{-1}} = J_{\Phi}^{-1} = \begin{bmatrix} cos\,(\varphi) & -A\,sin\,(\varphi) \\ sin\,(\varphi) & A\,cos\,(\varphi) \end{bmatrix}^{-1} = \begin{bmatrix} cos\,(\varphi) & sin\,(\varphi) \\ -\frac{1}{A}\,sin\,(\varphi) & \frac{1}{A}\,cos\,(\varphi) \end{bmatrix} \tag{2.30}$$

$$J_{\mathcal{X}_2} = \begin{bmatrix} cos\,(\varphi) & sin\,(\varphi) & 0 \\ -\frac{1}{A}\,sin\,(\varphi) & \frac{1}{A}\,cos\,(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.31}$$

$$J_f = J_{\mathcal{X}_2} \cdot J_{\mathcal{X}_1} = \frac{1}{2} \begin{bmatrix} cos\,(\varphi) & -sin\,(\varphi) & -cos\,(\varphi) & sin\,(\varphi) \\ -\frac{1}{A}\,sin\,(\varphi) & -\frac{1}{A}\,cos\,(\varphi) & \frac{1}{A}\,sin\,(\varphi) & \frac{1}{A}\,cos\,(\varphi) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.32}$$

Assuming the simple case where the measurements $m_0, m_{\pi/2}, m_\pi, m_{3\pi/2}$ are independent and identical distributed with some error σ. Under this assumption the input covariance matrix is given by $\Sigma_m = diag(\sigma^2, \sigma^2, \sigma^2, \sigma^2)$ while the output one results $\Sigma_f = diag(\frac{\sigma^2}{2}, \frac{\sigma^2}{2A^2}, \frac{\sigma}{4})$. The correlation function parameters turns out to be independent with errors respectively of:

$$\sigma_A = \frac{\sigma}{\sqrt{2}} \qquad \sigma_\varphi = \frac{\sigma}{\sqrt{2}A} \qquad \sigma_I = \frac{\sigma}{2} \tag{2.33}$$

Another common reasonable assumption is considering each raw acquired sample distributed according to a Poisson distribution. Therefore, since for a poisson distribution the variance is equal to the average value, the covariance input matrix can be expressed as $\Sigma_m = diag(\sigma^2_{m_0}, \sigma^2_{m_{\pi/2}}, \sigma^2_{m_\pi}, \sigma^2_{m_{2\pi/2}}) = diag(m_0, m_{\pi/2}, m_\pi, m_{2\pi/2})$. This assumption leads to a well-know formula for the error of the reconstructed phase value:

$$\sigma_\varphi = \frac{\sqrt{I}}{\sqrt{2}A} \tag{2.34}$$

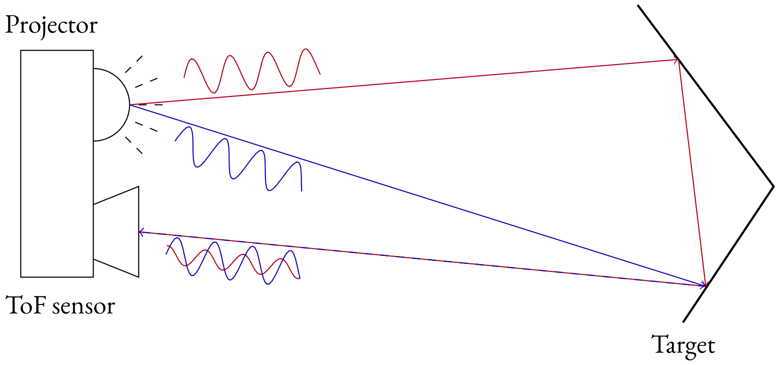


Figure 2.6: Generation of the MPI effect. The multi-path interference effect is caused by multiple propagation paths inside the scene which interference at the camera sensor.

The relation above holds on the average values and provides an estimation of the characteristic Signal-to-Noise Ratio (SNR) for the recorded signal, where the amplitude can be interpreted as the relevant signal while the square root of the intensity as the noise. Higher is the amplitude smaller is the error, and viceversa, higher is the intensity more noisy is the acquired correlation function. Finally, the error on the estimated depth is directly proportional to the error on the phase and depends strictly on the modulation frequency used. Higher modulation frequencies turn out to be more noise resilient:

$$\sigma_d = \frac{c\,\sigma_\varphi}{4\pi f_m} = \frac{c}{4\pi f_m}\frac{\sqrt{I}}{\sqrt{2}A} \tag{2.35}$$

2.5 Multi-Path Interference Effect

One of the main sources of error in ToF technology is the Multi-Path Interference (MPI) phenomenon. The basic principle of AMCW cameras relies on the hypothesis that each pixel receives a single optical ray emitted by the projector which bounced only once in the scene. This assumption, unfortunately, is violated in most real-world scenarios and each pixel receives a superposition of multiple optical rays coming from different points of the scene. This phenomenon introduces a non-systematic error in the depth estimation, typically very difficult to be corrected since it is highly scene-dependent and thus unpredictable.

The MPI effect can occur either intra-camera, due to the light reflection and scattering with the imaging lens and aperture, or extra-camera. Extra-camera multiple returns are caused by multiple propagation paths between the light source and the sensor's pixels. The primary return, called *direct component*, is the one that covers the smaller distance inside the scene and thus associated with the true depth value. All the other interfering rays, associated to higher order reflections, fall under the name of *global component*. Typically the direct component

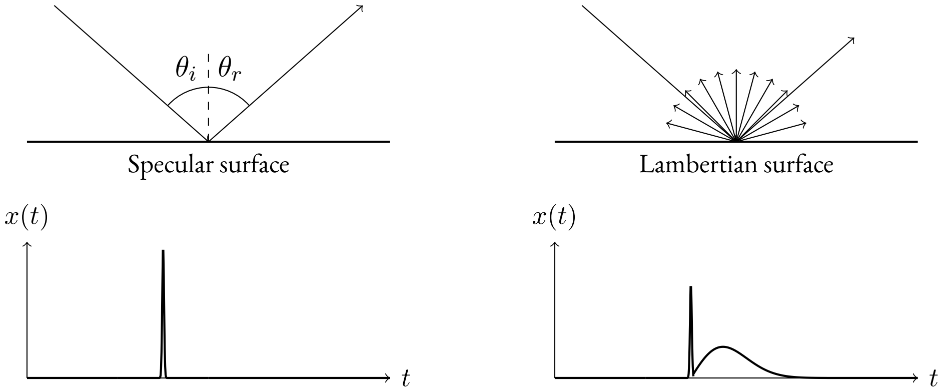


Figure 2.7: Specular and diffuse reflections. The figure shows the difference between specular (left) and diffuse (right) reflections. Specular surfaces reflect the light along a single direction, while lambertian surfaces diffuse the incident energy equally in all directions. The plots below report the characteristic shape of the backscattering vector in the two cases, specular reflections appear very concentrated in time while diffuse reflection are more spread.

results also the brightest one, though it need not be. Figure 2.6 reports an example of the generation process of the MPI effect.

There are two main types of interfering optical rays, those generated by *specular reflections* and those generated by *diffuse reflections*. Specular returns are typically caused by the reflection of the light with a specular surface, where all the incident energy is reflected in a singular direction according to the law of reflection. Conversely, diffuse returns are associated to the scattering of the light with a lambertian surface. Lambertian surfaces scatter incident illumination equally in all directions. Ideal lambertian surfaces are not physically plausible. Although, many real-world matte surfaces can be well approximated by a combination of both specular and diffuse reflections. In the following, we will use the name "diffuse reflection" to indicate this situation. The two different types of reflections generate a different MPI effect, as illustrated in figure 2.7. Specular reflections appear very concentrated in the time domain, while diffuse reflections present a first thin component followed by a more spread tail. This difference translates in a different shape of the backscattering signal.

2.5.1 Mathematical Description

The MPI effect leads to a coherent superposition of multiple interfering light signals at the camera sensor. Since, in general, multiple returns have covered different distances and have bounced on different surfaces inside the scene, they add up with different amplitudes and phases and thus the resulting correlation measure does not represent the real depth value.

For instance, the correlation measure generated by two specular reflections at time t_1 and

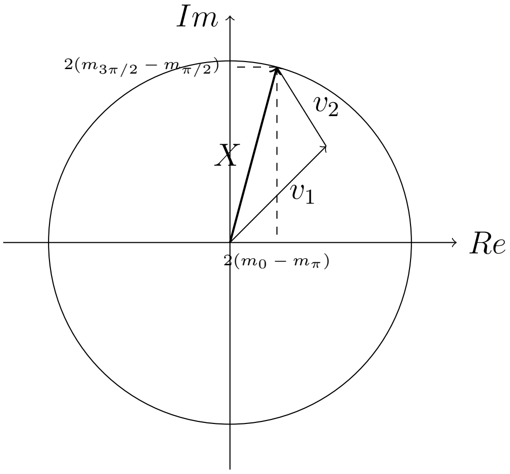


Figure 2.8: Phasor representation of the correlation function in a MPI case. The recorded correlation function in case of two specular reflections is given by the superposition of two interfering returns and therefore it does not reflect the real depth value.

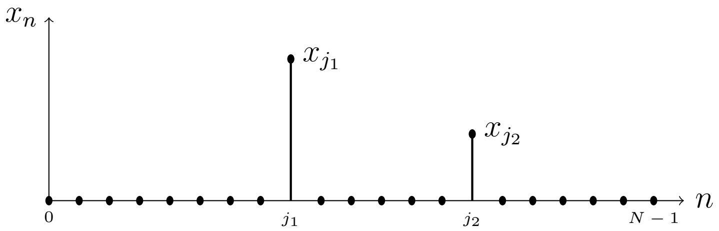


Figure 2.9: Backscattering vector in the MPI case. Considering two specular reflections at time t_1 and $t_2 > t_1$, the backscattering vector is formed by two non-zero peaks, *i.e.* $x_{j_1}, x_{j_2} > 0$ for $j_i = \lfloor (t_i - t_{min})/t_{res} \rfloor$, and $x_n = 0$ for $n \neq j_1, j_2$.

$t_2 > t_1$ can be expressed in phasor notation as shown in figure 2.8:

$$v = X_1 \, e^{j2\pi f_m t_1} + X_2 \, e^{j2\pi f_m t_2} = X \, e^{j\varphi} \in \mathbb{C} \tag{2.36}$$

Note that the measurement model derived in section 2.2 holds also in a MPI scenario, the difference is given by the shape of the backscattering vector. In general, the backscattering may have an arbitrary complicated envelope but it always enforces a relationship between the scene impulse response and the raw data acquired by the ToF camera. The typical shape of the backscattering signal in case of two specular reflections is reported in figure 2.9.

2.6 Multi-Frequency Acquisition

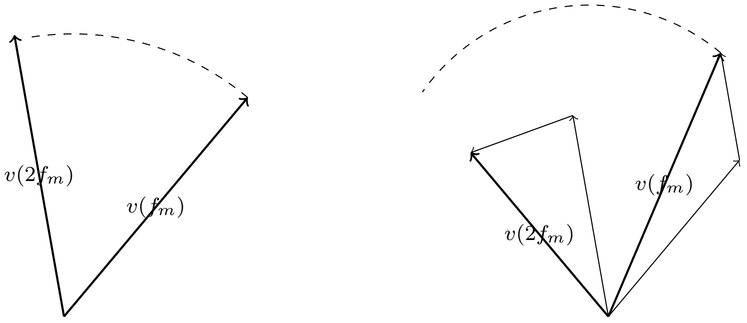
A single correlation measure in a MPI scenario does not allow to separate of multiple incoming interfering rays. Since the behaviour of each reflected optical ray depends on the modulation frequency used, one common approach to deal with the MPI phenomenon is to acquire raw data at $M \geq 1$ modulation frequencies. The additional information provided by multiple correlation measures can be exploited in order to mitigate the effect of interfering rays, trying to distinguish between direct and global components. Multi-frequency ToF data are acquired subsequently, illuminating the scene with M amplitude modulated signals at the frequencies $f_0, f_1, ..., f_{M-1}$ and capturing the M corresponding correlation functions. The overall acquisition time becomes $M\ K\ T_{int}$. Another advantage of a multi-frequency acquisition is that different frequencies have different unambiguous ranges that can be combined in order to extend the overall unambiguous range, without reducing the minimum depth resolution associated always to the higher frequency. This operation can be performed efficiently exploiting the chinese remainder theorem [8].

An intuitive explanation of the fact that capturing correlation measures at multiple modulation frequencies provides additional information about the MPI phenomenon comes from the following observation, schematically illustrated in figure 2.10. Let $v(f_m)$ and $v(2f_m)$ be the two phasors acquired at the base modulation frequency f_m and at the doubled frequency $2f_m$. In a multi-path free case, the two phasors are expected to have equal amplitudes and double phases, *i.e.* $|v(2f_m)| = |v(f_m)|$ and $arg(v(2f_m)) = 2\, arg(v(f_m))$. However, this is not true in case of MPI. For instance, if the MPI is produced by two specular reflections $v = v_1 + v_2$, the previous observation holds for each single return, *i.e.* $|v_i(2f_m)| = |v_i(f_m)|$ and $arg(v_i(2f_m)) = 2\, arg(v_i(f_m))$ for $i = 1, 2$, but not for their superposition. Due to the interfering effect, the camera records two phasors such that $|v(2f_m)| \neq |v(f_m)|$ and $arg(v(2f_m)) \neq 2\, arg(v(f_m))$. Therefore, looking at the phasors acquired at different modulation frequencies we can get some clues about the underlying multi-path interference effect.

The measurement model (2.24) can be easily generalized to the multi-frequency acquisition scenario. In this case the camera will acquire for every pixel a complex vector $\boldsymbol{v} \in \mathbf{C}^{M \times 1}$ representing the M recorded phasors. The relationship between the acquired phasors and the scene impulse response is always in the form:

$$\boldsymbol{v} = \begin{bmatrix} v_0 \\ \vdots \\ v_{M-1} \end{bmatrix} = \begin{bmatrix} e^{j2\pi f_0 t_0} & \dots & e^{j2\pi f_0 t_{N-1}} \\ \vdots & \ddots & \vdots \\ e^{j2\pi f_{M-1} t_0} & \dots & e^{j2\pi f_{M-1} t_{N-1}} \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ x_{N-1} \end{bmatrix} = \Phi\, \boldsymbol{x} \qquad (2.37)$$

where now the measurement matrix $\Phi \in \mathbb{C}^{M \times N}$ accounts for all the M modulation frequencies. Note that the previous relationship holds because changing the modulation frequency, the phase of each interfering ray changes but the amplitude remains constant. This means that the backscattering vector is uniquely defined independently from the modula-



MPI-free case MPI case

Figure 2.10: Frequency diversity of the MPI effect. On the left it is reported the multi-path free case for which the amplitude of the acquired phasors is constant and the phase is linear with the modulation frequency. In the two specular reflections case on the right the previous observation holds for each single interfering component but not for their superposition.

tion frequencies used during the acquisition process, in fact it indicates the amount of light returned.

A further modification can be introduced to turn the complex measurement model into a real one. In particular, stacking the real part of the previous system of equations on top of its imaginary part, it results:

$$\boldsymbol{v} = \begin{bmatrix} Re[v_0] \\ \vdots \\ Re[v_{M-1}] \\ Im[v_0] \\ \vdots \\ Im[v_{M-1}] \end{bmatrix} = \begin{bmatrix} cos\,(j2\pi f_0 t_0) & \cdots & cos\,(j2\pi f_0 t_{N-1}) \\ \vdots & \ddots & \vdots \\ cos\,(j2\pi f_{M-1} t_0) & \cdots & cos\,(j2\pi f_{M-1} t_{N-1}) \\ sin\,(j2\pi f_0 t_0) & \cdots & sin\,(j2\pi f_0 t_{N-1}) \\ \vdots & \ddots & \vdots \\ sin\,(j2\pi f_{M-1} t_0) & \cdots & sin\,(j2\pi f_{M-1} t_{N-1}) \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ x_{N-1} \end{bmatrix} = \Phi\,\boldsymbol{x} \tag{2.38}$$

With an abuse of notation, we will use the same terminology for the complex measurement model (2.37) and the real one (2.38). They express the same relationship in two slightly different mathematical ways.

2.7 Spatial Data Acquisition

One advantage of ToF cameras is the possibility to acquire dense data in one-shot. This means that the acquisition process discussed in the previous sections happens simultaneously for each pixel of the camera sensor. Let $W \times H$ be the resolution of the ToF camera, where W indicates the width and H the height of sensor in number of pixels. Note that typically a ToF camera has a lower resolution than a standard RGB camera.

The derived measurement model (2.37) holds for each single pixel, but can be reformulated to characterize the whole sensor behaviour. Let $\mathcal{V} \in \mathbb{C}^{W\times H\times M}$ be a three dimensional complex tensor such that:

$$\mathcal{V}_{uv} = \boldsymbol{v}^{uv} = \begin{bmatrix} v_0^{uv} \\ \vdots \\ v_{M-1}^{uv} \end{bmatrix} \qquad \begin{cases} u = 0, \dots, W-1 \\ v = 0, \dots, H-1 \end{cases} \tag{2.39}$$

and let $\mathcal{X} \in \mathbb{R}^{W\times H\times N}$ be the three dimensional real tensor such that:

$$\mathcal{X}_{uv} = \boldsymbol{x}^{uv} = \begin{bmatrix} x_0^{uv} \\ \vdots \\ x_{N-1}^{uv} \end{bmatrix} \qquad \begin{cases} u = 0, \dots, W-1 \\ v = 0, \dots, H-1 \end{cases} \tag{2.40}$$

where $\boldsymbol{v}^{uv}$ and $\boldsymbol{x}^{uv}$ represent respectively the multi-frequency phasors and the backscattering vector associated to the pixel with coordinates (u, v). The tensor $\mathcal{X}$ proportional to the whole scene impulse response is commonly referred to as *transient scene*. With the new notation, the measurement model results in:

$$\mathcal{V} = \Phi\,\mathcal{X} \tag{2.41}$$

The only difference is that instead of using the standard matrix multiplication it is necessary to use the so called n-mode product between tensors. In order to simplify the notation, the convention in our work is that the n-mode product between a matrix and a tensor will correspond always to a standard matrix multiplication between the matrix itself and the last dimension of tensor reshaped into a column vector, that is:

$$\mathcal{V}_{uvm} = \sum_{n=0}^{N-1} \Phi_{mn}\,\mathcal{X}_{uvn} = \sum_{n=0}^{N-1} \mathcal{X}_{uvn}\,e^{j2\pi f_m t_n} \qquad \begin{cases} u = 0, \dots, W-1 \\ v = 0, \dots, H-1 \\ m = 0, ..., M-1 \end{cases} \tag{2.42}$$

If I have seen further than others, it is by standing upon the shoulders of giants.

Isaac Newton

3 Literature Review

Many authors in the literature have faced the MPI effect proposing different methods to correct it, but this still remains an open research field [2]. The development of effective methods for multi-path interference compensation is a deeply studied topic, since it would resolve the main problem of ToF cameras promoting the diffusion of this technology to a wide range of real-world applications. Most of the methods proposed in the literature can be organized basically into four main categories. The first category contains methods which combine the ToF technology to other range imaging technique in order to obtain an improve depth estimation. The second one uses a single modulation frequency and an iterative procedure to adjust the scene geometry and compensate for the MPI effect. The third family contains those methods which acquire data at multiple modulation frequencies and rely on physical model to characterize and correct the MPI phenomenon. Finally, in the last category there are all the methods which employ a neural network architecture in order to learn from data the best strategy to correct the MPI effect. In the following sections we are going to present all these categories, highlighting the more relevant representatives and their main limitations.

3.1 Combined Approaches

An approach that has been proposed to improve the final depth estimation is the combination of ToF technology with other rage imaging techniques. The fusion of data coming from different sources allows to reduce the depth reconstruction error, limiting the multi-path effect typical of ToF acquisitions.

The methods in [21, 22] combine ToF and stereo technologies, exploiting the complementary characteristics of the two systems. On the other hand, in [23, 24, 25, 26] the authors propose to use a modified ToF projector able to emit a spatial high frequency pattern in order to separate the global and direct component of the light, and so correct MPI.

Method	MPI type	No. mod. frequencies	Real-time	Comment
Fuchs *et al.* [9]	2-Diffuse	1	No	Highly time-consuming
Fuchs *et al.* [10]	2-Diffuse	1	No	Highly time-consuming
Jiménez *et al.* [11]	1-Diffuse	1	No	Highly time-consuming
Dorrington *et al.* [12]	2-Specular	2	No	Solve minimization problem
Godbaz [13]	2-Specular	4	Yes	
Kirmani *et al.* [14]	2-Specular	5	Yes	Spectral reconstruction
Bhandari *et al.* [15]	3-Specular	77	n.a.	OMP recovery algorithm
Freedman *et al.* [7]	R-Specular	3	Yes	Sparse reconstruction
Peters *et al.* [16]	R-Specular	$M \geq R$	Yes	Burg entropy minimization
Marco *et al.* [17]	General	n.a.	Yes	MPI as time-varying convolution
Guo *et al.* [18]	Diffuse	3	Yes	
Agresti *et al.* [19]	General	3	n.a.	
Agresti *et al.* [20]	General	3	n.a.	Unsupervised domain adaptation

Table 3.1: Comparison between some state-of-the-art approaches for MPI compensation.

3.2 Iterative Single-Frequency Approaches

The second family contains those methods that acquire data using a single modulation frequency and rely on a radiometric model for the light in order to iteratively adjust the geometry of the scene and correct the MPI effect.

Fuchs *et al.* in [9] present an algorithm where a two bounces scenario on ideal lambertian surfaces is considered. The algorithm works iteratively performing inverse ray tracing. At each step it refines the scene geometry removing the multi-path contribution modelled according to a specific reflection model. The algorithm starts from the acquired corrupted depth measurements and it is expected to converge to the actual scene geometry. In [10] they refine this approach by improving the reflection model and taking into account materials with multiple albedo.

Jiménez *et al.* in [11] propose a method based on a similar idea. They develop a radiometric model assuming perfect lambertian surfaces and that the multi-path effect comes only from points within a neighbour region of each pixel. Multi-bounces are not considered. Jiménez *et al.* characterize the multi-path effect through a cost function and use the single-frequency ToF measurements to recover the actual scene structure solving an iterative optimization problem.

Both these approaches provide good correction capabilities, forcing the final scene geometry to be consistent with its own measured MPI. The main limitation is that the iterative refinement procedure is highly time consuming, refusing the possibility to use these methods in a real-time applications.

Finally, there is a study from Gupta *et al.* [6] which observes that diffuse multi-path reflections tends to cancel out at high frequencies, thus they propose to reconstruct range from few measurements at high frequency. Unfortunately this approach does not account for specular reflections and, moreover, it turns out to be unpracticable since it would require a very high modulation frequency difficult to be generated in the real-world.

3.3 Model-Based Multi-Frequency Approaches

Conversely, another approach is to use multiple modulation frequencies and exploit the frequency diversity of the MPI phenomenon to correct it. This third category contains all the methods based on a predefined physical model to describe the multi-path interference and to separate direct and global components. In order to have a mathematical treatable model it is necessary to make some simplifying assumptions about the nature of the interfering rays. The most common assumption is to approximate the received signal as the sum of a first direct return and a second global path, with both components produced by specular reflections.

In [12] Dorrington *et al.* propose to use two modulation frequencies with ratio 2:1 to separate direct and global components solving a non-linear minimization problem.

Godbaz in [13] uses two frequencies and a lookup table to resolve multi-path interference in real-time. In his Ph.D. work he also provides a closed form solution which requires four modulation frequencies.

Kirmani *et al.* acquire data at five modulation frequencies and adopt a two steps approach [14]. First they detect if a pixel is affected by MPI, then if it is not they use the standard ToF formulas, otherwise amplitude and phase of the two specular returns are estimated using some spectral estimation techniques. In particular, they use the Prony's method in the noiseless case and a singular value based method when noise is present.

Bhandari *et al.* generalize the problem using $M > 4$ frequencies to reconstruct $R > 2$ specular returns exploiting the well-studied compressed sensing theory [15]. Amplitude and phase of each return are estimated via Orthogonal Matching Purists (OMP) recovery algorithm. They proved the proposed method reconstructing three specular reflections acquiring raw data using 77 modulation frequencies.

The SRA method [7] proposed by Freedman *et al.* uses three modulation frequencies to estimate the backscattering vector assuming only its sparsity. It accounts only for specular reflections, even if the authors declares that it is able to achieve good results also in a more general multi-path scenario. The idea behind the SRA method is to reconstruct the shape the backscattering vector finding the sparsest vector compatible with the acquired measurements. To this end it defines an L0-minimization problem which solves introducing some approximations. The final solution is computed solving a linear program. The authors focus also in the computational speed. Using a lookup table they are able to run the method in real-time at 30 fps.

In [16] Peters *et al.* suggest to use M modulation frequencies to reconstruct the shape of the whole backscattering vector by minimizing the Burg entropy. The method guarantees an exact reconstruction only in the case of M specular returns or less, while if this condition is not met it reflects uncertainty in the reconstruction by smoothing the peaks.

Some of these methods have been successfully applied to real-world applications but their effectiveness remains constrained to the simplifying assumptions done for the MPI model derivation. The main limitation is that the predefined physical model represents only an approximation which does not reflect the actual multi-path phenomenon. These methods exhibits good performance in the correction of few specular reflections, but they are not able to deal with diffuse reflections therefore they inevitable fail on more complex scenarios.

3.4 Data-Driven Approaches

The last category overcomes the problem of having a predefined physical model for the multi-path interference. In this category there are all the methods based on some deep learning algorithm to learn from data the best strategy to correct the MPI effect. In this way, the learned model should be able to capture complex multi-path scenarios, accounting either for specular and diffuse reflections. The main precondition to run these algorithms is the availability of a large number of training samples from which the network can learn.

Marco *et al.* in [17] assume that most of the information about multi-path interference from a scene is available in image space and can be modelled as a spatially-varying convolution. Therefore MPI compensation could be achieved by a set of convolutions and deconvolutions operations in the depth space. They use a convolutional autoencoder trained through a particular two-stages approach which outputs the MPI corrected version of the input depth map.

Guo *et al.* introduce a learning-based approach to tackle dynamic scenes, multi-path interference, and shot noise simultaneously [18]. They use a convolutional architecture which works on raw multi-frequency ToF measurements and produces improved measurements that are compatible with standard equations for phase-unwrapping and conversion to depth. Together with their research, Guo *et al.* release also the FLAT (Flexible, Large, Augmentable, ToF dataset) dataset, a synthetic dataset containing static and dynamic transient scenes for the training and the evaluation of deep-leaning approaches in the ToF context.

In [19] Agresti *et al.* design a coarse to fine convolutional neural network that works on amplitude and depth maps acquired at three modulation frequencies to correct the MPI effect.

The key issue of using deep learning techniques in this range imaging field is the limited availability of real-world training data with ground truth. Different from classic computer vision problems there are not large datasets publicly available due to the fact that the acquisition of depth and raw ToF measurements with ground truth in a real scenario is a very complex and time consuming task. Many deep learning approaches in the literature are trained on synthetic data, where light transport events are simulated via software using some ray tracing techniques [27]. Unfortunately, it has been proven [20] that very well performing algorithms trained on the synthetic case perform poorly in the real case due to the inconsistency between real and simulated data. To solve this problem, Agresti *et al.* propose an interesting unsupervised domain adaptation technique to train the network on synthetic data and then adapt it to the real case in a unsupervised manner [20].

The thing that's going to make artificial intelligence so powerful is its ability to learn, and the way AI learns is to look at human culture.

Dan Brown

4

Proposed Deep Learning Approach

The backscattering vector contains important information about scene geometry, therefore its knowledge may be very useful in a wide range of applications. Capturing the propagation behaviour of the light at extreme temporal resolution can be exploited to see around corners [28], detect objects in highly-scattering media [29], infer material properties from a distance [30, 31], as well as perform optimal MPI correction. The direct acquisition of transient information requires highly expensive measurements tools based on femto-photography or interferometry-based systems [32]. Some attempts have been made in the literature to reconstruct that information indirectly from a sparse set of measurements, like the one provided by AMCW ToF technology. As we have seen, the acquisition process of a ToF camera can be described through a linear measurement model 2.37 which enforces a relationship between the backscattering vector $\boldsymbol{x} \in \mathbb{R}^N$ and the raw data $\boldsymbol{v} \in \mathbb{C}^M$ acquired by the camera. The vector $\boldsymbol{\eta} \in \mathbb{C}^M$ accounts for all the different sources of errors that affect a real ToF acquisition. The overall measurement model results in:

$$\boldsymbol{v} = \Phi \, \boldsymbol{x} + \boldsymbol{\eta} \tag{4.1}$$

The main objective of this work is to develop a method able to invert the previous system in order to estimate the unknown backscattering vector starting from noisy multi-frequency raw data. This is an ambitious task, because the underlying system is highly underdetermined, *i.e.* $M << N$, and we aim to recover the whole signal from far fewer samples than required by the Nyquist–Shannon sampling theorem. This problem can be reformulated in the compressed sensing framework [33], but in a sense it is even more challenging due to the small number of measures available to reconstruct the final signal. To have a rough idea about the Degrees of Freedom (DoFs) in the solution, off-the-shelf ToF cameras typical acquire data using $M = 3$ modulation frequencies and in order to get a reasonable depth resolution the backscattering vector should be discretized into $N = 500 \div 1000$ steps. The

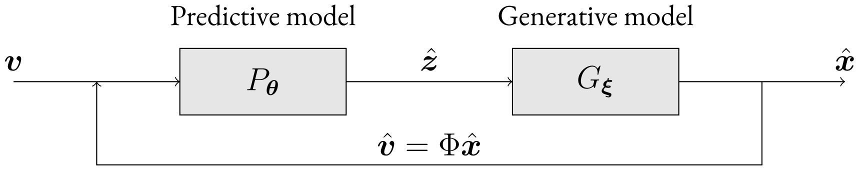


Figure 4.1: Proposed deep learning approach. The generative model generates the output backscattering vector starting from its compressed representation, while the predictive model predicts the most likely latent variable that satisfies the underlying measurement model.

estimation of the backscattering vector is an ill-posed problem that does not admit a clear unique solution, therefore to be solved it requires the introduction of some additional strong constraints. The SRA method proposed by Freedman *et al.* [7] estimates the backscattering vector relying only on its sparsity, but this is not sufficient and leads to limited performance in real-world applications. Due to the propagation and reflection properties of the light in a real environment, typically the backscattering vector presents some very characteristic structures we can take advantage of. The idea is to use a deep neural network architecture in order to learn the underlying reflection structure and use it as strong prior to optimize the estimation of the backscattering vector. Deep networks turn out to be the perfect mathematical tools to accomplish the proposed task, since they are powerful machine learning models that have been proved to be able to capture complex structures on data and outperform other state-of-art algorithms on classification, regression as well as generation tasks [34].

More in details, the proposed deep learning approach relies on two main ingredients. The first one is a *generative model* $G_{\boldsymbol{\xi}}$ that starting from a compressed representation $\hat{\boldsymbol{z}}$ generates in output the corresponding backscattering vector $\hat{\boldsymbol{x}} = G_{\boldsymbol{\xi}}(\hat{\boldsymbol{z}})$, while the second one is a *predictive model* $P_{\boldsymbol{\theta}}$ that takes in input the raw ToF data $\boldsymbol{v}$ and predicts the most likely latent variable $\hat{\boldsymbol{z}} = P_{\boldsymbol{\theta}}(\boldsymbol{v})$ that satisfies the underlying measurement model (4.1). In this way we can decouple the problem of generating output vectors with a particular structure by the problem of predicting the output vector given the input ToF measures. This pipeline allows to gain a finer control over the part of the network that captures the characteristic structures of backscattering vectors, and helps to mitigate the negative effect of the high number of DoFs in the solution. As already pointed out, there is a significant difference between the input and output dimensions. Using a single model that starts from the M input variables and outputs an N-dimensional backscattering vector, with $M << N$, leads to instabilities during the optimization phase and to a network converging to a bad local minima. In contrast to directly optimising the signal $\boldsymbol{x}$, the proposed pipeline performs the optimisation in the domain of the latent representation $\boldsymbol{z}$, lying in a lower dimensional space by definition.

4.1 Generative Model

Let $\mathcal{D}_{\boldsymbol{x}}^* \subseteq \mathbb{R}^N$ be the signal space where all possible backscattering vectors lie, it depends on the propagation and reflection properties of the light in a real environment and must account for both specular and diffuse reflections. Note that it represents our prior knowledge about the shape and the typical structure of each backscattering vector. The *generative model* $G_{\boldsymbol{\xi}}$ maps a latent variable $\boldsymbol{z} \in \mathbb{R}^L$ into an approximation of the backscattering domain $\mathcal{D}_{\boldsymbol{x}} \approx \mathcal{D}_{\boldsymbol{x}}^*$:

$$\begin{aligned} G_{\boldsymbol{\xi}}: \quad \mathbb{R}^L &\rightarrow \quad \mathcal{D}_{\boldsymbol{x}} \subseteq \mathbb{R}^N \\ \boldsymbol{z} &\rightarrow \quad \boldsymbol{x} = G_{\boldsymbol{\xi}}(\boldsymbol{z}) \end{aligned} \tag{4.2}$$

It must implement a non-linear mapping, parametrized by weights $\boldsymbol{\xi}$. Imposing the constraint $L << N$, the generative model, through its architecture and its weights, implicit ensures that the output backscattering vector reflects a particular structure. The goal is to define a model whose range $\mathcal{D}_{\boldsymbol{x}}$ provides a good approximation of the actual domain $\mathcal{D}_{\boldsymbol{x}}^*$. The generative model can be implemented through an fixed analytical model, or can be obtained using some more advanced deep learning generative strategies, such as Variational Autoencoders (VAEs) or Generative Adversarial Networks (GANs).

A VAE is formed by an encoder followed by a decoder, trained to minimize the reconstruction error between input and output. In this way, the encoder learns to build a compressed representation of the input which preserves most of the original information and that can be used alone by the decoder to reconstruct the original data. In addition, VAEs are specifically designed to create continuous latent spaces, in order to allow the possibility to sample from the latent space to generate completely new output data never seen during the training but that reflect the original data distribution. VAEs have been successfully applied to different generation tasks. In our context, a VAE can be trained to mimic the typical structure of real backscattering vectors. The decoder of a trained VAE corresponds exactly to our generative model, which given a continuous latent variable produces an output backscattering vector reflecting the probability distribution seen during the training.

A GAN is another powerful generative neural network trained simulating a zero-sum non-cooperative game between two agents, a generator and a discriminator. The generator takes in input some random noise sampled from a latent space and tries to generate new candidates distributed according to a real data distribution, on the other hand, the objective of the discriminator is to distinguish between samples produced by the generator from real samples. The generator is trained to fool the discriminator by producing novel candidates that the discriminator thinks are not synthesized. As the training goes on, the generator learns to produce better candidates, while the discriminator becomes more skilled at flagging synthetic data. At the Nash equilibrium, the output of the discriminator should be a random choice, since it should not be able to distinguish between real or synthetic data. In our context, the discussed generative model corresponds exactly to the generator of a trained GAN.

It is possible to think also to other approaches to build the proposed generative model, but in any case it must be the first part of our pipeline to be designed and to be fixed since the subsequent optimization of the predictive model will rely on it. One fundamental requirement for the generative model is that it must be differentiable in order to have the possibility to apply the backpropagation algorithm for the optimization of the predictive model. In other words, it must be possible to compute the partial derivatives of the output backscattering vector with respect to the input latent variables, *i.e.* $\exists \frac{\partial x_n}{\partial z_i}, \; \forall \, n = 0, ..., N-1, \;\; i = 0, ..., L-1$.

4.2 Predictive Model

The second ingredient is the *predictive model* $P_{\boldsymbol{\theta}}$ that, given in input the raw ToF data $\boldsymbol{v} \in \mathbb{C}^M$, is going to predict the latent variable $\hat{\boldsymbol{z}} \in \mathbb{R}^L$ which is the most likely solution of the underdetermined measurement model (4.1), under some noise tolerance:

$$\begin{aligned} P_{\boldsymbol{\theta}} : \quad & \mathbb{C}^M \quad \rightarrow \quad \mathbb{R}^L \\ & \;\; \boldsymbol{v} \quad \rightarrow \quad \hat{\boldsymbol{z}} = P_{\boldsymbol{\theta}}(\boldsymbol{v}) \\ & \text{subject to } \textit{soft-constraint}: \\ & ||\boldsymbol{v} - \Phi \, G_{\boldsymbol{\xi}}(\hat{\boldsymbol{z}})|| = 0 \end{aligned} \tag{4.3}$$

Our proposal is to define a neural network architecture for the predictive model, parametrized by weights $\boldsymbol{\theta}$. The predictive model has to be optimized in a supervised manner, minimizing a suitable loss function in order to learn the relationship between inputs and outputs. In contrast to directly optimize the backscattering vector $\boldsymbol{x}$, the optimization is carried on in the space of latent representation $\boldsymbol{z}$. At each optimization step, the output latent variable $\hat{\boldsymbol{z}}$ is mapped into the corresponding backscattering vector $\hat{\boldsymbol{x}} = G_{\boldsymbol{\xi}}(\hat{\boldsymbol{z}})$, keeping fix the generative model $G_{\boldsymbol{\xi}}$.

The design of a "good" loss function, which satisfies some desirable properties, is a critical point that we are going to discuss later. For now, since the predictive model has to accomplish two goals simultaneously, let's assume that also its loss function can be divided into two main terms. The first term, called *measurement error* ℓ_m, will quantify how consistent is the final prediction with the recorded correlation measures. In others word it forces the predicted backscattering vector to be a solution of the underdetermined system (4.1) under some noise tolerance.

$$\ell_m = \ell_m(\boldsymbol{v}, \Phi \, \hat{\boldsymbol{x}}) \tag{4.4}$$

The second term, called *reconstruction error* ℓ_r, will measure the goodness of the network to fit the ground truth, and therefore it tries to select between all possible solutions the most likely one. In this way the network will learn the typical probability distribution of the solution from the data ground truth.

$$\ell_r = \ell_r(\boldsymbol{x}, \hat{\boldsymbol{x}}) \tag{4.5}$$

The overall loss function, computed for each single training sample $(\boldsymbol{v}, \boldsymbol{x})$, is given by:

$$\ell(\boldsymbol{v}, \boldsymbol{x}) = \ell_m(\boldsymbol{v}, \Phi\, \hat{\boldsymbol{x}}) + \lambda\, \ell_r(\boldsymbol{x}, \hat{\boldsymbol{x}}) \tag{4.6}$$

where the parameter $\lambda \in \mathbb{R}_+$ controls the trade-off between the two contributions.

Theoretically speaking, let $p_{data}(\boldsymbol{v}, \boldsymbol{x})$ be the data-labels distribution for our task, and let's assume we have available a proper training set $S = \{(\boldsymbol{v}_i, \boldsymbol{x}_i)\}_{i=0}^{M-1}$ sampled from $p_{data}(\boldsymbol{v}, \boldsymbol{x})$. The optimization of the proposed predictive model will be carried on according to the Empirical Risk Minimization (ERM) rule, which aims to minimize the *true error* $\mathcal{L} = \mathbb{E}_{(\boldsymbol{v},\boldsymbol{x})\sim p_{data}}\big[\ell(\boldsymbol{v}, \boldsymbol{x})\big]$ over the whole data distribution by minimizing the *empirical error* over the available training set:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_S = \frac{1}{M} \sum_{(\boldsymbol{v}_i, \boldsymbol{x}_i)\in S} \ell(\boldsymbol{v}_i, \boldsymbol{x}_i) \tag{4.7}$$

For a sufficient large training set, the theoretical PAC learnability principle ensures an upper bound for the gap between empirical and true errors. In practice, the ERM rule can be implemented employing some gradient descent strategies and using the backpropagation algorithm to compute all the partial derivatives of the output with respect to the predictive model's weights, *i.e.* $\frac{\partial x_n}{\partial \theta_i}$, $\forall\, \theta_i \in \boldsymbol{\theta}$. Note that since the generative model is fixed during this optimization we have $\frac{\partial x_n}{\partial \xi_i} = 0$, $\forall\, \xi_i \in \boldsymbol{\xi}$. Moreover, we recall that, since the first step of the backpropagation algorithm is the computation of the partial derivatives of the output backscattering vector with respect to its latent representation, the generative model must be differentiable.

4.3 Pixel-level and Local level Prediction

In the previous section we proposed a *predictive model* that, for each pixel, takes in input the multi-frequency phasors vector $\boldsymbol{v}$ and predicts the most likely latent variable $\hat{\boldsymbol{z}}$ that satisfies underlying measurement model (4.1). In section 2.7, we have also said that, since ToF cameras are able to acquire in one-shot the raw correlation measures for each pixel in the camera sensor, the linear relation between the transient scene $\mathcal{X} \in \mathbb{R}^{W\times H\times N}$ and the dense tensor of raw data $\mathcal{V} \in \mathbb{C}^{W\times H\times M}$ recorded by the camera can be equivalently formulated as:

$$\mathcal{V} = \Phi\, \mathcal{X} + \mathcal{H} \tag{4.8}$$

Note that this relation is equivalent to (4.1) but looking at the acquisition process at the whole sensor level, where the tensor $\mathcal{H} \in \mathbb{C}^{W\times H\times M}$ accounts always for the noise.

There are two main possibilities to invert this system and reconstruct the unknown transient scene. The first option is to work at single pixel-level as detailed before, this means that the predictive model, in order to produce the output latent variable for pixel (u, v), looks only at the corresponding input pixel:

$$\mathcal{Z}_{uv} = P_{\boldsymbol{\theta}}\Big(\mathcal{V}_{uv}\Big) \tag{4.9}$$

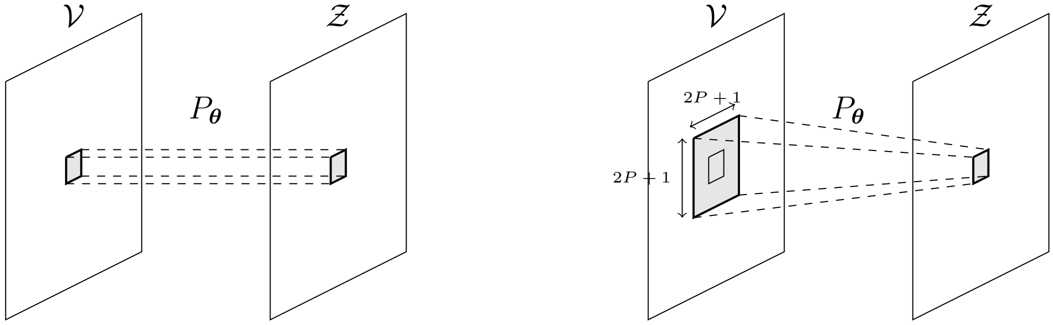


Figure 4.2: Pixel-level and local level prediction. On the left it is reported the predictive model working at single pixel-level, while on the right the predictive model working at local level. Working at local level allows to exploit the spatial correlation on the raw ToF data to improve the final prediction. The generative model works always at single pixel-level.

This is the approach followed by most of the methods proposed in the literature. The main limitation is that it works independently for each pixel even if from geometrical considerations it is clear that there should be some spatial correlation between adjacent pixels in the camera sensor.

The second option is working at local level. In order to produce the output for pixel (u, v), the predictive model can look at a small neighbourhood of size $(2P+1) \times (2P+1)$ around the corresponding input pixel:

$$\mathcal{Z}_{uv} = P_{\boldsymbol{\theta}}\Big(\big\{\mathcal{V}_{u'v'} \big| u' = u + k;\ v' = v + j;\ k, j = -P, ..., +P\big\}\Big) \tag{4.10}$$

This second strategy can improve the accuracy of the final prediction taking advantage of the spatial correlation on the input data. In principle, multiple interfering returns are generated by the reflections of the emitted light on the entire scene and thus can reach the camera sensor in any point, but due to the exponential decay of scattering events and the quadratic attenuation with distance, they are mainly relevant on a local neighbourhood of each pixel. This observation suggests that it is very likely that pixels inside a local neighbourhood share similar MPI effects and thus show correlation. Increasing the receptive field P allows to capture higher order reflections and exploit better the spatial correlation on data, but it also increases the computational cost for the predictive model. Also in this case, there is a trade-off between computational cost and performance. Moreover, the spatial correlation can be also exploited to learn typical patterns in the input data associated to recurrent scene geometries in the real-world, such as edges, corners, flat surfaces, spheres and so on.

In figure 4.2 it is illustrated the working principle behind the single pixel-level and local level prediction. Note that in the proposed method the generative model is designed to work always at pixel-level since there is a one-to-one relationship between the output backscattering vector and its compressed representation for each pixel. All the advantages coming from the spatial correlation are exploited by the predictive model.

4.4 Temporal Correlation

One advantage of ToF technology with respect to other range imaging techniques is the possibility to capture data in real-time, therefore the proposed predictive model can be further extended to account also for the temporal correlation. The predictive model can be designed in such a way that the latent variable is predicted looking at the evolution of the acquired raw ToF data through time, reflecting the temporal evolution of the scene. In particular, since the system must be causal, the current prediction at time $t > 0$ will be a function of the set of recorded tensors $\{\mathcal{V}^\tau\}$ at time $\tau = t, t-1, \ldots, t-T$ using some recurrent neural network techniques.

Temporal correlation between subsequent frames is naturally present, since typically a camera acquires static scenes or partial dynamic scenes where a large portion of image does not change. Differently, if it is the camera that moves we get a completely dynamic scene, but still natural camera movements ensure temporal correlation between a small set of subsequent frames, *i.e.* small T. Random camera movements destroy completely the temporal correlation, but very likely we are not interested to reconstruct the transient scene in these cases.

It is a capital mistake to theorize before one has data.
Sherlock Holmes (Arthur Conan Doyle)

5
ToF Datasets

The incoming chapter discusses different ToF datasets which have been employed for training and testing the proposed approach. Appendix A reports the procedures used for the generation and the processing of ToF data. Note that all algorithms must be considered as proofs of concept, and that the actual implementation must take into account also the computational efficiency and other small details that have been omitted in the appendix for brevity.

For the supervised optimization of the proposed approach we need a training set containing raw ToF data together with the corresponding ground truth transient scenes. Moreover, since in this work the estimated backscattering vector is used mainly for MPI compensation and depth correction, the training set must contain also the associated depth ground truth, *i.e.* $T = \{(\mathcal{V}_i, \mathcal{X}_i, \mathcal{D}_i)\}_{i=0}^{M-1}$. Note the information about the depth map is implicitly present in the transient scene, but it is provided separately for the sake of convenience. From geometrical considerations it is clear that the direct component is always associated to the shortest path, therefore the real depth corresponds to the index of the first non-zero element in the backscattering vector. The depth can be easily extracted from the transient scene using the algorithm A.1, where factor $c > 0$ has been introduced to account for numerical issues and noise. We use $c = 0.01$.

The acquisition of a real dataset with transient ground truth is a very complex and expensive task and since there are not publicly available datasets suitable for our purposes, we were forced to run the optimization of the proposed approach in a simulated environment. Clearly, simulated data differ from the real one and introduce approximations in the final prediction. One can think to many different generation procedures, from the simplest to the more advanced ones which model a large number of physical effects. As usual, there is a trade-off between complexity and accuracy. In chapter 6 we will propose three methods for backscattering vector estimation, namely BVE_1SRP, BVE_2SRP and BVE_2SRL. Each method relies on different assumptions and therefore it requires a specialized training set.

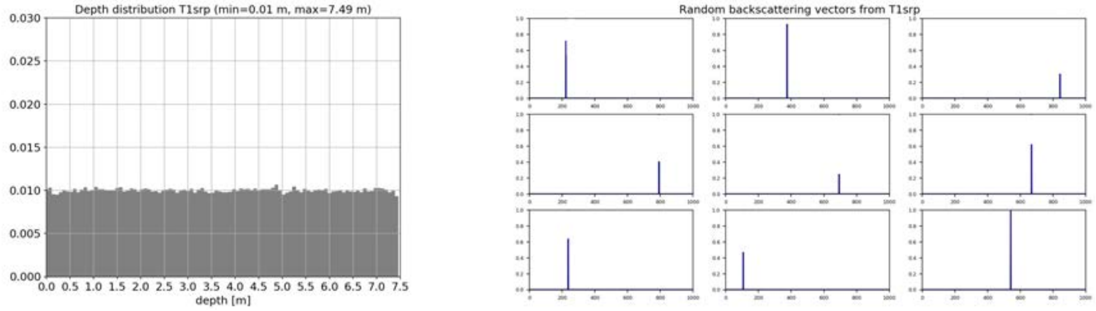


Figure 5.1: Qualitative analysis of simulated dataset T^{train}_{ISRP}. The plot on the left reports the depth distribution, while on the right some examples of single-peak backscattering vector are shown.

The next three sections describe the synthetic datasets used for the training of our networks, as well as their generation procedures.

Conversely, the final test is instead carried on real ToF data for which we do not have the transient scenes but only the ground truth depth maps, *i.e.* $S = \{(\mathcal{V}_i, \mathcal{D}_i)\}_{i=0}^{M-1}$. The performance are evaluated on real-world scenes looking at the MPI correction capabilities based on the estimated backscattering vector. In the last section three real-world datasets are introduced. These three datasets are the ones we will use for the final test of the MPI correction capabilities of the proposed methods.

5.1 Simulated Training Dataset T_{ISRP}

Since the first developed method BVE_ISRP does not account for the MPI effect, we need to create a dataset $T_{\text{ISRP}} = \{(\mathcal{V}_i, \mathcal{X}_i, \mathcal{D}_i)\}_{i=0}^{M-1}$ where each backscattering vector is formed by a single peak. Moreover, BVE_ISRP works independently for each pixel so there is no need to have spatial correlation on the simulated data. The algorithm A.3 generates a completely random dataset, where each pixel has a depth value sampled uniformly in a given depth range and a backscattering vector formed by a single non-zero element, *i.e.* single specular reflection. We chose to generate a uniform depth distribution because this corresponds to not assume any prior knowledge about the scene geometry, therefore the network is forced to learn the real relationship between input and output instead of some characteristic geometric structures contained in the training data. The uniform distribution is also the one which provides higher entropy, meaning that it maximizes the amount of information supplied by each training sample. The acquired raw data are then simulated according to the measurement model (2.41). Note that the resulting dataset does not contain any source of noise since it will be added randomly at each optimization step as form of regularization.

Using the discussed procedure we generate a training set T^{train}_{ISRP} for the optimization of our network, and a validation set T^{valid}_{ISRP} for hyperparameters tuning. Both two datasets have

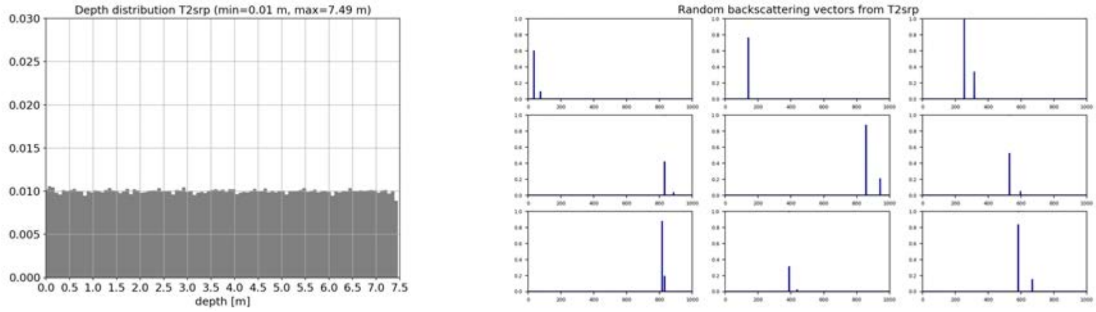


Figure 5.2: Qualitative analysis of simulated dataset T_{2SRP}^{train}. The plot on the left reports the depth distribution, while on the right some examples of two-peaks backscattering vector are shown.

spatial resolution $W = H = 1$ and backscattering vectors discretized into $N = 1000$ steps and with a maximum amplitude of one. The training set is formed by $M_{train} = 211\,200$ samples, while the validation test by $M_{train} = 2064$ samples. Figure 5.1 reports a qualitative analysis of the simulated data.

$$T_{\text{1SRP}}^{train} = \textsc{SimulateT1srp}(211\,200, 1, 1, 1000, 7.49\text{mm}, 1) \qquad (5.1)$$

$$T_{\text{1SRP}}^{valid} = \textsc{SimulateT1srp}(2064, 1, 1, 1000, 7.49\text{mm}, 1) \qquad (5.2)$$

5.2 Simulated Training Dataset T_{2SRP}

The second method BVE_2SRP considers the simple two specular reflection case, therefore we need to simulate a dataset $T_{\text{2SRP}} = \{(\mathcal{V}_i, \mathcal{X}_i, \mathcal{D}_i)\}_{i=0}^{M-1}$ with backscattering vectors formed by two non-zero elements. Also in this case spatial correlation on data is not required. Similarly to before, algorithm A.4 produces a completely random dataset. The second peak is generated as a function of the first one: the position is obtained adding a random offset to the position of the first peak, while the amplitude multiplying the amplitude of the first peak by a random scale factor. This procedure aims to reproduce the strong correlation naturally present between the two main peaks in a real backscattering vector. Both the offset and the scale factor are randomly and uniformly distributed. The generation process accounts also for the possibility to have MPI-free pixels, in particular the global component is added only with a certain probability P_{MPI}.

As before, we generate a training set T_{2SRP}^{train} and a validation set T_{1SRP}^{valid} with the same generation parameters. For the second peak we use a maximum offset of $O = 100$ indices, corresponding to a maximum depth offset of $\Delta d = 74.9\text{cm}$, a maximum scale factor $S = 0.4$ and a multi-path probability $P_{MPI} \sim 0.5$. Also in this case, the choice $P_{MPI} \sim 0.5$ has been made to avoid that the network learns some priors about the distribution of MPI-affected pixels which may vary a lot depending on the considered scenes and forces it to detect the

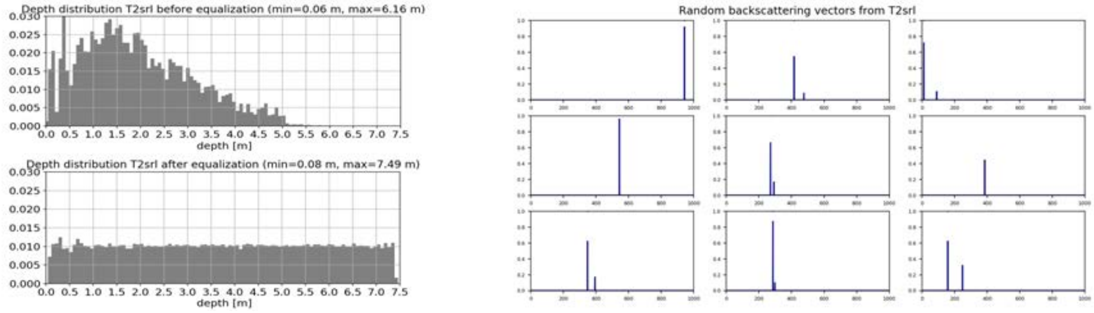


Figure 5.3: Qualitative analysis of simulated dataset $T_{2\text{SRL}}^{train}$. The two plots on the left report the depth distribution before and after the equalization procedure, while on the right some examples of two-peaks backscattering vectors are shown.

presence of multiple returns only looking at the input data. Figure 5.2 reports a qualitative analysis of the simulated data.

$$T_{2\text{SRP}}^{train} = \textsc{SimulateT2srp}(211\,200, 1, 1, 1000, 100, 0.4, 7.49\text{mm}, 1, 0.5) \tag{5.3}$$

$$T_{2\text{SRP}}^{valid} = \textsc{SimulateT2srp}(2064, 1, 1, 1000, 100, 0.4, 7.49\text{mm}, 1, 0.5) \tag{5.4}$$

5.3 Simulated Training Dataset $T_{3\text{SRL}}$

The last developed method BVE_2SRL relies on the spatial correlation to improve the final prediction so the training dataset must account for it. Now we have to simulate a dataset of patches, which exhibits spatial correlation inside each patch and formed by backscattering vectors with two peaks. A completely random generation process is no more suitable. In order to generate spatial correlated data the idea is to start from a proper depth map, computing the direct component position in a deterministic manner and finally using a random procedure similar to the one used before for the global component generation. The raw data are always simulated through the measurement model (2.41). Algorithm A.6 implements exactly this idea. As source of spatial correlation we use random depth map patches extracted from the FLAT dataset released by Guo *et al.* [18]. The problem is that the FLAT dataset provides a depth distribution different from the uniform one that we want to use for the training of our network. This can be solved applying a depth equalization procedure, detailed in algorithm A.5. Depth equalization is designed to add a constant offset to each patch in order to obtain a final depth distribution as uniform as possible. Note that the offset is constant inside each patch but differs between different patches. This can be seen as a form of data augmentation, since adding a constant positive offset to a depth map is equivalent to move the camera far away from the scene. In principle, since ToF cameras acquire radial depth maps, adding a constant offset slightly modifies the scene geometry, but this is not a big problem since we are not interested to the scene geometry but only to the spatial correla-

Dataset	M	$W \times H$	N	O	S	d_{res}	x_{max}	P_{MPI}	No. spec. reflections	Spatial corr.
T^{train}_{1SRP}	211200	1×1	1000			7.49mm	1		1	No
T^{valid}_{1SRP}	2064	1×1	1000			7.49mm	1		1	No
T^{train}_{2SRP}	211200	1×1	1000	100	0.4	7.49mm	1	0.5	2	No
T^{valid}_{2SRP}	2064	1×1	1000	100	0.4	7.49mm	1	0.5	2	No
T^{train}_{2SRL}	211200	3×3	1000	100	0.4	7.49mm	1	0.5	2	Yes
T^{valid}_{2SRL}	2064	3×3	1000	100	0.4	7.49mm	1	0.5	2	Yes

Table 5.1: Summary of the generation parameters for the simulated training and validation datasets.

tion on data which is preserved. A more sophisticated depth equalization algorithm can be developed in order to add the constant offset in the z-depth space and then transform back the z-depth into a radial depth value.

In order to simulate the required training and validation tests, we extract from FLAT two sets of depth map patches $\{\mathcal{D}_i^{train}\}_{i=0}^{M^{train}-1}$ and $\{\mathcal{D}_i^{valid}\}_{i=0}^{M^{valid}-1}$ with spatial resolution $W = H = 3$, then we run the depth equalization procedure and finally we use them as source of spatial correlation. The spatial resolution 3×3 has been chosen because BVE_2SRL , which uses *valid* convolutions, has exactly this receptive field and therefore the number of output valid pixels is the same of before. Figure 5.3 reports a qualitative analysis of the simulated datasets, showing the depth distribution before and after the equalization procedure and some two-peaks backscattering vectors chosen at random from T^{train}_{2SRL}.

$$\{\mathcal{D}_i^{train,eq}\}_{i=0}^{M^{train}-1} = \textsc{DepthEqualization}(\{\mathcal{D}_i^{train}\}_{i=0}^{M^{train}-1}, 0, 7.49\text{m}) \tag{5.5}$$

$$\{\mathcal{D}_i^{valid,eq}\}_{i=0}^{M^{valid}-1} = \textsc{DepthEqualization}(\mathcal{D}_i^{valid}\}_{i=0}^{M^{valid}-1}, 0, 7.49\text{m}) \tag{5.6}$$

$$T^{train}_{\text{2SRL}} = \textsc{SimulateT2srl}(\{\mathcal{D}_i^{train,eq}\}_{i=0}^{M^{train}-1}, 1000, 100, 0.4, 7.49\text{mm}, 1, 0.5) \tag{5.7}$$

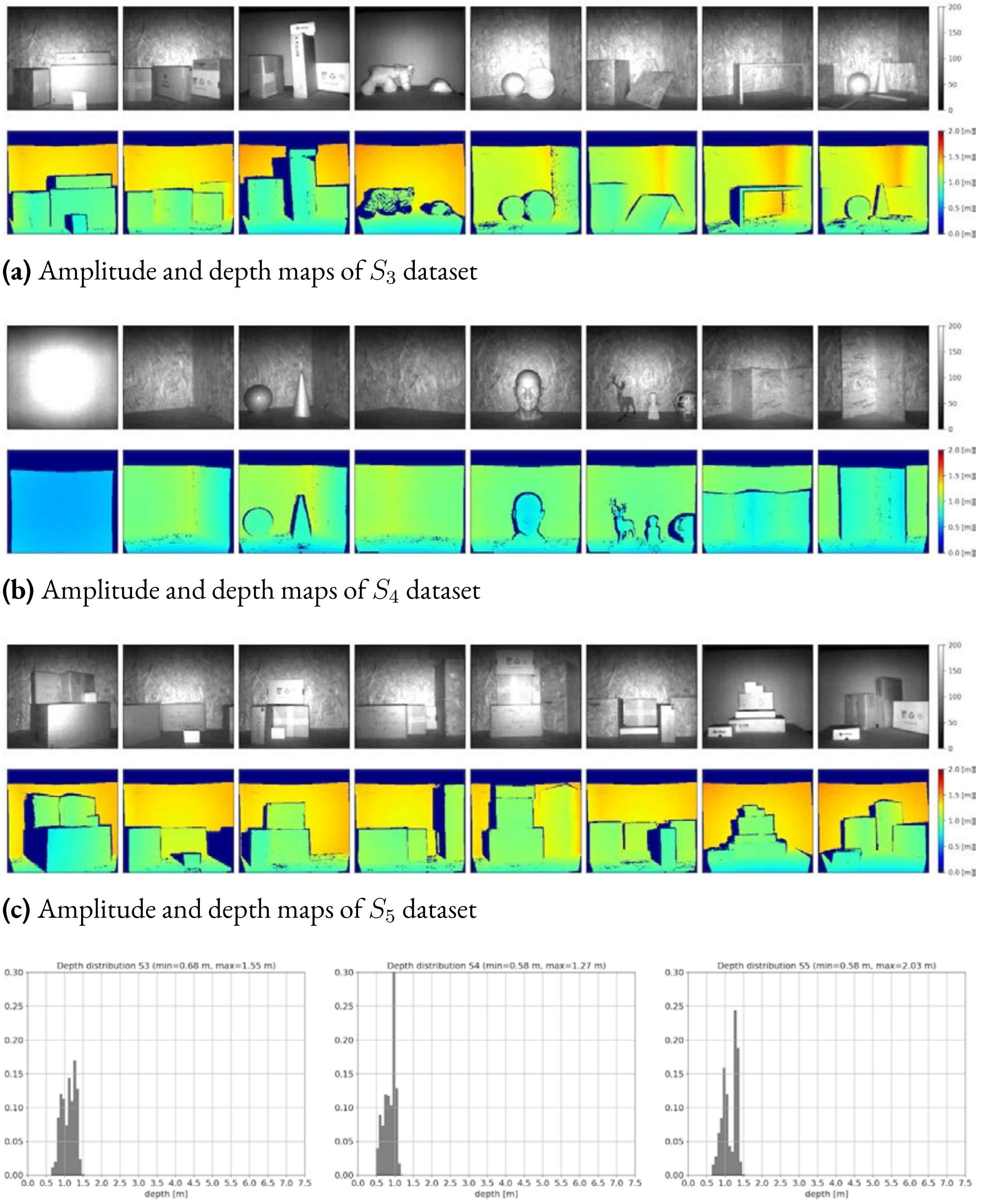
$$T^{valid}_{\text{2SRL}} = \textsc{SimulateT2srl}(\{\mathcal{D}_i^{valid,eq}\}_{i=0}^{M^{valid}-1}, 1000, 100, 0.4, 7.49\text{mm}, 1, 0.5) \tag{5.8}$$

5.4 Real-World Testing Datasets S_3, S_4 and S_5

In order to test the performance achieved by our methods in a real-world scenario we use three real ToF datasets provided together with the research papers of Agresti *et al.* . In particular dataset S_4 comes from [19] while datasets S_3 and S_5 (*box* dataset) have been acquired in [20]. In [20] the authors report a comparison between different state-of-the-art approaches for MPI compensation on these three real datasets. We will take their work as baseline and we will perform a consistent comparison, studying how our method behaves with respect to the others techniques. The three datasets have been all acquired in a laboratory without external illumination using the SoftKinetic ToF camera DS541 at multiple modulation frequencies. For each scene they provide unwrapped phase, amplitude and intensity, as well as depth ground truth. Before the acquisitions, the camera has been calibrated in order to remove the wiggling error. Depth ground truth acquisition is a very complex and time consuming task, therefore its availability makes these datasets particularly useful in the study and development of ToF data denoising methods. The depth ground truth has been acquired using an active stereo system. First, a light projector illuminates the scene with a series of phase shifted patterns while the stereo system is recording. Then, there is the depth map computation and the triangulation process that leads to an accurate depth estimation. Finally, the ground truth depth map is projected on the ToF camera field of view. In table 5.2 the main properties of the three datasets are summarized, while figure 5.4 shows qualitatively them. All the real scenes are in the depth range between 58 to 203 cm with the depth distribution in figure 5.4.d.

Dataset	Type	Depth GT	Trans. GT	No. Scenes	Spatial Res.	Modulation frequencies
S_3	Real	yes	no	8	320×239	10, 20, 30, 40, 50 and 60 MHz
S_4	Real	yes	no	8	320×239	20, 50 and 60 MHz
S_5 (*box*)	Real	yes	no	8	320×239	10, 20, 30, 40, 50 and 60 MHz

Table 5.2: Properties of the real-world testing datasets S_3, S_4 and S_5.



(a) Amplitude and depth maps of S_3 dataset

(b) Amplitude and depth maps of S_4 dataset

(c) Amplitude and depth maps of S_5 dataset

(d) Depth distribution of datasets S_3, S_4 and S_5

Figure 5.4: Qualitative analysis of real datasets S_3, S_4 and S_5. Figures (a),(b) and (c) show amplitude and depth map of each scene in the dataset. Figure (d) reports the depth distribution on the three datasets.

Develop a passion for learning. If you do, you will never cease to grow.

Anthony J. D'Angelo

6 Developing Pipeline

In this chapter we are going to discuss the entire pipeline that has lead to the development of the proposed approach for backscattering vector estimation, as well as all the related implementation details. The roadmap is organized following a bottom-up approach, starting from the simplest case and adding more and more details as the development proceeds upward. This paradigm allows to break down the main complex problem into many small and more treatable sub-problems. Each single sub-problem is going to be analysed and understood, in order to gain a complete knowledge regarding the behaviour of the proposed algorithm. In the following chapter we will focus on a restricted case, in particular we will assume that the MPI effect is generated only by specular reflections without accounting for possible diffuse reflections. Nevertheless, in principle our method can work also with diffuse

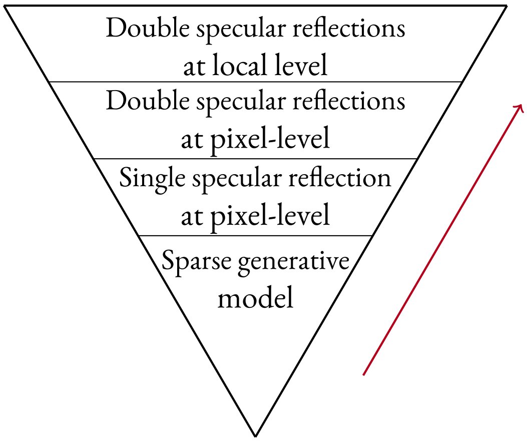

Double specular reflections at local level
Double specular reflections at pixel-level
Single specular reflection at pixel-level
Sparse generative model

Figure 6.1: Illustration of the bottom-up paradigm followed in the developing pipeline. We start from the simplest case and add more and more details as the development proceeds upward.

Parameter	Value
No. modulation frequencies	$M = 3$
Modulation frequencies	$f_0 = 20\text{MHz}$, $f_1 = 50\text{MHz}$, $f_2 = 60\text{ MHz}$
Minimum depth value	$d_{min} = 0\text{m}$
Minimum ToF value	$t_{min} = 0\text{sec}$
Maximum depth value	$d_{max} = c/2f_0 = 7.49\text{m}$ (unambiguity range at f_0)
Maximum ToF value	$t_{max} = 1/f_0 = 5 \cdot 10^{-8}\text{sec}$
No. discretization steps	$N = 1000$
Depth resolution	$d_{res} = (d_{max} - d_{min})/N = 7.49\text{mm}$
ToF resolution	$t_{res} = (t_{max} - t_{min})/N = 5 \cdot 10^{-11}\text{ sec}$

Table 6.1: Summary of the main design parameters which define the our reference setup.

reflections, but its extension to the more general case requires further effort and is going to be the objective of a future work.

We will start introducing in section 6.1 the generative model we are going to adopt in the specular reflections case. In section 6.2 we will discuss the baseline implementation of our method considering only a single reflection and working at pixel-level. The objective is to perform a feasibility study and to investigate possible issues that may arise in the subsequent steps. Then, in section 6.3, we will keep working at single pixel-level but we will account for a double reflections MPI effect. Even if also in this case the prior assumption is very strong, from the experimental results we will observe how the proposed method is able to provide good MPI correction capabilities in a real-world scenario. Finally, in section 6.4 we will investigate the benefits of exploiting also the spatial correlation focusing always in the double reflections scenario. Looking at the experimental results, we will see that the spatial correlation can help filtering out the measurement noise, improving the final prediction and outperforming other state-of-art techniques in the MPI removal task.

Before starting the discussion we introduce the main design parameters which define our reference scenario. We assume to have available a ToF camera that captures raw data at $M = 3$ modulation frequencies. For simplicity, the three frequencies are supposed to be acquired simultaneously without accounting for possible motion artefacts that may arise. We use the set of modulation frequencies $f_0 = 20\text{MHz}$, $f_1 = 50\text{MHz}$ and $f_2 = 60\text{MHz}$. Our method is supposed to work in the depth range corresponding to the unambiguous range of the lower modulation frequency, that spans between 0 and 7.49 meters. We discretize the backscattering vector into $N = 1000$ steps, therefore the depth resolution is about 7.49 millimetres while the time-of-flight resolution is 5×10^{-11} seconds. Table 6.1 provides a complete summary summary of the main design parameters which define the our reference setup.

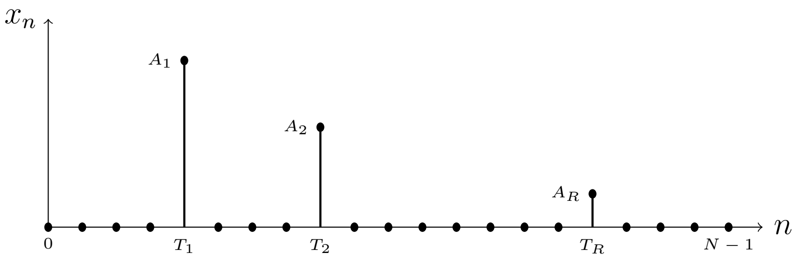


Figure 6.2: Graphical illustration of the output backscattering vector generated using the sparse generative model. Each pair (A_r, T_r) in the latent variable $\boldsymbol{z} = \left[A_1, T_1, ..., A_R, T_R\right]^T$ represents amplitude and time position of the r^{th} specular reflection.

6.1 Sparse Generative Model

As already stressed, in this work we assume the MPI phenomenon is generated by only specular reflections. This kind of approximation is the one adopted by many methods in the literature, see table 3.1, and the most treatable from a mathematical point of view. In particular, the backscattering domain is the signal space formed by at most $R \geq 1$ sparse reflections, that is:

$$\mathcal{D}_{\boldsymbol{x}}^{R} = \left\{ \boldsymbol{x} \in \mathbb{R}^N \,\middle|\, \|\boldsymbol{x}\|_0 \leq R \right\} \quad \subseteq \mathcal{D}_{\boldsymbol{x}}^{*} \tag{6.1}$$

where the $L0$ norm is defined as the total number of non-zero elements. For this very specific case it is not necessary to employ a complex generative neural network in order to develop a suitable *generative model*, because it can be fully descried by the sum of $R \geq 1$ Kronecker delta functions. More in details, given a latent variable in the form:

$$\boldsymbol{z} = \left[A_1, T_1, A_2, T_2, ..., A_R, T_R\right]^T \in \mathbb{R}^{2R} \tag{6.2}$$

the sparse generative model is defined as:

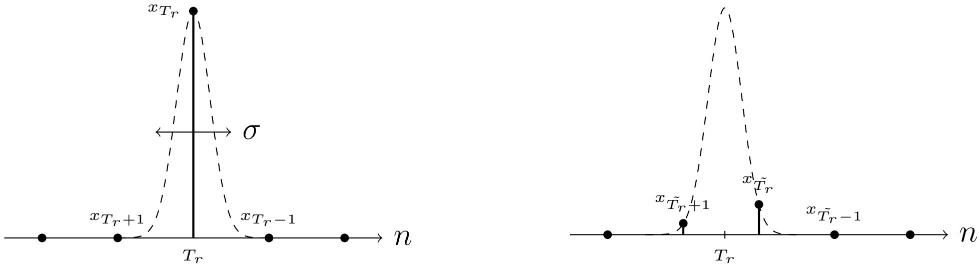
$$\boldsymbol{x} = G^R(\boldsymbol{z}) = \left[x_0, ..., x_{N-1}\right]^T \in \mathcal{D}_{\boldsymbol{x}}^{R} \qquad \text{with } x_n = \sum_{r=1}^{R} A_r\, \delta(n - T_r) \tag{6.3}$$

where the Kronecker delta is:

$$\delta(n-j) = \begin{cases} 1 & \text{if } n = j \\ 0 & \text{if } n \neq j \end{cases} \tag{6.4}$$

In this context the latent variable assumes an explicit meaning, each pair (A_r, T_r) represents amplitude and time position of the r^{th} specular reflection. Figure 6.2 gives a graphical representation of the meaning of latent variables.

The first problem with this formulation of the sparse generative model is that it is not differentiable due to the impulsive nature of the Kronecker delta function. In the continuous



(a) Small standard deviation

(b) Discretization problem

Figure 6.3: Graphical illustration of the approximation of the Kronecker delta trough a sampled gaussian function.

case, the equivalent Dirac function can be seen as a gaussian function with a standard deviation that tends to zero. The idea is to make the sparse generative model differentiable by substituting the Kronecker delta with a sampled gaussian function with very small standard deviation, that is:

$$x_n = \sum_{r=1}^{R} A_r \, e^{-\frac{(n-T_r)^2}{2\sigma^2}} \qquad , \sigma << 1 \tag{6.5}$$

Using this approximation it is possible to compute explicitly the partial derivatives of the output backscattering vector with respect to the input latent variables:

$$\frac{\partial x_n}{\partial A_r} = \frac{\partial}{\partial A_r}\left[\sum_{j=1}^{R} A_j \, e^{-\frac{(n-T_j)^2}{2\sigma^2}}\right] = e^{-\frac{(n-T_r)^2}{2\sigma^2}} \qquad r = 1, ..., R \tag{6.6}$$

$$\frac{\partial x_n}{\partial T_r} = \frac{\partial}{\partial T_r}\left[\sum_{j=1}^{R} A_j \, e^{-\frac{(n-T_j)^2}{2\sigma^2}}\right] = A_r \frac{n-T_r}{\sigma^2} e^{-\frac{(n-T_r)^2}{2\sigma^2}} \qquad r = 1, ..., R \tag{6.7}$$

The value for the small standard deviation $\sigma > 0$ is chosen in order to have a good approximation of the Kronecker delta after the sampling operation, as illustrated in figure 6.3a. In our implementation we use $\sigma = 0.2$.

The second problem is related to the discrete nature of the output backscattering vector. In particular, the time position of each sparse reflection must be an integer number, while typically the latent variable is formed by all real components. Note that approximating the Kronecker delta with a gaussian function allows to have also real time positions, but the sampling operation leads to a completely wrong output, see figure 6.3b. The simplest approach is to round each time position $T_r \in \mathbb{R}$ to the nearest integer number $\tilde{T_r} = Q(T_r) \in \mathbb{N}$ before applying the model in equation (6.5):

$$x_n = \sum_{r=1}^{R} A_r \, e^{-\frac{(n-\tilde{T_r})^2}{2\sigma^2}} \tag{6.8}$$

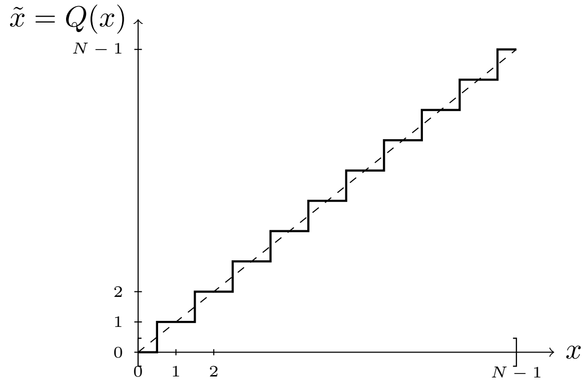


Figure 6.4: Mid-tread quantizer with unitary quantization step used to round the time position to the nearest integer value.

The rounding is achieved through a uniform mid-tread quantizer with unitary quantization step like the one depicted in figure 6.4:

$$\tilde{T} = Q(T) = \left\lfloor T + \frac{1}{2} \right\rfloor \in \mathbb{N} \qquad T \in [0, N-1] \subset \mathbb{R} \tag{6.9}$$

Unfortunately the quantization is not a differentiable operation and therefore the whole generative model would become non differentiable. Here we can apply a trick that is not perfectly mathematically correct but it is stable from a numerical point of view and makes the model differentiable. During the computation of the partial derivatives of the output backscattering vector with respect to the time position of each sparse reflection, we can ignore the quantizer, approximating the non-differentiable quantization rule with an identity function. The final expressions for the partial derivatives of the sparse generative model becomes:

$$\frac{\partial x_n}{\partial A_r} = e^{-\frac{(n-\tilde{T}_r)^2}{2\sigma^2}} \qquad r = 1, ..., R \tag{6.10}$$

$$\frac{\partial x_n}{\partial T_r} = \frac{\partial x_n}{\partial \tilde{T}_r} \underbrace{\frac{\partial Q(T_r)}{\partial T_r}}_{\approx 1} \approx A_r \frac{n - \tilde{T}_r}{\sigma^2} e^{-\frac{(n-\tilde{T}_r)^2}{2\sigma^2}} \qquad r = 1, ..., R \tag{6.11}$$

6.2 Single Specular Reflection at Pixel-level

The first step is the development of the proposed approach in the simplest case, which corresponds to the ideal case where the backscattering vector is generated by a single specular reflection. In other words, we do not consider the MPI effect. Clearly, this is a strong suboptimal assumption in a real-world scenario, therefore we do not expect that the resulting algorithm will be able to correct errors caused by multiple returns. The objective is to perform a feasibility study to identify possible issues that may arise also in the subsequent steps.

In this context we adopt the sparse generative model discussed in section 6.1, in particular we consider the single specular reflection case defined as follows:

$$\begin{aligned} G^1 : \qquad\qquad \mathbb{R}^2 \quad &\rightarrow \quad \mathcal{D}^1_{\boldsymbol{x}} \subseteq \mathbb{R}^N \\ \boldsymbol{z} = \big[A_1, T_1\big]^T \quad &\rightarrow \quad \boldsymbol{x} = G^1(\boldsymbol{z}) = \big[x_0, ..., x_{N-1}\big]^T \end{aligned} \tag{6.12}$$

with:

$$x_n = A_1 \, e^{-\frac{(n-\tilde{T}_1)^2}{2\sigma^2}} \tag{6.13}$$

Moreover, for now, we work at single pixel-level. This means that the predicted backscattering vector for pixel (u, v) is a function only of the corresponding pixel in the input raw ToF data:

$$\mathcal{X}_{uv} = G^1\big(\mathcal{Z}_{uv}\big) = G^1\big(P_{\boldsymbol{\theta}}\big(\mathcal{V}_{uv}\big)\big) \tag{6.14}$$

We will refer to this first developed method with the name BVE_1SRP, an abbreviation which stands for "Backscattering Vector Estimation in the Single Specular Reflection case working at Pixel-level". The goal is to achieve a performance comparable with the standard ToF technique for depth estimation. Since the standard depth estimation uses a single modulation frequency, in principle we expect to perform slightly better. Using data acquired at multiple frequencies, the network should be able to filter out the noise in a more reliable manner.

In the following sections we are going to discuss the implementation details of the predictive neural network, such as its architecture, the loss functions used for the measurement and the reconstruction errors and the training strategy adopted. Moreover, we will present a problem which arises with the reconstruction error in our particular case, together with an analysis of several loss functions tested to solve the issue. The final experimental results on the real-world case are going to be provided in section 7.1.

6.2.1 Predictive Neural Network Architecture

For the implementation of the predictive model we propose a neural network that takes in input the raw ToF data and predicts the latent variable associated to the corresponding backscattering vector. This process is repeated for each pixel of the camera sensor in order to obtain in output the complete transient scene. The predictive model works on tensors with fixed spatial resolution $W \times H$, and variable number of channels. The input tensor $\mathcal{V}$ has $2M = 6$ channels, corresponding to the real and the imaginary parts of the complex phasors acquired at the three modulation frequencies, while the output tensor $\mathcal{Z}$ has $L = 2$ channels, corresponding to amplitude and position of the single specular reflection we aim to predict. Since in principle all the pixels in the image are equivalent, the inversion of the underdetermined system (4.1) has to be performed using the same strategy for each pixels. Moreover, BVE_1SRP works at single pixel-level and so there is a one-to-one correspondence

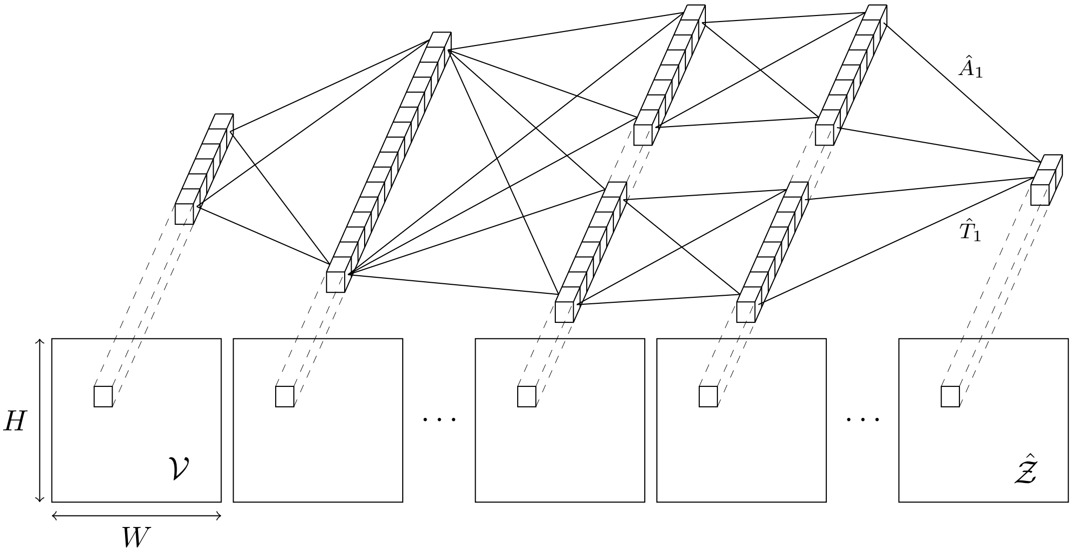


Figure 6.5: Graphical illustration of the predictive network architecture of BVE_ISRP.

between input and output pixels. The idea is to use a neural network composed by stacking of a set of layers one over the other. The layers are designed to implement a fully-connected network along the channel dimension which works in the same manner for all the pixels. Each layer extracts a variable number of features from the previous one, where each feature is obtained applying a non-linear activation function to a linear combination of the previous layer's features with learnable weights. This network can be seen as a convolutional network with kernels of size 1×1 along the spatial dimensions and a variable number of filters for each layer. The architecture of the proposed network is depicted in figure 6.5, while table 6.2 reports the details of each layer. The network is formed by a first block of three convolutional layers, all with 64 filters, which is supposed to denoise the input ToF data and to extract relevant features. Then, it is split into two parallel specialized branches. One branch predicts the amplitude $\hat{A}_1$, while the other estimates the position $\hat{T}_1$ of the non-zero element in the backscattering vector. Both are formed by three convolutional layers with decreasing number of filters, respectively 32, 16 and 1. Finally, the two branches are concatenated to produce the output latent variable $\hat{\boldsymbol{z}} = \left[\hat{A}_1, \hat{T}_1\right]^T$. We use the ReLU activation function for all hidden layers and the sigmoid for the output. The output sigmoid is rescaled in order to cover the whole latent variable range, that is $[0; x_{max})$ for $\hat{A}_1$, and $[t_{min}, t_{max})$ for $\hat{T}_1$. The total number of learnable parameters of the proposed predictive model is 14 018.

6.2.2 Loss Function Analysis

The choice of a proper loss function is a crucial point in the machine learning pipeline. It has the fundamental role of driving the algorithm towards the optimal solution. Mathemat-

Layer	Kernel size	Input dimension	Output dimension	Activation
Conv $c_1(\mathcal{V})$	$1 \times 1 \times 6 \times 64$	$W \times H \times 6$	$W \times H \times 64$	ReLU
Conv $c_2(c_1)$	$1 \times 1 \times 64 \times 64$	$W \times H \times 64$	$W \times H \times 64$	ReLU
Conv $c_3(c_2)$	$1 \times 1 \times 64 \times 64$	$W \times H \times 64$	$W \times H \times 64$	ReLU
Conv $a_1(c_3)$	$1 \times 1 \times 64 \times 32$	$W \times H \times 64$	$W \times H \times 32$	ReLU
Conv $a_2(a_1)$	$1 \times 1 \times 32 \times 16$	$W \times H \times 32$	$W \times H \times 16$	ReLU
Conv $a_3(a_2)$	$1 \times 1 \times 16 \times 1$	$W \times H \times 16$	$W \times H \times 1$	Sigmoid
Conv $t_1(c_3)$	$1 \times 1 \times 64 \times 32$	$W \times H \times 64$	$W \times H \times 32$	ReLU
Conv $t_2(t_1)$	$1 \times 1 \times 32 \times 16$	$W \times H \times 32$	$W \times H \times 16$	ReLU
Conv $t_3(t_2)$	$1 \times 1 \times 16 \times 1$	$W \times H \times 16$	$W \times H \times 1$	Sigmoid
Concat $z(a_3, t_3)$		$W \times H \times 1$, $W \times H \times 1$	$W \times H \times 2$	

Table 6.2: Hyperparameters used in the predictive network architecture of BVE_ISRP. For each layer, the table reports the kernel dimensions, the input and output dimensions and the activation function applied at its output.

ically, a loss function maps a certain event onto a real number, intuitively representing the *cost* associated with the event. In a supervised context, typically, the event is represented by the current prediction of a learning algorithm and the loss function measures how it differs from known ground truth. The choice of the loss function is strictly dependent on the task under investigation and unfortunately there is not an universal loss that works for all kind of data.

In section 4.2 we explained the general idea behind the proposed predictive model and we said that its loss function is formed by two main terms, the *measurement error* ℓ_m and the *reconstruction error* ℓ_r. In order to quantify how consistent is the final prediction with the recorded correlation measures, the measurement error compares the input phasor vector $\boldsymbol{v}$ with the one obtained applying the measurement model 2.38 to our predicted backscattering vector, *i.e.* $\hat{\boldsymbol{v}} = \Phi\,\hat{\boldsymbol{x}}$. Note that we are working on the real version of the input complex vector $\boldsymbol{v}$ obtained stacking the real part on top of its imaginary part. The goal is to keep this difference as small as possible, preferring small errors with respect to fewer very large errors. The best suitable loss function for this task is the *squared error loss*, also know as *L2 loss*:

$$\ell_m(\boldsymbol{v}, \hat{\boldsymbol{v}}) = \sum_{m=0}^{2M-1} \left(v_m - \hat{v}_m\right)^2 \tag{6.15}$$

In case of noisy measures this error is never going to zero, but it will converge to the power of the noise. Recall that higher modulation frequencies have finer depth resolution and higher noise resilience. Therefore it makes sense to give more relevance to higher frequencies in order to force the solution to be more related to those. Experimental results show that a reasonable way to fix the weight associated to each modulation frequency is to chose it inversely

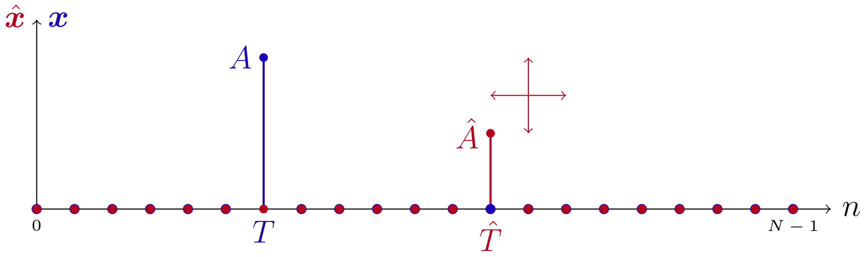


Figure 6.6: Graphical illustration of the considered optimization example. The blue signal represents the fixed ground truth backscattering, while the red one is our prediction that needs to be optimized.

proportional to the corresponding depth resolution, equations (2.2) and (2.23):

$$d_{res,m} = \frac{d_{max} - d_{min}}{N} = \frac{c}{2Nf_m} \qquad w_m = \frac{1/d_{res,m}}{\sum_{m'=0}^{2M-1} 1/d_{res,m'}} = \frac{f_m}{2(f_0 + f_1 + f_2)} \tag{6.16}$$

In this case we can use the *weighted squared error loss* for the measurement error:

$$\ell_m(\boldsymbol{v}, \hat{\boldsymbol{v}}) = \sum_{m=0}^{M-1} w_m \left(v_m - \hat{v}_m\right)^2 \tag{6.17}$$

More critical is the choice of the loss function for the reconstruction error. It should measure how well the predicted backscattering matches the ground truth, but the problem is that we are trying to compare two highly sparse vectors. A common problem with neural networks arises when they work on highly sparse data because they tend to stuck on bad local minima due to the fact that in most of the points the gradient of the loss function is going to be zero. In the following we will present an analysis on different formulations of the reconstruction error. For convenience, we will avoid the use of the subscript r, writing:

$$\ell_r(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \ell_s(\boldsymbol{x}, \hat{\boldsymbol{x}}) \tag{6.18}$$

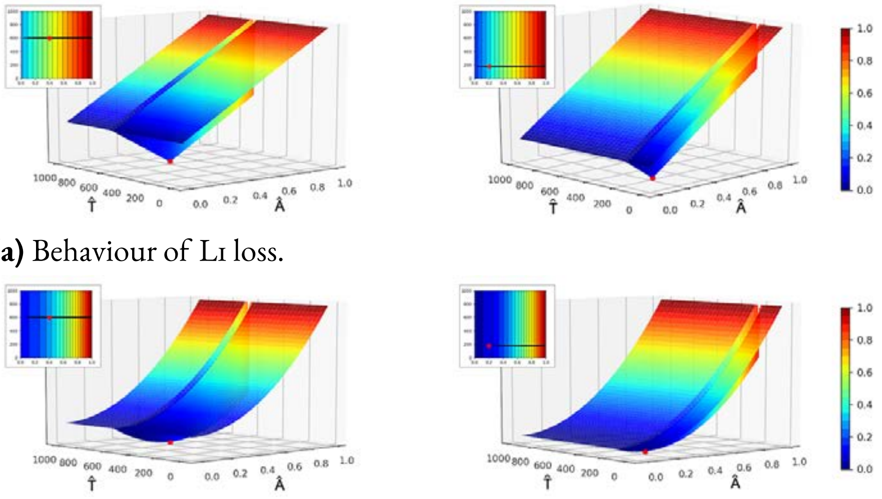
where s is a string indicating the type of reconstruction loss we are considering.

6.2.2-1 Absolute Error and Squared Error Loss Functions

The most common loss functions typically adopted for regression tasks are the *absolute error loss* (or *L1 loss*) and the *squared error loss* (or *L2 loss*):

$$\ell_{\text{L1}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \sum_{n=0}^{N-1} \left|x_n - \hat{x}_n\right| \tag{6.19}$$

$$\ell_{\text{L2}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \sum_{n=0}^{N-1} \left(x_n - \hat{x}_n\right)^2 \tag{6.20}$$



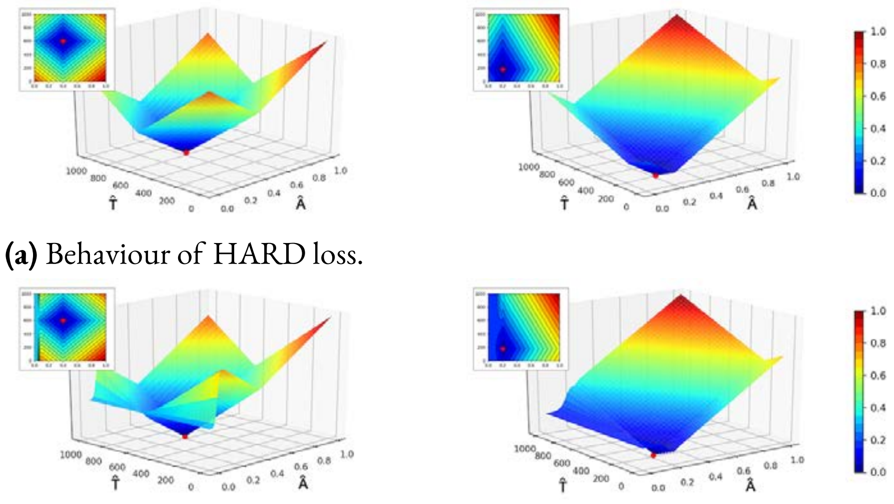
(a) Behaviour of L1 loss.

(b) Behaviour of L2 loss.

Figure 6.7: Behaviour of L1 and L2 loss functions. On the left the ground truth is fixed to $A = 0.4, T = 600$, while on the right it is fixed to $A = 0.2, T = 200$.

Unfortunately, in this setting the network learns to predict always an all-zeros backscattering vector because it is the most likely local minimum in the loss function. In order to understand the behaviour of the network in this particular situation let's consider the following optimization example.

Let $\boldsymbol{z} = [A, T]^T$ be the latent variable representing the ground truth backscattering vector $\boldsymbol{x} = G^1(\boldsymbol{z})$, where A is the amplitude and T the time position of the specular reflection. Equivalently, let $\hat{\boldsymbol{z}} = [\hat{A}, \hat{T}]^T$ and $\hat{\boldsymbol{x}} = G^1(\hat{\boldsymbol{z}})$ be the predicted backscattering vector. Let's assume we are optimizing the network and therefore we can vary amplitude and position of the specular reflection in our prediction, while the ground truth remains fixed. Figure 6.6 illustrates schematically the situation we are considering, where the blue signal represents the ground truth backscattering and the red one is our prediction that needs to be optimized. We can gradually change the parameters $\hat{A} \in [0, 1)$ and $\hat{T} \in [0, N-1)$, and plotting for each pair $(\hat{A}, \hat{T})$ the corresponding loss function value. In figure 6.7 we repeat this experiment for both the L1 and L2 loss functions. Note that in order to compare the behaviour of different losses which may have different dynamic ranges, we have plotted the values renormalized into the interval $[0, 1]$. Looking at the plots we observe that, as expected, both the L1 and L2 losses have their absolute minimum in the optimal solution $(\hat{A}, \hat{T}) = (A, T)$, but we can also get a clue on the reason behind the failure of the learning process. During the optimization phase, the searching of the minimum is performed using the gradient descent algorithm. It works starting from a random initialization and decreasing the loss function value moving the final prediction along the negative direction of the gradient. From the plots



(a) Behaviour of HARD loss.

(b) Behaviour of SOFT loss.

Figure 6.8: Behaviour of HARD and SOFT loss functions. On the left the ground truth is fixed to $A = 0.4, T = 600$, while on the right it is fixed to $A = 0.2, T = 200$.

it is clear that we have a meaningful negative gradient only in a small neighbourhood of the optimal solution. In most of the cases the it does not point to the optimal solution, but to the "bad" local minimum $\hat{A} = 0$. Another issue that arises in this situation is the gradient vanishing problem. Since the gradient is null along the T-direction, the application of the backpropagation algorithm leads to vanishing weights updates.

6.2.2-II Hardmax and Softmax Loss Functions

From these considerations follow that, in order to use the gradient descent algorithm, a desirable property of the loss function is that it should have a negative gradient pointing in direction of the optimal solution in most of the cases. It must be able to drive the optimization algorithm towards the optimal solution in a smooth way, avoiding discontinuity. In our case, this means we cannot use a loss function that compares the two sparse vectors pointwise, instead we have to design a suitable loss function that works on the entire sequences and compares some other higher level features. For instance, one solution can be, first extract amplitude and position of the non-zero element in the two vectors, and then compare these two features. The most immediate approach is to use the standard max and $argmax$ functions, but since they are not differentiable they cannot be used in the gradient descend algorithm:

$$\ell_{\text{HARD}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \lambda_A \left| \underset{n}{max}\, \hat{\boldsymbol{x}} - \underset{n}{max}\, \boldsymbol{x} \right| + \lambda_T \left| \underset{n}{argmax}\, \hat{\boldsymbol{x}} - \underset{n}{argmax}\, \boldsymbol{x} \right| \tag{6.21}$$

A possibility is to replace the non-differentiable operations with the corresponding $soft$ versions, that are:

$$\ell_{\text{SOFT}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \lambda_A \left| \underset{n}{soft\,max}\, \hat{\boldsymbol{x}} - \underset{n}{soft\,max}\, \boldsymbol{x} \right| + \lambda_T \left| \underset{n}{soft\,argmax}\, \hat{\boldsymbol{x}} - \underset{n}{soft\,argmax}\, \boldsymbol{x} \right| \tag{6.22}$$

The typical implementation of the differentiable $soft\,max$ function relies on the idea of converting the vector $\boldsymbol{x}$ into a Probability Mass Distribution (PMD) $p_{\boldsymbol{x}}$ using:

$$p_{\boldsymbol{x}}(n) = \frac{e^{\beta x_n}}{\sum_{n'=0}^{N-1} e^{\beta x_{n'}}} \tag{6.23}$$

and then computing its expectation with respect to the precomputed probability distribution:

$$\underset{n}{soft\,max}\, \boldsymbol{x} = \mathbb{E}_{p_{\boldsymbol{x}}}\big[\boldsymbol{x}\big] = \sum_{n=0}^{N-1} p_{\boldsymbol{x}}(n)\, x_n \tag{6.24}$$

Similarly, the $soft\,argmax$ is implemented using the previous probability distribution to compute the expectation of the linear indices vector $\boldsymbol{n} = [0, ..., N-1]^T$:

$$\underset{n}{soft\,argmax}\, \boldsymbol{x} = \mathbb{E}_{p_{\boldsymbol{x}}}\big[\boldsymbol{n}\big] = \sum_{n=0}^{N-1} p_{\boldsymbol{x}}(n)\, n \tag{6.25}$$

The factor $\beta \geq 1$ has been introduced to raise the maximum value and lower the others in order to make the functions more discriminative. Figure 6.8 shows the behaviour of the hard and soft loss functions repeating the optimization example of before. Even if in this case the two losses exhibit the desired property, they are not good cost functions which we can use for our optimization. The hard loss is not differentiable, while its soft version is numerically unstable. The exponential term makes the soft loss prone to under and overflows for small and large input values.

6.2.2-iii Cross-Correlation Based Loss Functions

Since we are trying to compare two signals that can be time displaced to each other, another idea is to construct a suitable loss function based on their *cross-correlation*. The cross-correlation is a measure of similarity between two signals. In particular, the normalized cross-correlation is a function r_k that for each time displacement $k \in [-N+1, N-1]$ measures how similar $\boldsymbol{x} \in \mathbb{R}^N$ and $\hat{\boldsymbol{x}} \in \mathbb{R}^N$ are. It is defined as:

$$r_k = \frac{\sum_{n=0}^{N-1} x_n \cdot \hat{x}_{n+k}}{\sum_{n=0}^{N-1} x_n^2} \tag{6.26}$$

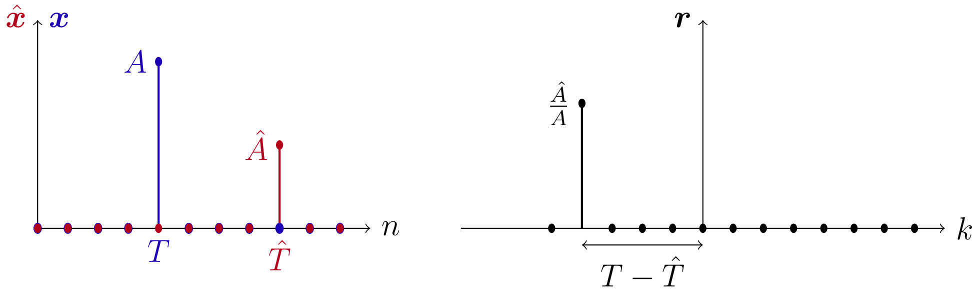


Figure 6.9: Normalized cross-correlation computation between the ground truth and the predicted backscattering vectors. On the left we have the two signals, while on the right the resulting normalized cross-correlation function.

where the normalization is performed with respect to the fixed ground truth backscattering vector. By construction, the cross-correlation between two backscattering vectors produced by a singular specular reflection, $\boldsymbol{x}, \hat{\boldsymbol{x}} \in \mathcal{D}^1_{\boldsymbol{x}}$, corresponds to a translated Kronecker delta:

$$r_k = R \cdot \delta(k - \Delta T) \tag{6.27}$$

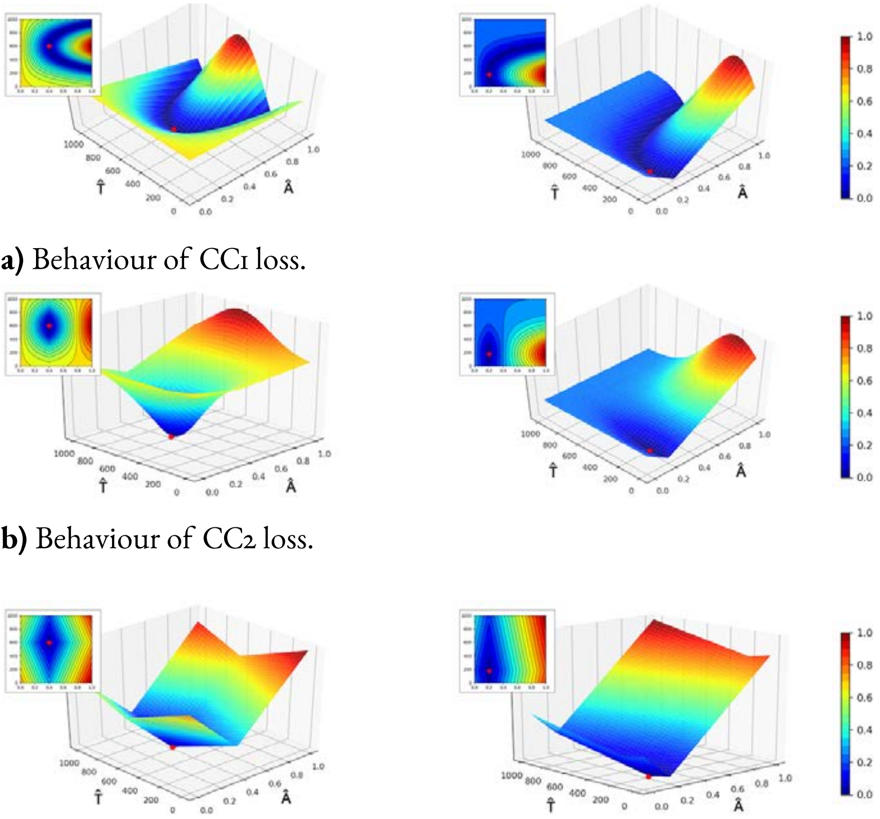
where $R = \hat{A}/A$ is the amplitude of the non-zero element while $\Delta T = T - \hat{T}$ is its position. Figure 6.9 gives a graphical illustration of the cross-correlation shape.

In the perfect matching case, *i.e.* $\boldsymbol{x} = \hat{\boldsymbol{x}}$, the correlation is formed by a single peak centred in the origin with unitary amplitude, *i.e.* $R_0 = 1$ and $\Delta T = 0$. This observation can be exploited to construct a proper loss function which makes the correlation tending to the perfect matching case. We need to associate a cost to each cross-correlation function that is minimized in the perfect matching case. There are different possibilities. One approach is to give a reward to each correlation peak proportional to its closeness to the origin and simultaneously constraining the amplitude of the main peak to be approximately unitary. The following are two proposals based on this idea. Both use a gaussian reward, the difference is in the amplitude constrain:

$$\ell_{\text{CC1}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \left| 1 - \sum_{k=-N+1}^{N-1} e^{-\frac{k^2}{2\sigma^2}} \cdot r_k \right| \tag{6.28}$$

$$\ell_{\text{CC2}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \sum_{k=-N+1}^{N-1} e^{-\frac{k^2}{2\sigma^2}} \cdot \left| 1 - r_k \right| \tag{6.29}$$

Looking at figures 6.10a-b we observe that the CC1 loss is not suitable for our purposes because it is minimized in the optimal matching case but also in many other points, introducing uncertainty about the optimal solution. Differently, the CC2 loss exhibits the desired behaviour having a gradient that points always towards the optimal solution.



(a) Behaviour of CC1 loss.

(b) Behaviour of CC2 loss.

(c) Behaviour of CC3 loss.

Figure 6.10: Behaviour of CC1, CC2 and CC3 loss functions. On the left the ground truth is fixed to $A = 0.4, T = 600$, while on the right it is fixed to $A = 0.2, T = 200$.

Another possibility to construct a suitable loss function exploiting the cross-correlation signal is first extracting amplitude R and position ΔT of the non-zero element and then imposing a constraint on these two features:

$$\ell_{\text{CC3}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \lambda_A \left| R - 1 \right| + \lambda_T \left| \Delta T \right| \tag{6.30}$$

This approach is very similar to the hard loss function, but in this case all the operations will be performed in the cross-correlation space in a differentiable manner. Recall that there exists a relationship between cross-correlation and convolution operations, that is:

$$r_k = (x_n * \hat{x}_{-n})(k) \tag{6.31}$$

Therefore, cross-correlation can be computed efficiently in the Fourier domain exploiting the well-known DFT properties. The DFT of the cross-correlation $\mathcal{R}_f = DFT[r_k]$ is given by the multiplication between the DFT of the first signal $\mathcal{X}_f = DFT[x_n]$ and the DTF of the time reversed version of the second signal $\hat{\mathcal{X}}_f^* = DFT[\hat{x}_{-n}]$, where the symbol $*$ indicates the complex conjugate:

$$\mathcal{R}_f = \mathcal{X}_f \cdot \hat{\mathcal{X}}_f^* \tag{6.32}$$

Moreover, from our prior knowledge on the cross-correlation shape, equation 6.27, we have:

$$\mathcal{R}_f = R\, e^{-j\, \Delta T \frac{2\pi}{N} f} \tag{6.33}$$

At this point we observe that the information about amplitude and position of the correlation peak can be fully retrieved looking only at the DFT sample $\mathcal{R}_1 = \mathcal{X}_1 \cdot \hat{\mathcal{X}}_1^*$:

$$\begin{cases} R = |\, \mathcal{R}_1 \,| \\ \Delta T = -Arg(\mathcal{R}_1) \frac{N}{2\pi} \end{cases} \tag{6.34}$$

Putting all the previous considerations together, the overall loss function can be rewritten as:

$$\ell_{\text{CC3}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \lambda_A \left| |\mathcal{X}_1 \cdot \hat{\mathcal{X}}_1^* \,| - 1 \right| + \lambda_T \left| Arg(\mathcal{X}_1 \cdot \hat{\mathcal{X}}_1^*) \right| \tag{6.35}$$

where $\mathcal{X}_1 = \sum_{n=0}^{N-1} x_n \, e^{-jn\frac{2\pi}{N}}$ and $\hat{\mathcal{X}}_1^* = \sum_{n=0}^{N-1} \hat{x}_n \, e^{jn\frac{2\pi}{N}}$. Note that $CC3$ loss is more efficient than the previously introduced correlation based losses, since it does not require the computation of the whole DFTs, but only of the two samples $\mathcal{X}_1$ and $\hat{\mathcal{X}}_1$. Figure 6.10c confirms that $CC3$ loss exhibits the desired behaviour and can be used for the training of our network.

One disadvantage of all correlation based loss functions is the computational inefficiency. They require a lot of computations to be evaluated and this leads to a significant decrease in the optimization speed. In addition, they do not scale well to more complex scenarios. They show the desirable property in the simple case in which the backscattering vectors are formed by a singular specular reflection, but the same behaviour is not guaranteed for more general backscattering shapes.

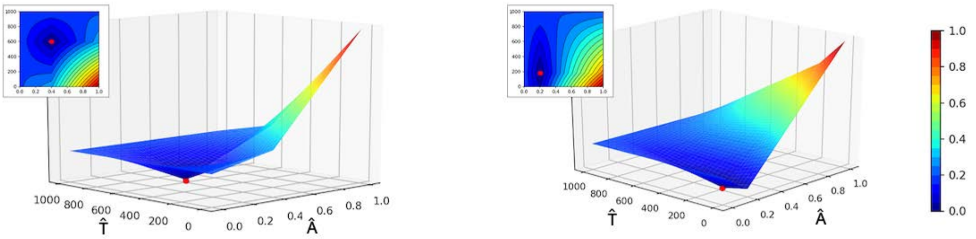


Figure 6.11: Behaviour of EMD loss function. On the left the ground truth is fixed to $A = 0.4, T = 600$, while on the right it is fixed to $A = 0.2, T = 200$.

6.2.2-IV Earth Mover's Distance Loss Function

The last proposed reconstruction loss comes from the idea of comparing the two backscattering vectors considering them as two PMDs and measuring their statistical distance. To convert the backscattering vectors $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ into probability distributions $p_{\boldsymbol{x}}$ and $p_{\hat{\boldsymbol{x}}}$ we simply rescale them in such a way that they satisfy the normalization property. Since the ground truth vector is fixed while the predicted one changes during the optimization, the normalization is performed with respect to the ground truth:

$$p_{\boldsymbol{x}}(n) = \frac{x_n}{X} \qquad p_{\hat{\boldsymbol{x}}}(n) = \frac{\hat{x}_n}{X} \qquad \text{where } X = \sum_{n'=0}^{N-1} x_{n'} \tag{6.36}$$

In this context there are different metrics to quantify the divergence between two PMDs [35], such as Kullback–Leibler divergence, Hellinger distance, Jensen–Shannon divergence, Jeffrey-divergence, and so on. Unfortunately, all these metrics can be evaluated only for probability distributions with a common support and this is not our case. A metric which measures the distance between two distributions without requiring a common support is the *Earth Mover's Distance* (EMD), also knows as *Wasserstein* metric. Informally, if the distributions are interpreted as two different ways of piling up a certain amount of dirt, the EMD represents the minimum cost of turning one pile into the other, where the cost is assumed to be the amount of dirt moved times the distance by which it is moved. It indicates how much mass must be transported to transform one distribution into the other. The EMD has been successfully applied in the literature [36, 37] as loss function in the training of different neural network architectures. In [36] the authors do some considerations showing that apparently simple sequences of probability distributions converge under the EMD but do not under other metrics.

In general, the earth mover's distance is computed solving a linear programming problem, but in the special one-dimensional case it can be efficiently computed scanning the two PMDs and keeping track of how much mass needs to be transported between consecutive bins. This procedure leads to a closed-form solution for the EMD computation, involving the difference between the two Cumulative Mass Distributions (CMDs). Let $P_{\boldsymbol{x}}$ and $P_{\hat{\boldsymbol{x}}}$ be

Loss	Definition	
L1	$\ell_{\text{L1}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \sum_{n=0}^{N-1} \lvert x_n - \hat{x}_n \rvert$	
L2	$\ell_{\text{L2}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \sum_{n=0}^{N-1} \left(x_n - \hat{x}_n\right)^2$	
HARD	$\ell_{\text{HARD}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \lambda_A \Big\lvert \underset{n}{max}\, \hat{\boldsymbol{x}} - \underset{n}{max}\, \boldsymbol{x} \Big\rvert + \lambda_T \Big\lvert \underset{n}{argmax}\, \hat{\boldsymbol{x}} - \underset{n}{argmax}\, \boldsymbol{x} \Big\rvert$	
SOFT	$\ell_{\text{SOFT}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \lambda_A \Big\lvert soft\, \underset{n}{max}\, \hat{\boldsymbol{x}} - soft\, \underset{n}{max}\, \boldsymbol{x} \Big\rvert + \lambda_T \Big\lvert soft\, \underset{n}{argmax}\, \hat{\boldsymbol{x}} - soft\, \underset{n}{argmax}\, \boldsymbol{x} \Big\rvert$	
CC1	$\ell_{\text{CC1}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \Big\lvert 1 - \sum_{k=-N+1}^{N-1} e^{-\frac{k^2}{2\sigma^2}} \cdot r_k \Big\rvert$	(see text for r_k)
CC2	$\ell_{\text{CC2}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \sum_{k=-N+1}^{N-1} e^{-\frac{k^2}{2\sigma^2}} \cdot \lvert 1 - r_k \rvert$	(see text for r_k)
CC3	$\ell_{\text{CC3}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \lambda_A \Big\lvert \big\lvert \mathcal{X}_1 \cdot \hat{\mathcal{X}}_1^* \big\rvert - 1 \Big\rvert + \lambda_T \Big\lvert Arg(\mathcal{X}_1 \cdot \hat{\mathcal{X}}_1^*) \Big\rvert$	(see text for $\mathcal{X}_1$ and $\hat{\mathcal{X}}_1^*$)
EMD	$\ell_{\text{EMD}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \frac{1}{N\,X} \sum_{n=0}^{N-1} \lvert c_n - \hat{c}_n \rvert$	(see text for c_k and $\hat{c}_n$)
wEMD	$\ell_{\text{wEMD}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \frac{1}{N\,X} \sum_{n=0}^{N-1} w_n \lvert c_n - \hat{c}_n \rvert$	(see text for c_k, $\hat{c}_n$ and w_n)

Table 6.3: Summary of all discussed loss functions. Note that the weighted wEMD loss will be introduced in section 6.3.2.

the two CMDs, then the EMD can be computed as:

$$\text{EMD}(p_{\boldsymbol{x}}, p_{\hat{\boldsymbol{x}}}) = \sum_{n=0}^{N-1} \big\lvert P_{\boldsymbol{x}}(n) - P_{\hat{\boldsymbol{x}}}(n) \big\rvert \tag{6.37}$$

Putting all the previous observations together, we can design a proper loss function making it proportional to the average earth mover's distance over the two equivalent probability distributions:

$$\ell_{\text{EMD}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \frac{1}{N}\text{EMD}(p_{\boldsymbol{x}}, p_{\hat{\boldsymbol{x}}}) = \frac{1}{N\,X} \sum_{n=0}^{N-1} \lvert c_n - \hat{c}_n \rvert \tag{6.38}$$

where the cumulative backscattering vectors are:

$$c_n = \sum_{n'=0}^{n} x_{n'} \qquad \hat{c}_n = \sum_{n'=0}^{n} \hat{x}_{n'} \tag{6.39}$$

Figure 6.11 shows that the EMD loss exhibits the desirable property and it is able to drive the optimization algorithm towards the optimal solution in a smooth way. It converges to zero in the perfect matching case, and more important, it is scalable to more complex reflection scenarios. Experimental results proved that the EMD is the best performing loss function between all proposals, thus it is the one used for the reconstruction loss in BVE_ISRP.

6.2.3 Training Procedure

As already said, BVE_ISRP was trained on the simulated dataset T_{ISRP}^{train}. We used the Adam optimization algorithm [38], a variant of the classic stochastic gradient descent which adapt

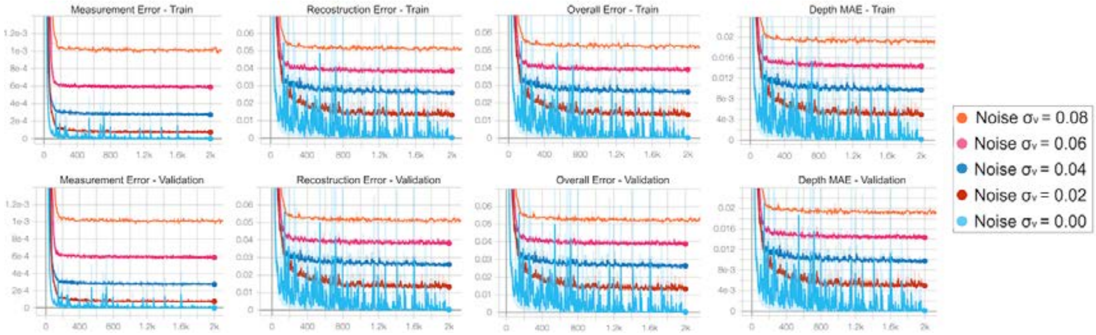


Figure 6.12: Training curves obtained running the optimization of BVE_ISRP for noise levels σ_v =0.00, 0.02, 0.04, 0.06 and 0.08 on training and validation sets. The metrics monitored are, from left to right, the measurement error, the reconstruction error, the overall error and the MAE on the depth estimated using the predicted output backscattering vector.

the learning rate for each weight in the network. It uses an adaptive moment estimation strategy to dynamically change the learning rates based on the behaviour of the optimization. The final setup for BVE_ISRP consists of the weighted squared error loss for the measurement error and the EMD loss for the reconstruction error. We set the parameter λ in equation (4.6) to 20 in order to give more weight to the reconstruction error and forcing the output of the network to be more related to the transient ground truth. The entire dataset was divided into batches of $M^b = 1024$ samples each, and the gradient at each iteration was computed on a single batch. At each iteration, the corresponding batch is loaded into memory and a gaussian zero-mean random noise is added to the real and imaginary parts of the simulated raw ToF data. The input data at each iteration are given by:

$$\mathcal{V} = \Phi\,\mathcal{X} + \mathcal{H} \qquad\qquad \mathcal{H}_{uvm} \sim \mathcal{N}(0, \sigma_v^2) \tag{6.40}$$

The noise is independent and identically distributed across all the pixels in the image and across all the acquired phasors at different modulation frequencies. Changing the noise at each iteration helps to avoid overfitting on the training data and acts as a form of data augmentation. The network never sees the exact same input data twice. Moreover, it helps the network to learn to denoise the input data, giving more importance to the more stable relationships between the acquired phasors and less to small fluctuations around the average value. The noise power is a hyperparameter which defines the Signal-to-Noise ratio in the input data. We run the training for a total number of $E = 2000$ epochs on a Nvidia GeForce GTX 1060 graphic unit. The overall amount of time required was about 6 hours. We repeated the training multiple times, varying the amount of noise σ_v added to the input data. At each epoch we monitored the behaviour of measurement error ℓ_m, reconstruction error ℓ_r and overall error $\ell = \ell_m + \ell_r$, as well as the MAE on the depth estimated using the predicted output backscattering vector. Figure 6.12 reports the behaviour of the considered metrics during the optimization on both the training T^{train}_{ISRP} and the validation T^{valid}_{ISRP}

sets. Clearly, the higher the noise level, the larger the errors of the predictive model will be. Note that in the noise-free case the final estimation is almost perfect with an overall error converging to zero. Looking at the curves we observe also that the errors on the training and validation sets are consistent, indicating good generalization capabilities of the network. The best performing set of weights, *i.e.* the one which minimizes the overall loss function on the validation set, is saved and will be used for the final test on the three real ToF datasets.

6.3 Double Specular Reflection at Pixel-level

At this point we want to apply the proposed approach in an MPI case, trying to reconstruct the shape of a backscattering vector generated by two specular reflections. As will be confirmed by experimental results, the two specular reflections hypothesis covers a large number of real-world cases since the quadratic decay with distance produces very weak higher order reflection, that can be well approximated by only two main returns. This is the same rationale underlying the work of many authors [12, 13, 14] which correct the MPI phenomenon assuming it is formed only by two components. Clearly it is a restrictive assumption, therefore we do not expect that the developed method will be able to compensate for the multi-path effect in all the cases. It will definitely fail in case of diffuse reflections or more than two reflections.

The generative model used for this second method is the sparse generative model discussed in section 6.1, where we consider only two specular reflections:

$$\begin{aligned} G^2 : \qquad & \mathbb{R}^2 & \rightarrow \quad & \mathcal{D}^2_{\boldsymbol{x}} \subseteq \mathbb{R}^N \\ \boldsymbol{z} = \left[A_1, T_1, A_2, T_2\right]^T & & \rightarrow \quad & \boldsymbol{x} = G^2(\boldsymbol{z}) = \left[x_0, \ldots, x_{N-1}\right]^T \end{aligned} \tag{6.41}$$

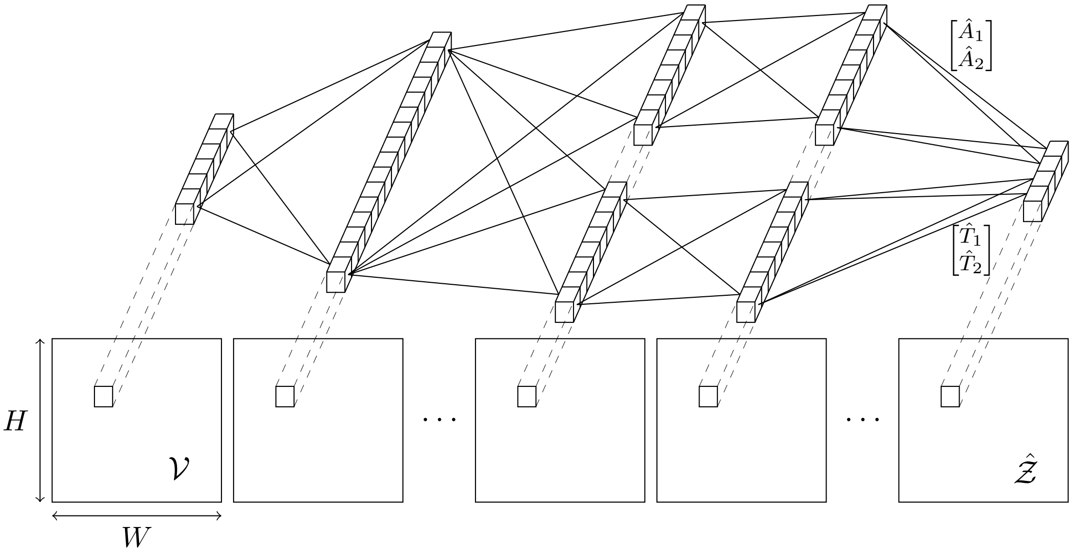
with:

$$x_n = A_1\, e^{-\frac{(n-\tilde{T}_1)^2}{2\sigma^2}} + A_2\, e^{-\frac{(n-\tilde{T}_2)^2}{2\sigma^2}} \tag{6.42}$$

We keep working independently for each pixel, neglecting the relationships between adjacent pixels and estimating the output backscattering vector for pixel (u, v) looking only at the corresponding acquired phasors:

$$\mathcal{X}_{uv} = G^2(\mathcal{Z}_{uv}) = G^2(P_{\boldsymbol{\theta}}(\mathcal{V}_{uv})) \tag{6.43}$$

We will refer to this method with the name BVE_2SRP, an abbreviation which stands for "Backscattering Vector Estimation in the Double Specular Reflection case working at Pixel-level". In the following sections we are going to describe its architecture and a small variant of the EMD loss function used in this case. The training strategy is exactly the same adopted for the first method, discussed in section 6.2.3. The final experimental results are provided in section 7.2.



$\begin{bmatrix}\hat{A}_1\\\hat{A}_2\end{bmatrix}$
$\begin{bmatrix}\hat{T}_1\\\hat{T}_2\end{bmatrix}$
H
W
$\mathcal{V}$
$\hat{\mathcal{Z}}$
...
...

Figure 6.13: Graphical illustration of the predictive network architecture of BVE_2SRP.

6.3.1 Predictive Neural Network Architecture

The network architecture of the predictive model used in BVE_2SRP is very similar to the one employed for the first method. It is always formed by a first stack of convolutional layers to denoise the input data and extract relevant features, followed by two parallel branches. The first branch is used to estimate the amplitudes $[A_1, A_2]^T$ of the two specular reflections, while the second branch estimates their time positions $[T_1, T_2]^2$. The number of filters in the hidden layers and activation functions are preserved. The architecture of this second proposed method is depicted in figure 6.13, while table 6.4 reports the details of each layer. The total number of learnable parameters of this second predictive model is 14 052.

6.3.2 Weighted EMD Loss Function

All the considerations done in section 6.2.2 remain valid also in this context. In BVE_2SRP, we use the weighted squared error loss for the measurement error and a variant of the EMD for the reconstruction error. BVE_2SRP must learn to predict a first peak associated to the direct component and a second peak associated to the global component, where the direct component is very likely to be larger than the global component. Experimental results show that the EMD does not perform very well in this case, since in many cases the network confuses the MPI-free case, *i.e.* single large peak, with the case in which the backscattering vector is formed by a large main peak and a small second one. The measurement error helps in the discrimination, but for two very close peaks it is not sufficient. The final effect is that the network learns to always predict a small global component even if it is not present in the ground truth backscattering vector and this inevitably introduces an error in the final depth

Layer	Kernel size	Input dimension	Output dimension	Activation
Conv $c_1(\mathcal{V})$	$1\times1\times6\times64$	$W\times H\times6$	$W\times H\times64$	ReLU
Conv $c_2(c_1)$	$1\times1\times64\times64$	$W\times H\times64$	$W\times H\times64$	ReLU
Conv $c_3(c_2)$	$1\times1\times64\times64$	$W\times H\times64$	$W\times H\times64$	ReLU
Conv $a_1(c_3)$	$1\times1\times64\times32$	$W\times H\times64$	$W\times H\times32$	ReLU
Conv $a_2(a_1)$	$1\times1\times32\times16$	$W\times H\times32$	$W\times H\times16$	ReLU
Conv $a_3(a_2)$	$1\times1\times16\times2$	$W\times H\times16$	$W\times H\times2$	Sigmoid
Conv $t_1(c_3)$	$1\times1\times64\times32$	$W\times H\times64$	$W\times H\times32$	ReLU
Conv $t_2(t_1)$	$1\times1\times32\times16$	$W\times H\times32$	$W\times H\times16$	ReLU
Conv $t_3(t_2)$	$1\times1\times16\times2$	$W\times H\times16$	$W\times H\times2$	Sigmoid
Concat $z(a_3,t_3)$		$W\times H\times2$, $W\times H\times2$	$W\times H\times4$	

Table 6.4: Hyperparameters used in the predictive network architecture of BVE_2SRP. For each layer, the table reports the kernel dimensions, the input and output dimensions and the activation function applied at its output.

estimation. Figure 6.14a reports a typical situation where the ground truth is formed by the direct component only, but the network predicts also a small global component.

Different strategies have been tested to solve the problem. In particular we tried different values for the parameter λ in equation (4.6) but we were not able to find a good trade-off to balance measurement and reconstruction errors. Note that this problem arises because in the EMD the cost of moving a small mass for a long distance is much lower than the cost of moving a large mass. The network learns to minimize the loss function predicting with a good accuracy the direct component and neglecting the small errors due to the erroneous global component positioning. Figures 6.14b-c illustrate the standard EMD loss computation through the absolute difference between the two cumulative backscattering vectors. Looking at the plots we can identify two main types of errors, namely errors of the first and second type. The error e_1 of type 1, indicated in figures with green colour, is due to a small imprecision in the prediction of the direct component. Instead, the error e_2 of type 2, indicated with the purple colour, is due to a completely wrong positioning of the small global component. In the standard EMD loss computation, errors of the first type dominate and therefore the network prefers to minimize them, neglecting the errors of the second type. Using the standard earth mover's distance, the ratio e_2/e_1 is in general low. One idea to solve the problem is to design a new variant of the EMD loss which changes the proportions between these two types of error, giving more importance to errors of the second type. Our proposal is to use a *weighted EMD* loss function defined as:

$$\ell_{\mathrm{wEMD}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \frac{1}{N\,X}\sum_{n=0}^{N-1} w_n \underbrace{\left|c_n - \hat{c}_n\right|}_{d_n} \tag{6.44}$$

where the weights, intuitively, should be proportional to the "density" of d_n. Errors of type

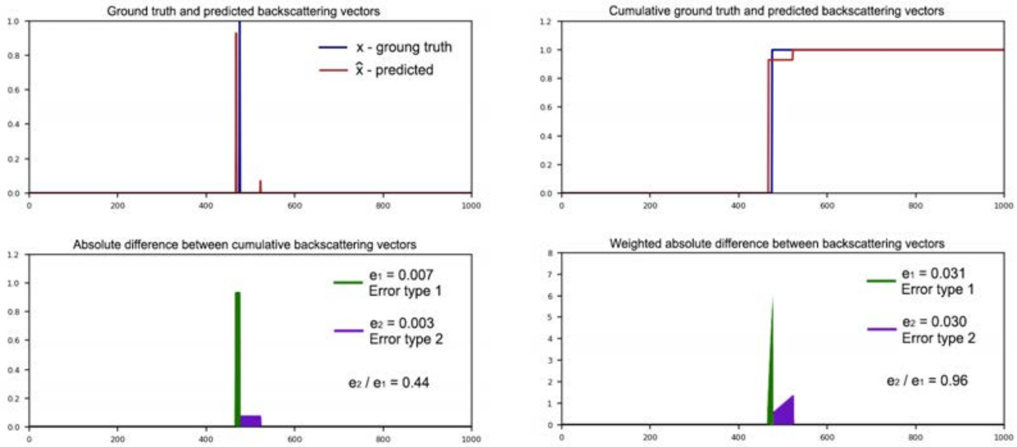


Figure 6.14: Graphical illustration of the difference between the standard EMD and the weighted wEMD.

two are peaked and very concentrated in time and thus have lower "density" than errors of type two, which instead are smaller but more time dilated. To compute the weights we use the causal moving average of d_n with a sliding windows of $W = 100$ samples, that is:

$$w_n = \frac{1}{W}\sum_{k=0}^{W-1} d_{n-k} = \frac{1}{W}\sum_{k=0}^{W-1} \left|c_{n-k} - \hat{c}_{n-k}\right| \tag{6.45}$$

In this way we are giving more relevance to points which are preceded by other non-zero samples, like the ones contributing to error of type 2. Figure 6.14d shows how the proportion between the two errors changes using the weighted variant and how the ratio e_2/e_1 increases. Note that the weighted earth mover's loss continues to satisfy the desirable property of the gradient. It is the loss function used for the reconstruction error in BVE_2SRP.

6.3.3 Training Procedure

For the optimization of BVE_2SRP we adopted exactly the same procedure described for the first method in section 6.2.3. We used T^{train}_{2SRP} as training set and T^{valid}_{2SRP} as validation set. We adopted the weighted L2 loss for the measurement error and the weighted EMD loss for the reconstruction error. Figure 6.15 reports the behaviour of the metrics of interest on both training and validation sets. Note that, for the same level of noise, BVE_2SRP is more prone to errors and it converges to a higher residual error with respect to the previous method. This is expected because now there are more DoFs in the solution. In any case, the network converges to a low residual error.

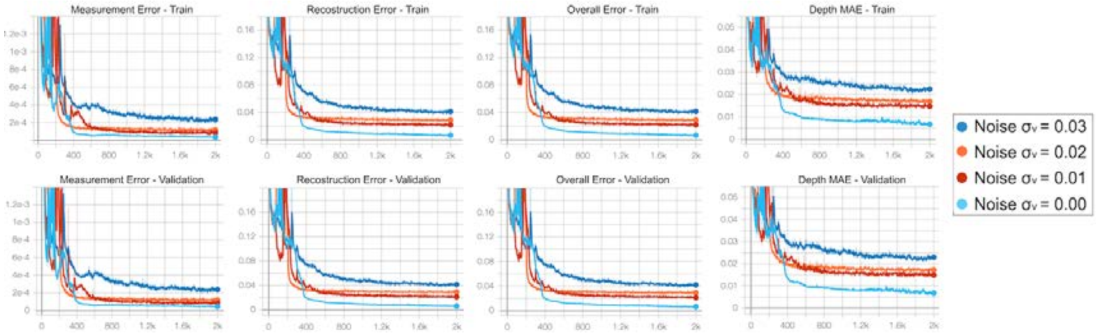


Figure 6.15: Training curves obtained running the optimization of BVE_2SRP for noise levels σ_v =0.00, 0.02, 0.04, 0.06 and 0.08 on training and validation sets. The metrics monitored are, from left to right, the measurement error, the reconstruction error, the overall error and the MAE on the depth estimated using the predicted output backscattering vector.

6.4 Double Specular Reflection at Local level

The third and last method investigated in this work relies on the idea that it is possible to take advantage of the spatial correlation on the input data to improve the accuracy of the final prediction. Here, we consider always the two specular reflections case. The objective is to study how the spatial correlation can be exploited to gain some benefits over the previous implementation. The predictive model is designed in such a way that the output for pixel (u, v) is a function of the whole neighbourhood of size $(2P + 1) \times (2P + 1)$ centred on the corresponding input pixel:

$$\mathcal{Z}_{uv} = P_{\boldsymbol{\theta}}\Big(\big\{\mathcal{V}_{u'v'}\big|u' = u + k;\ v' = v + j;\ k, j = -P, ..., +P\big\}\Big) \tag{6.46}$$

In our implementation we use a square receptive field of size $(2P + 1) = 3$. This choice has be done to limit the exponential growth of the network complexity and because experimental results show that larger values favours the overfitting of the network on the training data, decreasing its generalization capabilities. The predictive model uses the raw ToF data from 9 input pixels to estimate the backscattering vector for each output pixel. The generative model continues to work at single pixel level since there is a one-to-one relationship between the output backscattering vector and its compressed representation.

We will refer to this last method with the name BVE_2SRL, an abbreviation which stands for "Backscattering Vector Estimation in the Double Specular Reflection case working at Local level". We expect that it will be more robust against noise, but still limited by the two specular reflections assumption. In the next section we are going to present the network architecture used to exploit the spatial correlation. The loss functions and the training strategy are the same adopted before. The experimental results on the real-world datasets are provided in section 7.3.

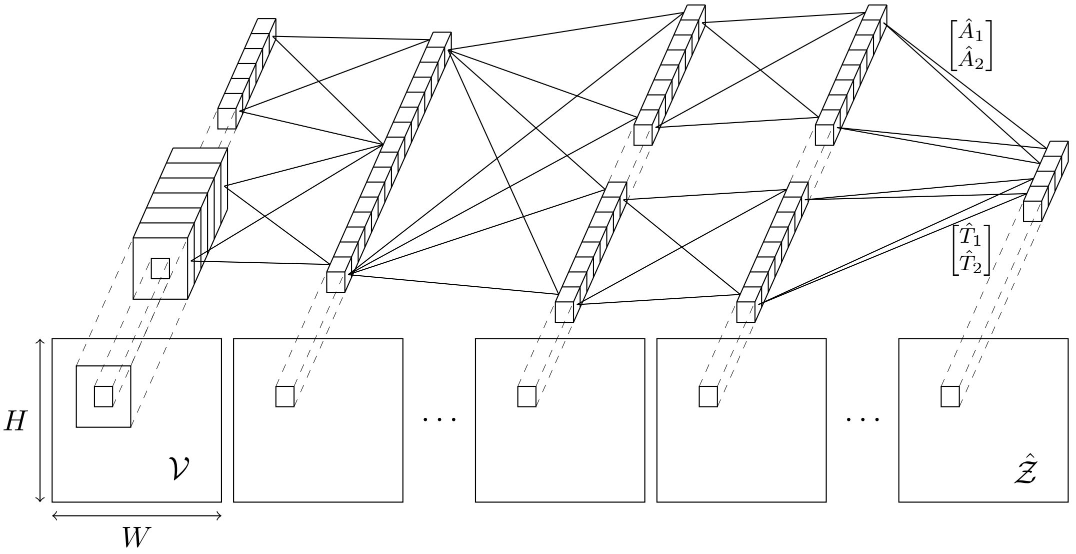

$\begin{bmatrix}\hat{A}_1\\ \hat{A}_2\end{bmatrix}$
$\begin{bmatrix}\hat{T}_1\\ \hat{T}_2\end{bmatrix}$
H
W
$\mathcal{V}$
$\hat{\mathcal{Z}}$
...
...

Figure 6.16: Graphical illustration of the predictive network architecture of BVE_2SRL.

6.4.1 Network architecture

There are different approaches to exploit spatial correlation in the input raw ToF data. Our proposal is to use the same network architecture of before increasing only the receptive field of the first layer. Now, the first convolutional layer extracts features for each pixel looking at a neighbourhood of size 3 around the pixel itself. All the subsequent layers keep working pixel-wise. Particular attention must be paid to the design of the first layer. The results achieved by BVE_2SRP suggest that the predictive model is able to perform a good estimation for output pixel (u, v) only looking at the corresponding input pixel. This means that most of the information required for the final prediction is contained in the input pixel (u, v). The first layer of BVE_2SRL has been designed with the intent of using the spatial correlation to increase the amount of information available, without modifying what is already present in the central pixel (u, v). For this reason, the first layer is formed by the concatenation of the features extracted using two convolutional kernels. The first kernel with size $1 \times 1 \times C_p$ acts only on the central pixels (u, v) as before, while the second one with size $3 \times 3 \times C_l$ acts on the whole neighbourhood $\{(u', v')|u' = u + k;\ v' = v + j;\ k, j = -1, 0, +1\}$:

$$\mathcal{C}_{1,uv} = \big[\underbrace{-\mathcal{P}_{uv}-}_{C_p}, \underbrace{-\mathcal{L}_{uv}-}_{C_l}\big]^T \tag{6.47}$$

where:

$$\mathcal{P}_{u,v} = \mathcal{P}\big(\mathcal{V}_{u,v}\big) \tag{6.48}$$

$$\mathcal{L}_{u,v} = \mathcal{L}\big(\{\mathcal{V}_{u'v'}\big|u' = u + k;\ v' = v + j;\ k, j = -1, 0, +1\}\big) \tag{6.49}$$

Layer	Kernel size	Input dimension	Output dimension	Activation
Conv & Slice $p(\mathcal{V})$	$1 \times 1 \times 6 \times 64$	$(W+2) \times (H+2) \times 6$	$W \times H \times 64$	ReLU
Conv $l(\mathcal{V})$	$3 \times 3 \times 6 \times 64$	$(W+2) \times (H+2) \times 6$	$W \times H \times 64$	ReLU
Concat $c_1(p, l)$		$W \times H \times 64$, $W \times H \times 64$	$W \times H \times 128$	
Conv $c_2(c_1)$	$1 \times 1 \times 64 \times 64$	$W \times H \times 128$	$W \times H \times 64$	ReLU
Conv $c_3(c_2)$	$1 \times 1 \times 64 \times 64$	$W \times H \times 64$	$W \times H \times 64$	ReLU
Conv $a_1(c_3)$	$1 \times 1 \times 64 \times 32$	$W \times H \times 64$	$W \times H \times 32$	ReLU
Conv $a_2(a_1)$	$1 \times 1 \times 32 \times 16$	$W \times H \times 32$	$W \times H \times 16$	ReLU
Conv $a_3(a_2)$	$1 \times 1 \times 16 \times 2$	$W \times H \times 16$	$W \times H \times 2$	Sigmoid
Conv $t_1(c_3)$	$1 \times 1 \times 64 \times 32$	$W \times H \times 64$	$W \times H \times 32$	ReLU
Conv $t_2(t_1)$	$1 \times 1 \times 32 \times 16$	$W \times H \times 32$	$W \times H \times 16$	ReLU
Conv $t_3(t_2)$	$1 \times 1 \times 16 \times 2$	$W \times H \times 16$	$W \times H \times 2$	Sigmoid
Concat $z(a_3, t_3)$		$W \times H \times 2$, $W \times H \times 2$	$W \times H \times 4$	

Table 6.5: Hyperparameters used in the predictive network architecture of BVE_2SRL. For each layer, the table reports the kernel dimensions, the input and output dimensions and the activation function applied at its output.

With this configuration, the receptive field of the whole network turns out to be equivalent to the receptive field of the first layer, that is 3×3. Since all layers use *valid* convolutions, the output spatial dimensions are 2 pixels smaller than the input ones.

Figure 6.16 reports a graphical illustration of the predictive network architecture used in BVE_2SRL, while table 6.5 gives the details of each layer. The total number of learnable parameters in this case is 21 668.

6.4.2 Training Procedure

We run the optimization of BVE_2SRLaccording to the procedure already adopted for the other two methods and described in section 6.2.3. We used T_{2SRL}^{train} as training set and T_{2SRL}^{valid} as validation set. We adopted the weighted L2 loss for the measurement error and the weighted EMD loss for the reconstruction error. Figure 6.17 shows the behaviour of the metrics of interest during the optimization procedure. Comparing the residual errors after convergence of BVE_2SRP and BVE_2SRL , we observe that spatial correlation makes the model more resilient to random noise in the validation set. In the noise-free case the two methods converge to a very similar value, while in case of noise this last method reaches a lower residual error.

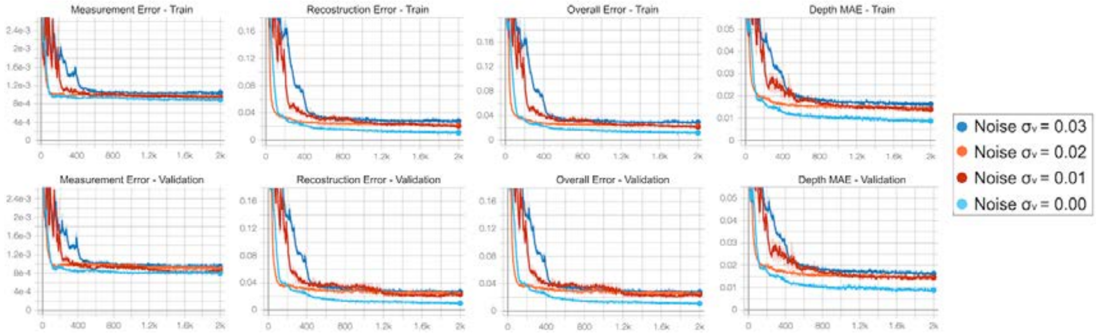


Figure 6.17: Training curves obtained running the optimization of BVE_2SRL for noise levels σ_v =0.00, 0.02, 0.04, 0.06 and 0.08 on training and validation sets. The metrics monitored are, from left to right, the measurement error, the reconstruction error, the overall error and the MAE on the depth estimated using the predicted output backscattering vector.

If you've got the truth you can demonstrate it. Talking doesn't prove it.

Robert A. Heinlein

7 Experimental Results

To conclude our work, in this chapter we are going to present some experimental results obtained running the three proposed methods for backscattering vector estimation. The performance are evaluated from a qualitative and quantitative point of view on the three real datasets S_3, S_4 and S_5 introduced in section 5.4. Due to the fact that for the real datasets we do not have available ground truth transient scenes but only ground truth depth maps, the three methods are evaluated with respect to their capabilities to correct the MPI effect. In particular, given the raw ToF data $\mathcal{V}$, we can run the proposed approach to reconstruct the transient scene $\hat{\mathcal{X}}$ and use algorithm A.1 to retrieve the corresponding depth map $\hat{\mathcal{D}}^{\text{DeepBVE}}$:

$$\hat{\mathcal{D}}^{\text{DeepBVE}} = \textsc{Trans2Depth}\big(\, \underbrace{G_{\boldsymbol{\xi}}(P_{\boldsymbol{\theta}}(\mathcal{V}))}_{\hat{\mathcal{X}}},\, d_{res} \,\big) \tag{7.1}$$

where $G_{\boldsymbol{\xi}}$ and $P_{\boldsymbol{\theta}}$ are respectively the generative and the predictive models and $d_{res} = 7.49\text{mm}$ is the depth resolution.

The proposed methods are then compared to the standard ToF technique for depth estimation, as well as to other state-of-the-art MPI correction algorithms. The standard ToF formula (2.2) for depth estimation uses a single modulation frequency and do not account for multi-path effects. For a fair comparison with our approach, we consider the higher modulation frequency used to acquire the data, *i.e.* $f_H = \texttt{max}(f_1, \dots, f_M) = 60\text{MHz}$. The depth map $\hat{\mathcal{D}}^{\text{60MHz}}$ provided by the standard ToF technique is computed through algorithm A.2:

$$\hat{\mathcal{D}}^{60MHz} = \textsc{ToFDepth}\big(\, \mathcal{V}, \{f_1, \dots, f_M\} \,\big) \tag{7.2}$$

Note that the higher modulation frequency provides the higher depth resolution and noise resilience, therefore we are comparing our methods with the best depth estimation obtained using a single frequency. The other modulation frequencies are used here only to perform phase unwrapping. Figure 7.1 reports the depth error maps obtained applying the standard ToF technique on the three real datasets.

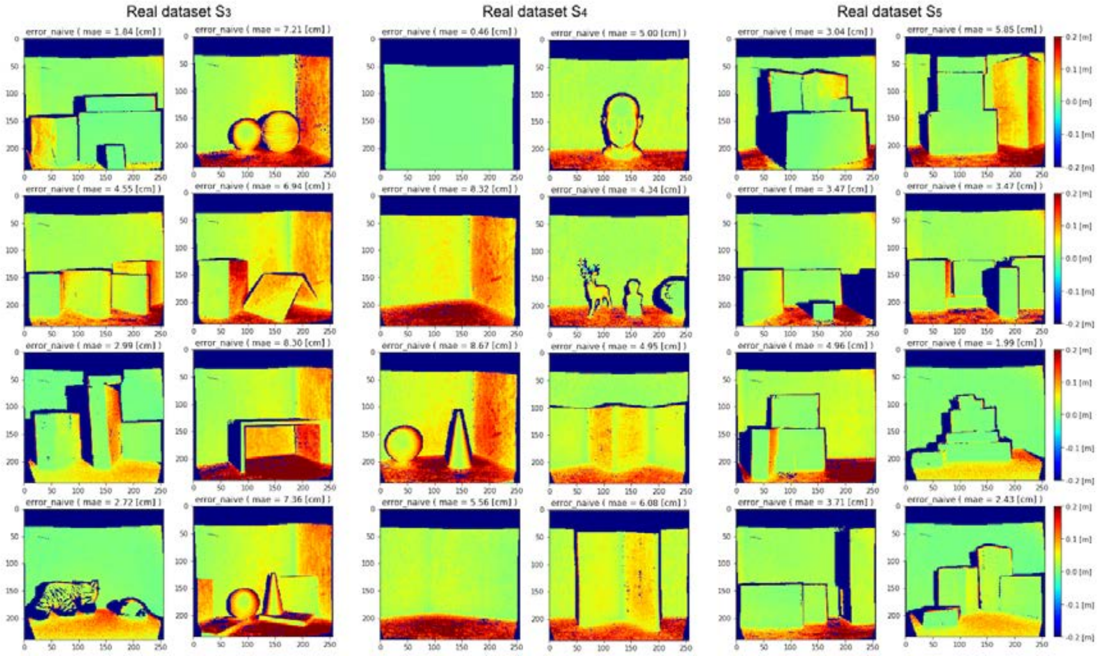


Figure 7.1: Depth error maps (7.3) obtained applying the standard ToF technique at 60MHz on the real datasets S_3, S_4 and S_5. Blue colour indicates depth underestimation, while red colour indicates depth overestimation. The dark blue areas are those for which we do not have ground truth depth. Each scene reports also the corresponding MAE (7.4).

The metric used to quantify the error in the depth estimation is the Mean Absolute Error (MAE). If we indicate with $\mathcal{D}^{\textsc{m}} \in \mathbb{R}^{W\times H}$ the depth map obtained applying method M and with $\mathcal{D} \in \mathbb{R}^{W\times H}$ the depth ground truth, the error map is given by:

$$\mathcal{E}^{\textsc{m}}_{uv} = \mathcal{D}^{\textsc{m}}_{uv} - \mathcal{D}_{uv} \tag{7.3}$$

and the MAE computed over the entire test set S results:

$$e^{\textsc{m}}_{S} = \frac{1}{M}\sum_{i=0}^{M-1}\sum_{(u,v)} \left|\mathcal{E}^{\textsc{m}}_{iuv}\right| = \frac{1}{M}\sum_{i=0}^{M-1}\sum_{(u,v)} \left|\mathcal{D}^{\textsc{m}}_{iuv} - \mathcal{D}_{iuv}\right| \tag{7.4}$$

The lower is the MAE the better is the matching between estimated and ground truth depth maps, and therefore the better are the MPI correction capabilities.

7.1 Performance of BVE_1SRP

Figure 7.2 reports the depth error maps obtained applying BVE_1SRP on the three real datasets. The regions affected by MPI are those that present high depth overestimation and which are indicated with the red colour in the plots. MPI is present mainly on surfaces with a strong inclination, in particular on the floor, and near corners and concavities, where the light rays bounce multiple time and arrive with different delays at the camera sensor. Comparing BVE_1SRP with the standard ToF technique we observe that the two approaches achieve roughly the same performance. As expected, they are not able to correct MPI because they do not account for it. Note that they are able to reliably reconstruct the range value on flat regions where the ideal ToF assumption holds with good approximation, like in the first flat wall scene of dataset S_4. The average MAEs over datasets S_3, S_4 and S_5 obtained applying the single frequency approach are respectively 5.24, 5.43 and 3.62 cm. The corresponding values obtained from our BVE_1SRP method are 5.00, 5.01 and 3.33 cm. The proposed method performs slightly better since, exploiting data acquired at multiple modulation frequencies, it is able to filter out the noise in a more reliable manner. These results are not good from a MPI correction prospective, but turn out to be very promising because they confirm the feasibility of the proposed approach as well as the the effectiveness of the loss function and the training procedure designed.

7.2 Performance of BVE_2SRP

Figure 7.3 shows the depth error maps obtained applying BVE_2SRP on the three real datasets. Even if we are considering only two specular reflections, now we can appreciate a strong improvement in the performance compared to the previous method. The average MAEs over the three datasets are now reduced to 2.86, 3.43 and 2.52 cm. Experimental results confirm

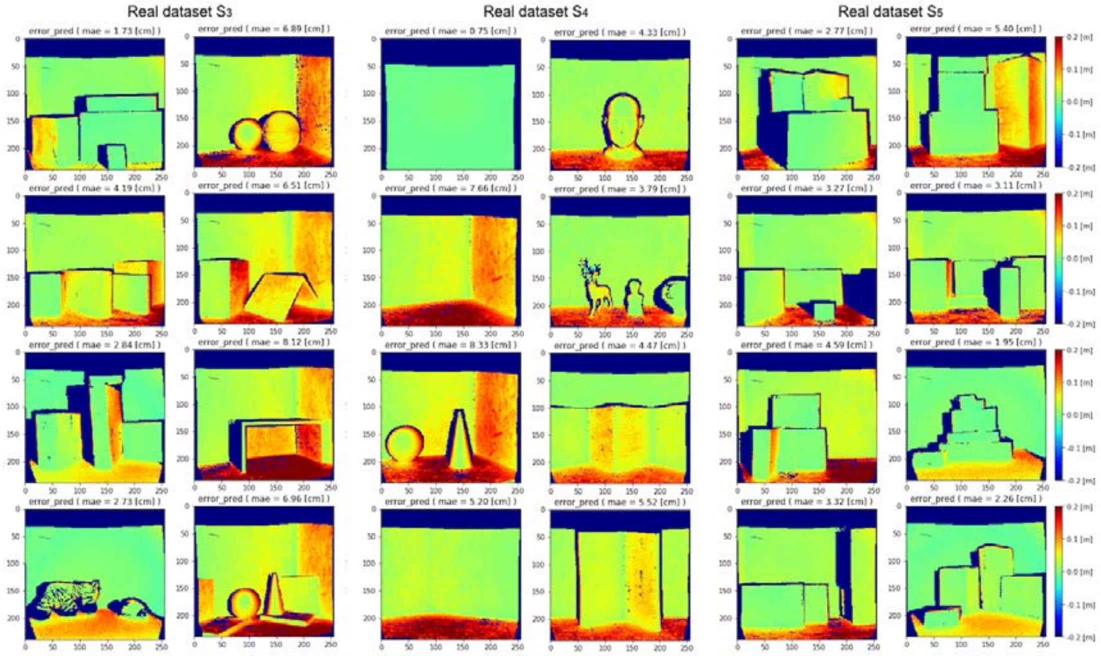


Figure 7.2: Depth error maps (7.3) obtained applying BVE_ISRP on the real datasets S_3, S_4 and S_5. Blue colour indicates depth underestimation, while red colour indicates depth overestimation. The dark blue areas are those for which we do not have ground truth depth available. Each scene reports also the corresponding MAE (7.4).

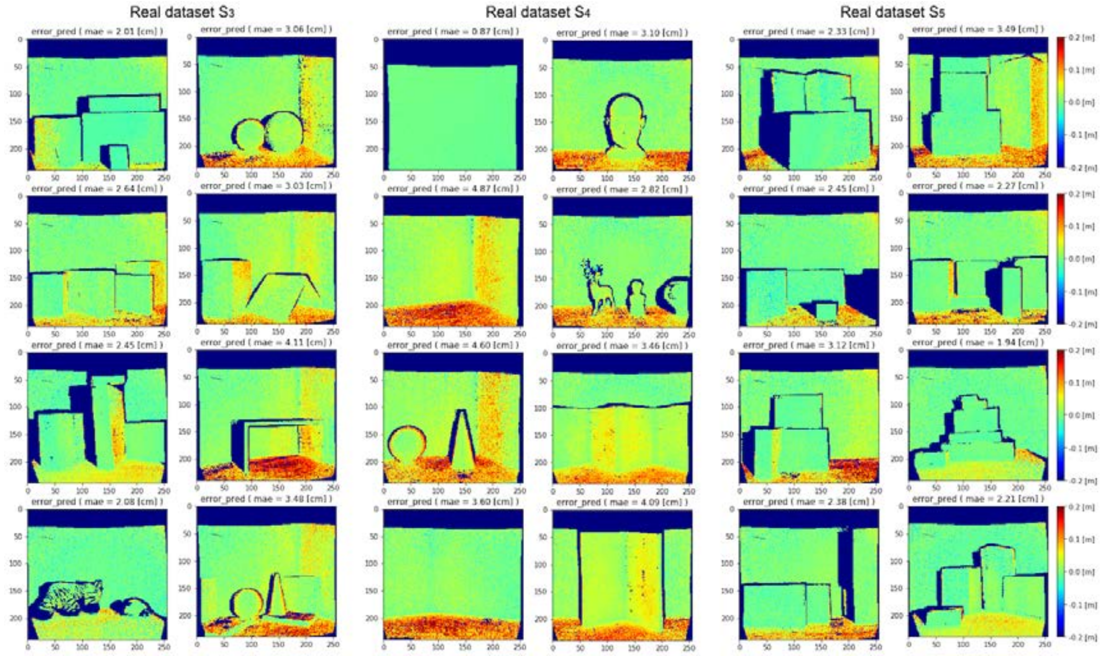


Figure 7.3: Depth error maps (7.3) obtained applying BVE_2SRP on the real datasets S_3, S_4 and S_5. Blue colour indicates depth underestimation, while red colour indicates depth overestimation. The dark blue areas are those for which we do not have ground truth depth available. Each scene reports also the corresponding MAE (7.4).

that many real-world cases can be approximated by two specular reflections. This is due to the fact that, since light power decays with the square of the distance, higher order reflections reach the camera sensor very weak. Moreover, real lambertian surfaces present always a fraction of specular reflections and thus our assumption, in first approximation, holds also for those surfaces. Note that this second method is able to improve the depth estimation on MPI-affected regions without degrading too much the performance on MPI-free areas. In the first flat wall scene of dataset S_4 the MAE slightly increases of 12mm. We achieved this result only after the introduction of the weighted EMD loss function. Some depth reconstruction errors are still present, especially on the floor regions. These areas are probably subjected to more complex light transport events and therefore require the extension of our hypotheses to be resolved.

7.3 Performance of BVE_2SRL

In figure 7.4 the depth error maps obtained applying BVE_2SRL on the three real datasets are shown. These results confirm our intuition. The spatial correlation helps to improve the final prediction making the network more robust to noise. With respect to the single pixel

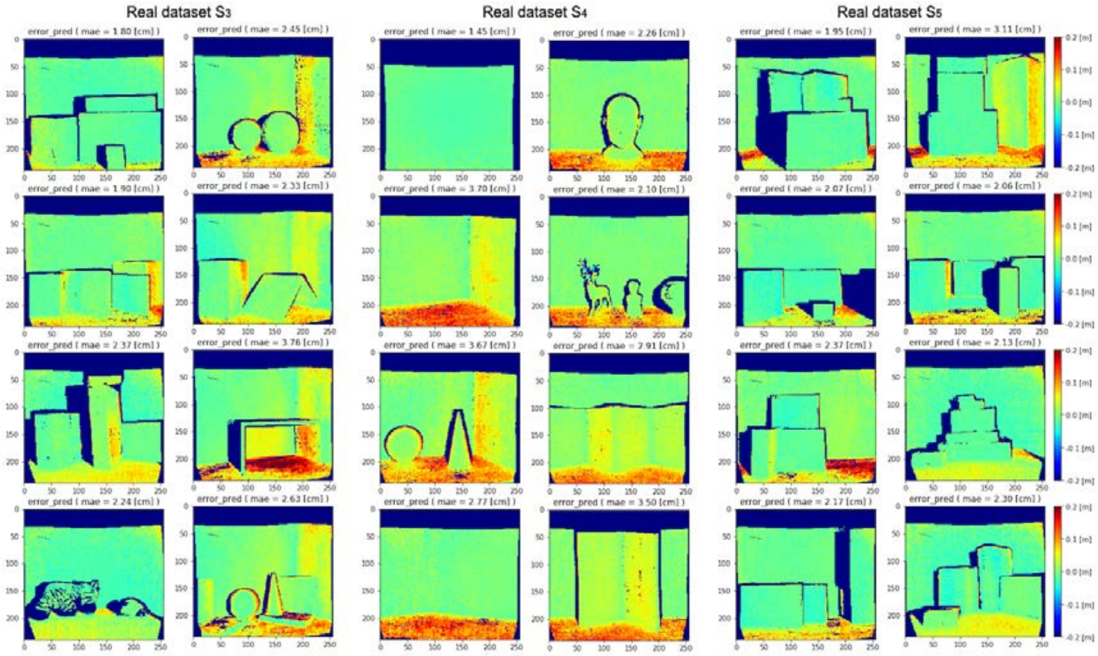


Figure 7.4: Depth error maps (7.3) obtained applying BVE_2SRL on the real datasets S_3, S_4 and S_5. Blue colour indicates depth underestimation, while red colour indicates depth overestimation. The dark blue areas are those for which we do not have ground truth depth available. Each scene reports also the corresponding MAE (7.4).

approach, this last method produces more smooth outputs. In this case, the average MAEs over datasets S_3, S_4 and S_5 result 2.43, 2.79 and $2.27\ \mathrm{cm}$. Note that, since BVE_2SRL is still constrained by the two specular reflections assumption, it cannot do better that BVE_2SRP on more complex MPI scenarios. The gap in performance is only due to the advantages provided by the spatial correlation.

Figure 7.5 reports a comparison between the outputs of BVE_1SRP and BVE_2SRL in a real scene. We can see that the introduction of a second specular reflection allows to compensate for the MPI effect in many cases, providing a more reliable depth estimation. In particular, for the considered scene the MAE is reduced by 64%. The estimation of the entire time-resolved backscattering vector allows to compute either the depth associated to the direct component, which represents the real range value, and the depth associated to the second specular reflection. Unfortunately, we do not have transient ground truth in the real datasets so we cannot quantify the precision of our second reflection prediction. However, some qualitative observations can be done. Looking at the differential depth, obtained as difference between the direct and the global depths, we observe spatial correlation. This means that the network does not place the global component at random only to minimize the loss function, but it learned a reasonable strategy that produces outputs coherent with the real

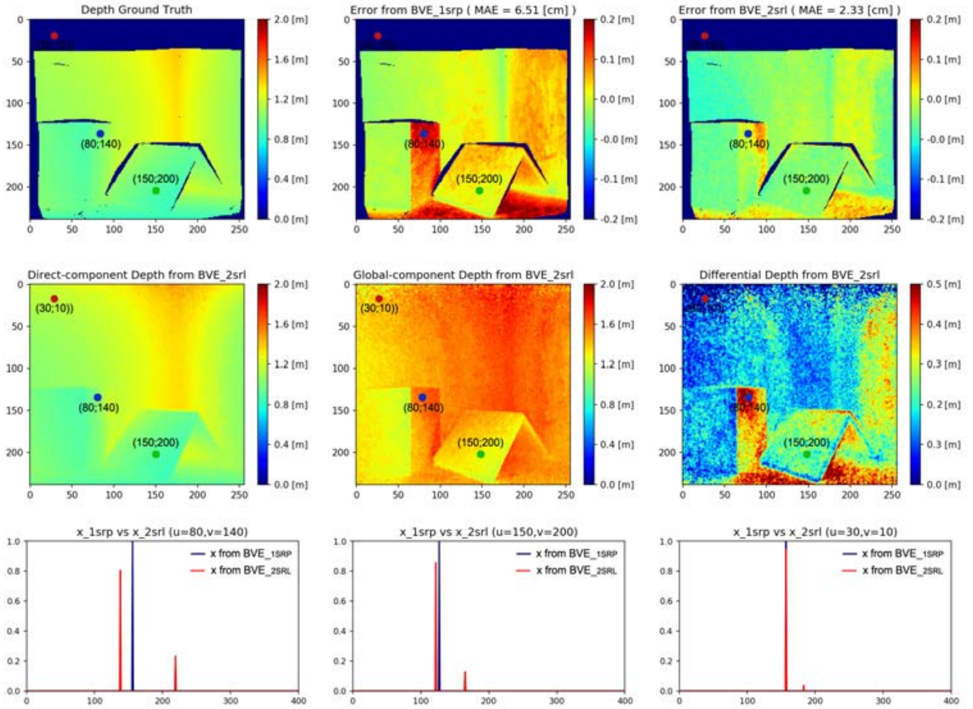


Figure 7.5: Comparison between BVE_1SRP and BVE_2SRL. In the first row we have the depth ground truth of the considered real-world scene and the depth error maps obtained respectively from BVE_1SRP and BVE_2SRL. The second row refers to the output of BVE_2SRL. It reports the depth obtained looking at the direct component, the depth obtained looking at the global component and the difference between the two. In the last raw there is a comparison between the backscattering vectors obtained from BVE_1SRP and from BVE_2SRL , for three different pixels.

scene geometries.

7.4 Comparison with State-Of-The-Art Techniques

Finally, we want to show how the proposed methods behave compared to other state-of-the-art algorithms for MPI compensation. The compared algorithms are the SRA method proposed by Freedman *et al.* [7], the DeepToF proposed by Marco *et al.* [17] and the two methods proposed by Agresti *et al.* [19, 20]. For this comparison we will take as baseline the work of Agresti *et al.* [20]. Table 7.1 reports the MAEs (7.4) for the compared algorithms on the three real datasets S_3, S_4 and S_5. The values indicated in brackets refer to the MAEs after the bilateral filtering of the corresponding depth maps. A bilateral filter [39] is a non-linear, edge-preserving noise reduction technique which substitutes each pixel value with an weighted average on nearby pixels. The weights are given by a combination of two gaussian functions, one depending on the euclidean distance of pixels and the other on the absolute difference of pixel values. The comparison confirms what we have already said. The first proposed BVE_1SRP method performs slightly better than the single frequency approach, while BVE_2SRP and BVE_2SRL are able to provide a much more reliable depth estimation. The fact that the exploitation of spatial correlation allows to reconstruct a smoothed depth map is appreciable also looking at the difference in the gaps between raw and filtered MAEs for BVE_2SRP and BVE_2SRL. In real-world scenes our last method achieves performance comparable with respect to the other state-of-the-art algorithms. After filtering, it performs better than all the other approaches, except for the unsupervised domain adaptation technique of Agresti *et al.* [20]. BVE_2SRL, after filtering, produces an error of 2.60 cm on S_4 and an error of 2.12 cm on S_5, while the method of Agresti *et al.* [20] produces errors respectively of 2.36 and 1.66 cm. Note that the unsupervised domain adaptation technique has been trained on real scenes similar to the ones in S_3, S_4 and S_5, while our approach uses a completely different synthetic training set generated at random. It is remarkable to notice that, BVE_2SRL outperforms with a large margin the SRA method, that across the compared algorithms is the one which focuses on the most similar setup. Both acquire data at three modulation frequencies and use a physical model to describe the MPI effect under the specular reflections assumption. The MAEs on datasets S_4 and S_5 obtained applying the SRA method are 5.11 and 3.37 cm.

To conclude, figure 7.6 shows the depth profiles estimated in proximity of a corner using the compared algorithms. Also in this case, BVE_2SRL is able to reconstruct depth values very closed to the ground truth.

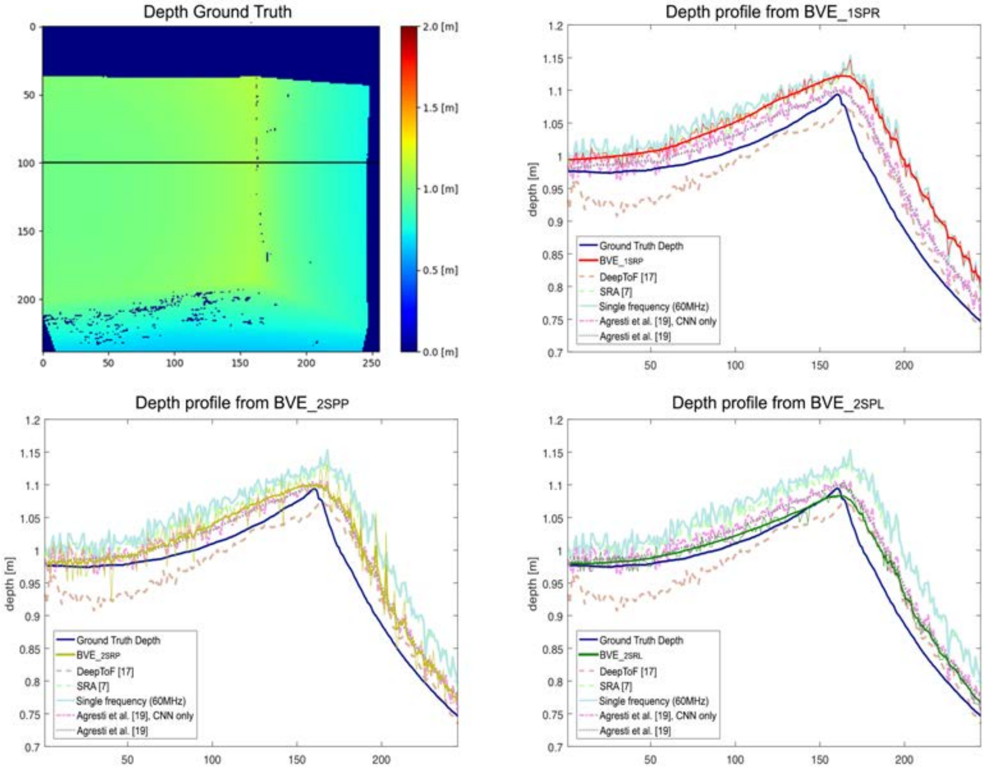


Figure 7.6: Depth profile estimation in proximity of a corner. The first image reports the depth ground truth of the considered real-world scene. The other plots compare the depth profile estimated applying different state-of-the-art MPI correction algorithms with the one obtained from the three proposed methods. The depth plotted corresponds to the highlighted horizontal line is the first figure, *i.e.* $v = 100$.

	S_3 dataset $[cm]$	S_4 dataset $[cm]$	S_5 dataset $[cm]$
Single frequency (60MHz)	5.24	5.43	3.62
SRA [7]	n.a.	5.11	3.37
DeepToF [17]	n.a.	5.13	6.68
+ calibration	n.a.	5.46	3.36
Agresti *et al.* [19]	n.a.	3.19	2.22
+ unsupervised DA [20]	n.a.	2.36	1.66
BVE_1SRP	5.00 (4.94)	5.01 (5.01)	3.33 (3.25)
BVE_2SRP	2.86 (2.31)	3.43 (2.99)	2.52 (1.88)
BVE_2SRL	2.43 (2.30)	2.79 (2.60)	2.27 (2.12)

Table 7.1: Comparison between several state-of-the-art MPI correction algorithms and the three developed methods on the real datasets S_3, S_4 and S_5. Each row reports the depth MAE (7.4) obtained applying the corresponding method. The values indicated in brackets refer to the MAEs after the bilateral filtering of the corresponding depth maps. This comparison is derived from the work of Agresti *et al.* [20].

In literature and in life we ultimately pursue, not conclusions, but beginnings.

Sam Tanenhaus

8 Conclusion and Future Work

The objective of this work was the development of a technique for backscattering vector estimation given the raw ToF data acquired by the camera. We started from the idea of using a neural network architecture to learn the typical reflection structure of the light in a real environment and use it as strong prior to optimize the backscattering vector estimation. We proposed a deep leaning approach based on two models. A predictive model which takes in input the raw data and produces in output the most likely representation of the corresponding backscattering vector, and a generative model that converts the compressed representation in the final output estimation. We focused on the assumption that the MPI effect is generated only by two specular reflections of the light inside the scene. We proceeded developing three methods, working either at single pixel-level and at local level. The last method, called BVE_2SRL, makes use of the spatial correlation on the input data to improve the final prediction and turns out to be the best performing one.

Experimental results confirm the effectiveness of the proposed approach showing performance comparable to other state-of-the-art algorithms for MPI correction. On the three real datasets, it is outperformed only by the unsupervised domain adaptation technique proposed by Agresti *et al.* [20]. This is highly encouraging, since we achieved these results in the sub-optimal two specular reflections assumption and optimizing our network only on synthetic data. The network during the training sees only random synthetic data, obtained through a very simple simulation process that do not consider many real-world phenomenons. The fact that the unsupervised domain adaptation technique performs better than our approach in the datasets S_4 and S_5 is not surprising since it has been optimized on similar real data without any restriction.

We think the proposed approach has a great potential and that relaxing some restrictive simplifying assumptions it would be able to reach very high levels of accuracy. In particular, the generative model can be substituted by a more sophisticated deep learning generative algorithm, such as a Variational Auto Encoder (VAE) or a Generative Adversarial Network

(GAN). In this way, the best latent representation is learned from data and the proposed approach can account also for diffuse reflections or higher order specular reflections. Removing the two specular reflections assumption, we expect to be able to produce a more realistic final estimation and consequently to improve the MPI correction capabilities.

One fundamental aspect to consider in order to proceed in this direction is the availability of meaningful training data containing a large number of realistic MPI cases. One first possibility is to adopt more advanced ToF data simulation techniques, which model physical effects that in our work have been neglected and use a more accurate model for the noise. To this end, synthetic transient ground truth can be generated trough the transient render engine developed by Jarabo *et al.* [27], or we can use the precompute transient scenes provided by Guo *et al.* in the FLAT dataset [18]. Then, the actual ToF acquisitions can be simulated using the ToF Explorer simulator realized by Sony EuTEC starting from the work of Meister *et al.* [40]. The software is able to faithfully reproduce ToF acquisition issues like the shot and thermal noise, the read-out noise, artefacts due to lens effects, mixed pixels and specially the multi-path interference effect. The second possibility is to use some domain adaptation technique to fine tune the network on real ToF data in an unsupervised manner.

Another idea to improve the final prediction is to provide to our predictive model more informations about the noise distribution in the input data. A roughly estimation of the noise level in a ToF acquisition can be derived from equation (2.34). With this additional information, during the training the network should learn to give more relevance to less noisy data, finding an optimal strategy to manage them.

Finally, it is possible to extend the proposed idea in order to take advantages also form the temporal correlation. Since time-of-flight cameras acquire data in real-time, the temporal correlation across subsequent frames can be exploited to improve the final prediction using some recurrent neural network techniques.

A
Datasets Processing Algorithms

Algorithm A.1 Extract depth map from transient scene.

```
Input:
  - Transient scene $\mathcal{X} \in \mathbb{R}^{W \times H \times N}$ indicating the amount of light returned to the camera sensor from a certain distance
  - Depth resolution $d_{res}$ of the backscattering vector
Output:
  - Corresponding depth map $\mathcal{D} \in \mathbb{R}^{W \times H}$
  function TRANS2DEPTH($\mathcal{X}, d_{res}$)
    for all $u \in \{0, \dots, W\}, v \in \{0, \cdots, H\}$ do
      $n \leftarrow \texttt{argmin}(n' : \mathcal{X}_{uvn} > c \cdot \texttt{max}(\mathcal{X}_{uv}))$      ▷ Get index of first non-zero element
      $\mathcal{D}_{uv} \leftarrow n \cdot d_{res}$      ▷ Compute the corresponding depth
    end for
  end function
```

Algorithm A.2 Depth map computation using the standard ToF technique.

```
Input:
  - Tensor of raw data $\mathcal{V} \in \mathbb{C}^{W \times H \times M}$ acquired by the camera.
  - Set of modulation frequencies used to acquire the data $\{f_1, \dots, f_M\}$
Output:
  - Corresponding depth map $\mathcal{D} \in \mathbb{R}^{W \times H}$
  function ToFDepth($\mathcal{V}, \{f_1, \dots, f_M\}$)
    $f_H = \texttt{max}(f_1, \dots, f_M)$
    for all $u \in \{0, \dots, W\}, v \in \{0, \dots, H\}$ do
      $m_H \leftarrow$ Raw data from $\mathcal{V}_{uv}$ corresponding to frequency $f_H$
      $\varphi_H \leftarrow$ Unwrapped phase $Arg(m_H)$
      $\mathcal{D}_{uv} \leftarrow \frac{c\,\varphi_H}{4\pi f_H}$
    end for
  end function
```

Algorithm A.3 Simulate training dataset T_{ISRP}

```
Input:
  - Size of dataset $M \in \mathbb{N}$
  - Spatial resolution $W \times H$ and number of discretization steps $N$
  - Depth resolution $d_{res}$ of the backscattering vector
  - Maximum $x_{max}$ backscattering amplitude value
Output:
  - Dataset $T_{\text{ISRP}} = \{(\mathcal{V}_i, \mathcal{X}_i, \mathcal{D}_i)\}_{i=0}^{M-1}$
  function SimulateTisrp($M, W, H, N, d_{res}, x_{max}$)
    for all $i \in \{0, \cdots, M\}$ do
      $\mathcal{X}_i \leftarrow \texttt{zeros}(W, H, N)$                ▷ Generate all-zero transient scene
      for all $u \in \{0, \dots, W\}, v \in \{0, \dots, H\}$ do
        $n \leftarrow \texttt{uniform}(0, N)$                          ▷ Sample random first peak index
        $x_n \leftarrow \texttt{uniform}(x_{min}, x_{max})$            ▷ Sample random first peak amplitude
        $\mathcal{X}_{iuvn} \leftarrow x_n$                            ▷ Set amplitude of the first peak
      end for
      $\mathcal{D}_i \leftarrow$ Trans2Depth$(\mathcal{X}_i, d_{res})$   ▷ Compute depth map
      $\mathcal{V}_i = \Phi\, \mathcal{X}_i$                           ▷ Simulate acquired raw ToF data
    end for
  end function
```

Algorithm A.4 Simulate training dataset T_{2SRP}

```
Input:
  - Size of dataset $M \in \mathbb{N}$
  - Spatial resolution $W \times H$ and number of discretization steps $N$
  - Maximum offset $O > 0$ between first and second peak indices
  - Maximum scale factor $0 < S < 1$ between the first and the second peak amplitudes
  - Depth resolution $d_{res}$ of the backscattering vector
  - Maximum $x_{max}$ backscattering amplitude value
  - Multi-path interference probability $0 \leq P_{MPI} \leq 1$
Output:
  - Dataset $T_{\text{2SRP}} = \{(\mathcal{V}_i, \mathcal{X}_i, \mathcal{D}_i)\}_{i=0}^{M-1}$
  function SimulateT2srp($M, W, H, N, O, S, d_{res}, x_{max}, P_{MPI}$)
    for all $i \in \{0, \cdots, M\}$ do
      $\mathcal{X}_i \leftarrow \texttt{zeros}(W, H, N)$                ▷ Generate all-zero transient scene
      for all $u \in \{0, \ldots, W\}, v \in \{0, \ldots, H\}$ do
        $n_1 \leftarrow \texttt{uniform}(0, N)$                         ▷ Sample random first peak index
        $x_{n_1} \leftarrow \texttt{uniform}(x_{min}, x_{max})$         ▷ Sample random first peak amplitude
        $\mathcal{X}_{iuvn_1} \leftarrow x_{n_1}$                       ▷ Set amplitude of the first peak
        if $\texttt{uniform}(0, 1) < P_{MPI}$ then                      ▷ With probability $P_{MPI}$ add the second peak
          $o \leftarrow \texttt{uniform}(0, O)$                         ▷ Sample random offset for second peak index
          $s \leftarrow \texttt{uniform}(0, S)$                         ▷ Sample random scale factor for second peak amplitude
          $n_2 \leftarrow n_1 + o$                                      ▷ Compute second peak index
          $x_{n_2} \leftarrow x_{n_1} \cdot s$                          ▷ Compute second peak amplitude
          $\mathcal{X}_{iuvn_2} \leftarrow x_{n_2}$                     ▷ Set amplitude of the second peak
        end if
      end for
      $\mathcal{D}_i \leftarrow \text{Trans2Depth}\big(\mathcal{X}_i, d_{res}\big)$    ▷ Compute depth map
      $\mathcal{V}_i = \Phi\, \mathcal{X}_i$                            ▷ Simulate acquired raw ToF data
    end for
  end function
```

Algorithm A.5 Equalize depth distribution

```
Input:
  - Depth maps $\{\mathcal{D}_i \in \mathbb{R}^{W \times H}\}_{i=0}^{M-1}$, to be equalized
  - Minimum $d_{min}^{eq}$ and maximum $d_{max}^{eq}$ equalized depth values
Output:
  - Depth maps $\{\mathcal{D}_i^{eq} \in \mathbb{R}^{W \times H}\}_{i=0}^{M-1}$ equalized with uniform depth distribution
  function DepthEqualization($\{\mathcal{D}_i\}_{i=0}^{M-1}, d_{min}^{eq}, d_{max}^{eq}$)
    for all $i \in \{0, \cdots, M\}$ do
      $d_i \leftarrow \texttt{mean}(\mathcal{D}_i)$                              ▷ Compute average depth
    end for
    $pdf_d(k) \leftarrow P[d_i = k] = \frac{\text{No. patches such that } d_i = k}{M}$          ▷ Compute average depth pdf
    $cdf_d(k) \leftarrow \sum_{j=0}^{k} pdf_d(j)$                              ▷ Compute average depth cdf
    for all $i \in \{0, \cdots, M\}$ do
      $d_i^{eq} \leftarrow d_{min}^{eq} + (d_{max}^{eq} - d_{min}^{eq}) \cdot cdf_d(d_i)$          ▷ Equalize average depth
      $\Delta d \leftarrow d_i^{eq} - d_i$                                   ▷ Compute constant offset
      $\mathcal{D}_i^{eq} \leftarrow \mathcal{D}_i + \Delta d$                  ▷ Add constant offset to the whole depth patch
    end for
  end function
```

Algorithm A.6 Simulate training dataset T_{2SRL}

```
Input:
  - Set of depth maps $\{\mathcal{D}_i \in \mathbb{R}^{W \times H}\}_{i=0}^{M-1}$, to be used as source of spatial correlation
  - Number of discretization steps $N$
  - Maximum offset $O > 0$ between first and second peak indices
  - Maximum scale factor $0 < S < 1$ between the first and the second peak amplitudes
  - Depth resolution $d_{res}$ of the backscattering vector
  - Maximum $x_{max}$ backscattering amplitude value
  - Multi-path interference probability $0 \leq P_{MPI} \leq 1$
Output:
  - Dataset $T_{\text{2SRL}} = \{(\mathcal{V}_i, \mathcal{X}_i, \mathcal{D}_i)\}_{i=0}^{M-1}$
  function SimulateT2srl($\{\mathcal{D}_i\}_{i=0}^{M-1}, N, O, S, d_{res}, x_{max}, P_{MPI}$)
    for all $i \in \{0, \cdots, M\}$ do
      $\mathcal{X}_i \leftarrow \texttt{zeros}(W, H, N)$                ▷ Generate all-zero transient scene
      $o \leftarrow \texttt{uniform}(0, O)$                ▷ Sample random offset for second peak index
      $s \leftarrow \texttt{uniform}(0, S)$                ▷ Sample random scale factor for second peak amplitude
      for all $u \in \{0, \ldots, W\}, v \in \{0, \ldots, H\}$ do
        $n_1 \leftarrow \left\lfloor \frac{\mathcal{D}_{iuv}}{d_{red}} \right\rfloor$                ▷ Get first peak index from input depth map
        $x_{n_1} \leftarrow \texttt{uniform}(x_{min}, x_{max})$                ▷ Sample random first peak amplitude
        $\mathcal{X}_{iuvn_1} \leftarrow x_{n_1}$                ▷ Set amplitude of the first peak
        if $\texttt{uniform}(0, 1) < P_{MPI}$ then                ▷ With probability $P_{MPI}$ add the second peak
          $o_p \leftarrow \texttt{normal}(0, \sigma_o)$                ▷ Sample pixel-dependent noise for offset $o$
          $s_p \leftarrow \texttt{normal}(1, \sigma_s)$                ▷ Sample pixel-dependent noise for scale factor $s$
          $n_2 \leftarrow n_1 + o + o_p$                ▷ Compute second peak index
          $x_{n_2} \leftarrow x_{n_1} \cdot s \cdot s_p$                ▷ Compute second peak amplitude
          $\mathcal{X}_{iuvn_2} \leftarrow x_{n_2}$                ▷ Set amplitude of the second peak
        end if
      end for
      $\mathcal{D}_i \leftarrow \text{Trans2Depth}\big(\mathcal{X}_i, d_{res}\big)$                ▷ Compute depth map
      $\mathcal{V}_i = \Phi\, \mathcal{X}_i$                ▷ Simulate acquired raw ToF data
    end for
  end function
```

References

[1] M. Frank, M. Plaue, H. Rapp, U. Köthe, B. Jähne, and F. A. Hamprecht, "Theoretical and experimental error analysis of continuous-wave time-of-flight range cameras," *Optical Engineering*, vol. 48, no. 1, p. 013602, 2009.

[2] R. Whyte, L. Streeter, M. J. Cree, and A. A. Dorrington, "Review of methods for resolving multi-path interference in time-of-flight range cameras," in *SENSORS, 2014 IEEE.* IEEE, 2014, pp. 629–632.

[3] S. Foix, G. Alenya, and C. Torras, "Lock-in time-of-flight (tof) cameras: A survey," *IEEE Sensors Journal*, vol. 11, no. 9, pp. 1917–1926, 2011.

[4] D. Lefloch, R. Nair, F. Lenzen, H. Schäfer, L. Streeter, M. J. Cree, R. Koch, and A. Kolb, "Technical foundation and calibration methods for time-of-flight cameras," in *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications.* Springer, 2013, pp. 3–24.

[5] R. Lange, P. Seitz, A. Biber, and S. C. Lauxtermann, "Demodulation pixels in ccd and cmos technologies for time-of-flight ranging," in *Sensors and camera systems for scientific, industrial, and digital photography applications*, vol. 3965. International Society for Optics and Photonics, 2000, pp. 177–188.

[6] M. Gupta, S. K. Nayar, M. B. Hullin, and J. Martin, "Phasor imaging: A generalization of correlation-based time-of-flight imaging," *ACM Transactions on Graphics (ToG)*, vol. 34, no. 5, p. 156, 2015.

[7] D. Freedman, Y. Smolin, E. Krupka, I. Leichter, and M. Schmidt, "Sra: Fast removal of general multipath for tof sensors," in *European Conference on Computer Vision.* Springer, 2014, pp. 234–249.

[8] X. Li and X.-G. Xia, "A fast robust chinese remainder theorem based phase unwrapping algorithm," *IEEE Signal Processing Letters*, vol. 15, pp. 665–668, 2008.

[9] S. Fuchs, "Multipath interference compensation in time-of-flight camera images," in *2010 20th International Conference on Pattern Recognition.* IEEE, 2010, pp. 3583–3586.

[10] S. Fuchs, M. Suppa, and O. Hellwich, "Compensation for multipath in tof camera measurements supported by photometric calibration and environment integration," in *International Conference on Computer Vision Systems.* Springer, 2013, pp. 31–41.

[11] D. Jiménez, D. Pizarro, M. Mazo, and S. Palazuelos, "Modeling and correction of multipath interference in time of flight cameras," *Image and Vision Computing*, vol. 32, no. 1, pp. 1–13, 2014.

[12] A. A. Dorrington, J. P. Godbaz, M. J. Cree, A. D. Payne, and L. V. Streeter, "Separating true range measurements from multi-path and scattering interference in commercial range cameras," in *Three-Dimensional Imaging, Interaction, and Measurement*, vol. 7864. International Society for Optics and Photonics, 2011, p. 786404.

[13] J. P. Godbaz, "Ameliorating systematic errors in full-field amcw lidar," Ph.D. dissertation, University of Waikato, 2012.

[14] A. Kirmani, A. Benedetti, and P. A. Chou, "Spumic: Simultaneous phase unwrapping and multipath interference cancellation in time-of-flight cameras using spectral methods," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2013, pp. 1–6.

[15] A. Bhandari, A. Kadambi, R. Whyte, C. Barsi, M. Feigin, A. Dorrington, and R. Raskar, "Resolving multipath interference in time-of-flight imaging via modulation frequency diversity and sparse regularization," *Optics letters*, vol. 39, no. 6, pp. 1705–1708, 2014.

[16] C. Peters, J. Klein, M. B. Hullin, and R. Klein, "Solving trigonometric moment problems for fast transient imaging," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 220, 2015.

[17] J. Marco, Q. Hernandez, A. Munoz, Y. Dong, A. Jarabo, M. H. Kim, X. Tong, and D. Gutierrez, "Deeptof: off-the-shelf real-time correction of multipath interference in time-of-flight imaging," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 6, p. 219, 2017.

[18] Q. Guo, I. Frosio, O. Gallo, T. Zickler, and J. Kautz, "Tackling 3d tof artifacts through learning and the flat dataset," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 368–383.

[19] G. Agresti and P. Zanuttigh, "Deep learning for multi-path error removal in tof sensors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[20] G. Agresti, H. Schaefer, P. Sartor, and P. Zanuttigh, "Unsupervised domain adaptation for tof data denoising with adversarial learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5584–5593.

[21] S. Á. Guðmundsson, H. Aanaes, and R. Larsen, “Fusion of stereo vision and time-of-flight imaging for improved 3d estimation,” *International Journal on Intelligent Systems Technologies and Applications (IJISTA)*, vol. 5, no. 3/4, pp. 425–433, 2008.

[22] G. Marin, P. Zanuttigh, and S. Mattoccia, “Reliable fusion of tof and stereo depth driven by confidence measures,” in *European Conference on Computer Vision*. Springer, 2016, pp. 386–401.

[23] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar, “Fast separation of direct and global components of a scene using high frequency illumination,” in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 935–944.

[24] A. Kadambi, R. Whyte, A. Bhandari, L. Streeter, C. Barsi, A. Dorrington, and R. Raskar, “Coded time of flight cameras: sparse deconvolution to address multipath interference and recover time profiles,” *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, p. 167, 2013.

[25] R. Whyte, L. Streeter, M. J. Cree, and A. A. Dorrington, “Resolving multiple propagation paths in time of flight range cameras using direct and global separation methods,” *Optical Engineering*, vol. 54, no. 11, p. 113109, 2015.

[26] G. Agresti and P. Zanuttigh, “Combination of spatially-modulated tof and structured light for mpi-free depth estimation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[27] A. Jarabo, J. Marco, A. Muñoz, R. Buisan, W. Jarosz, and D. Gutierrez, “A framework for transient rendering,” *ACM Transactions on Graphics (ToG)*, vol. 33, no. 6, p. 177, 2014.

[28] A. Kirmani, T. Hutchison, J. Davis, and R. Raskar, “Looking around the corner using transient imaging,” in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 159–166.

[29] D.-I. B. Hagebeuker and P. Marketing, “A 3d time of flight camera for object detection,” *PMD Technologies GmbH, Siegen*, 2007.

[30] D. Wu, A. Velten, M. O’toole, B. Masia, A. Agrawal, Q. Dai, and R. Raskar, “Decomposing global light transport using time of flight imaging,” *International journal of computer vision*, vol. 107, no. 2, pp. 123–138, 2014.

[31] S. Su, F. Heide, R. Swanson, J. Klein, C. Callenberg, M. Hullin, and W. Heidrich, “Material classification using raw time-of-flight measurements,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3503–3511.

[32] A. Jarabo, B. Masia, J. Marco, and D. Gutierrez, “Recent advances in transient imaging: A computer graphics and vision perspective,” *Visual Informatics*, vol. 1, no. 1, pp. 65–79, 2017.

[33] D. L. Donoho *et al.*, “Compressed sensing,” *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[34] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*. Citeseer, 2017, vol. 1.

[35] L. Pardo, *Statistical inference based on divergence measures*. Chapman and Hall/CRC, 2018.

[36] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein gan,” *arXiv preprint arXiv:1701.07875*, 2017.

[37] F. Wang and L. J. Guibas, “Supervised earth mover’s distance learning and its computer vision applications,” in *European Conference on Computer Vision*. Springer, 2012, pp. 442–455.

[38] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[39] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images.” in *Iccv*, vol. 98, no. 1, 1998, p. 2.

[40] S. Meister, R. Nair, and D. Kondermann, “Simulation of time-of-flight sensors using global illumination,” 2013.