# Lightweight Deep Learning Architecture for MPI Correction and Transient Reconstruction

Adriano Simonetto, Gianluca Agresti, Pietro Zanuttigh and Henrik Schäfer

*Abstract*—**Indirect Time-of-Flight cameras (iToF) are low-cost devices that provide depth images at an interactive frame rate. However, they are affected by different error sources, with the spotlight taken by Multi-Path Interference (MPI), a key challenge for this technology. Common data-driven approaches tend to focus on a direct estimation of the output depth values, ignoring the underlying transient propagation of the light in the scene. In this work instead, we propose a very compact architecture, leveraging on the direct-global subdivision of transient information for the removal of MPI and for the reconstruction of the transient information itself. The proposed model reaches state-of-the-art MPI correction performances both on synthetic and real data and proves to be very competitive also at extreme levels of noise; at the same time, it also makes a step towards reconstructing transient information from multi-frequency iToF data.**

## I. Introduction

The demand for more accurate and reliable range imaging devices has seen a constant rise over the years. Their applications are widespread ranging from autonomous driving [1], [2] to augmented reality [3], 3D reconstruction [4], [5] and even landing on planetary bodies [6]. The working principles are different for the various types of sensors, but the main objective remains the same: retrieving the distance information between the camera and the target object. Some of the most common technologies are stereo imaging [7], where the depth information is retrieved from a couple of RGB cameras at a fixed distance, Time-of-Flight (ToF) based devices [8], e.g. LIDARs [9] or matrix ToF sensors, and structured light scanners [10], that rely on light patterns. In this work we will focus our attention on ToF based technologies, more specifically on indirect Time-of-Flight (iToF) cameras. A direct Time-of-Flight (dToF) device sends an impulse of light towards the scene, measures the travel time of the impulse and computes the depth information from that. An iToF camera instead sends a modulated light signal and correlates the reflected signal with the sensor modulation signal; from these measures the distance is retrieved. iToF-based cameras are quite accurate, have a good spatial resolution and are nowadays sold at consumer level in some of the most recent mobile phones [11]. This technology however also has a significant drawback, which is intrinsic to its basic operating principle and is called Multi-Path Interference (MPI). This error source typically produces an overestimation of the depth, which is strictly linked to the scene geometry and that has been widely studied in the literature [12], [13], [14], [15]. Some early single frequency approaches such as [16], [17] proposed an algorithmic solution for the problem, but also had to set some unrealistic assumptions (e.g. knowing the scene structure) in order to make it tractable. Following attempts highlighted the need for multiple iToF acquisitions at different frequencies in order to better deal with MPI [18], [19], and also linked the iToF and dToF domains showing that it is possible to go from dToF to iToF with a simple linear model [12]. The true turning point however, came when deep learning started being applied to the field. The first deep learning architectures [20], [13] improved on the previous works but were still quite complex models that at the same time did not perform well on real data. The main issue is that the acquisition of real iToF data with matching depth ground truth is a challenging task, and synthetic images are therefore the main training tool. This problem has been tackled in [21] where an Unsupervised Domain Adaptation (UDA) approach was proposed showing that it is possible to improve the model generalization without the need for real ground truth; in [22] they expanded the method by considering different domain adaptation scenarios. Recently, a couple of data-driven approaches exploiting the relationship between iToF and dToF information [15], [23] have been proposed.

In this work, we introduce a novel modular deep learning approach that leverages on the dual nature of transient information for MPI correction and for the estimation of dToF data. The model is composed of three parts, the first one needed for dealing with temporal noise sources with zero mean such as shot noise, the second built for MPI denoising and the final module instead for the reconstruction of the transient representation. We propose in particular two architectures. The first, *SD*, has only $23k$ parameters but reaches state-of-the-art performance on synthetic data, beating the network from [14] which is 7 times its size. It also performs on par with state of the art approaches, e.g. [22], on real data without any need for unsupervised domain adaptation, using only $\frac{1}{7}$th of its parameters. Furthermore, it also outperforms by a noticeable margin [15], which has a similar size. The second architecture, *D*, falls instead only a little behind in performance, but is extremely lightweight, i.e. it has only $3k$ parameters and still beats on real datasets heavier approaches such as [15] (7 times its size) and [14] (50 times). An additional contribution of this paper is the introduction of the *Walls* dataset; it is a novel transient dataset based on simple geometries that is used for training both the modules for MPI denoising and the one for transient reconstruction.

A. Simonetto and P. Zanuttigh are with the Department of Information Engineering, University of Padova, Padova, Italy, e-mail: adriano.simonetto@phd.unipd.it, zanuttigh@dei.unipd.it

G. Agresti and H. Schäfer are with the R&D Center Europe Stuttgart Laboratory 1, Stuttgart, Germany, e-mail: Gianluca.Agresti@sony.com, Henrik.Schaefer@sony.com

The rest of this paper is structured as follows: Section II gives an overview of the current works on MPI denoising and transient reconstruction; Section III explains the working principle of iToF cameras and introduces the notation, while Section IV instead shows the proposed architecture and describes its modules; in Section V we discuss the employed datasets, focusing especially on the one that we introduce; finally, in Section VI we give an in depth quantitative and qualitative comparison with some state-of-the-art approaches, and in Section VII we draw our conclusions and describe some future developments.

## II. Related Works

As remarked in Section I, Multi-Path Interference is a non-zero mean error source that is intrinsic to the iToF technology, and at the same time one of its key limitations. The approaches that tackle MPI correction can be generally divided into two groups, single-frequency and multi-frequency ones. Those belonging to the first group such as [16], [17] and [24] exploit a reflection model together with the spatial information provided by the MPI-corrupted image for their solution. Jimenez et al [24] for example, proposed an iterative optimization algorithm based on the assumption that all scene surfaces are perfectly Lambertian. Early multi-frequency approaches show similar constraints. In [12], Freedman et al. introduced the relationship between iToF measurements and the transient behaviour of light extending the problem to the case of $K$ interfering rays. They then proposed an algorithm for MPI correction treating it as an $L_1$ optimization problem. Bhandari et al. [19], adopted similar assumptions but offered instead a non-iterative solution using Vandermonde matrices.

The restrictions of these models and the unrealistic amount of input frequencies required for the solutions lead to a rapid rise in popularity of deep learning based approaches. Marco et al. [20] proposed an encoder-decoder architecture with a split training approach: the encoder was trained on unlabelled real data, and the decoder on the synthetic dataset they introduced. Su et al. [13] proposed a multi-scale network working in combination with a discriminator module. The network has been trained combining three losses, one regarding the reconstruction performance, one enforcing a smoothness constraint, and an adversarial one. The architecture has then been tested on the synthetic dataset they introduced. Another dataset was introduced by Guo et al [25], together with a deep learning model able to tackle both MPI and shot noise, and that is able to handle dynamic scenes too. Their model consists of an encoder-decoder architecture combined with a kernel prediction network used to tackle the shot noise. Agresti et al. [14] observed that the information regarding the structure of the scene is particularly important for MPI correction and, in order to have a simple network with about $150k$ parameters, they built it with two branches, one capturing the details and the other focusing on the high level geometry of the image. A similar idea was employed in [26] where a pyramid network observes the MPI structure at multiple resolutions, putting then the information together for the final prediction. In [21] the authors aimed at filling the gap between prediction on synthetic and real data, using an unsupervised domain adaptation approach. They took the model from [14] and trained it as a GAN on unlabelled real data, clearly outperforming the original approach. The idea was later expanded by the same authors in [22], where they examined the possibility of performing domain adaptation also at input and feature level. More recently, a few works tackled MPI correction making use of a more or less refined version of the light transport model. Barragan et al. [23] worked on the Fourier domain, using a U-Net architecture that takes a two-frequency input and predicts MPI corrupted data at several frequencies. They then compute the inverse Fourier transform on the output data, perform some filtering and get the depth prediction using a peak finding algorithm. The method shows good shot noise and MPI denoising capabilities but is quite heavy, with around $1.8M$ parameters, differently from the architecture proposed here, that works in the time domain and is much lighter. In [15] we encoded the transient information with two peaks, the first one corresponding to the shortest light path, and the other to a weighted average of all other reflected light components. Even with such a rough approximation, we showed promising results on MPI correction on real data, all while using a network with a $3 \times 3$ receptive field. The method we propose now starts from this previous work, as both focus more on the information on the transient dimension for the reconstruction rather than that on the spatial one. Apart from this high level similarity, the approach proposed here differs from the previous one in several aspects. First of all, we build a different learning architecture, which directly reflects the dual structure of transient information and at the same time employs a module that helps dealing with shot noise. We also introduce a more accurate encoding of the light transient information, and construct a specific network for its prediction. A more thorough comparison between the proposed method and [15] can be found in the Appendix VII

In the literature, works directly targeted at transient recovery from iToF information are very few, all focusing on strong simplifying assumption for their solutions. Heide et al. [27] used an iToF camera to recover the depth information of a scene using the light reflected by a diffuse surface. They treated transient recovery as an optimization problem, constrained their solution both regarding spatial gradients and height field and introduced an algorithm in order to solve it. Lin et al. [28], showed that the information recovered from a multi-frequency iToF camera corresponds to the Fourier transform of a transient image. They then proposed an algorithm for transient reconstruction from a high number of iToF modulation frequencies. On a different note, Liang et al. [29] devised a deep learning model for the compression of rendered transient data, an important task due to the high volume of the data and the large amount of rendering noise.

## III. iToF Model

ITоF cameras consist of an emitter and a sensor. The light sent by the emitter is a modulated signal $i(t)$, typically a sinusoidal wave with a frequency in the range of $10 - 100$ MHz, while the sensor function $s(t)$ is instead a periodic square wave with the same frequency. The iToF measurements are the result of the correlation between the reflected light signal $r(t)$ and $s(t)$,

$$v_\theta = \int_0^{T_{int}} r(t)s\Big(t + \frac{\theta}{2\pi f_m}\Big)\mathrm{d}t, \tag{1}$$

where $\theta$ is an internal phase shift, $v_\theta$ is the iToF measurement, $T_{int}$ is the integration time and $f_m$ the modulation frequency. In the ideal scenario where the reflected signal corresponds to a single light reflection, we have an analytical solution of the integral as $v_\theta = I + A \cdot \cos(\varphi + \theta)$, with $I$ the intensity of the signal, $A$ the amplitude of the sinusoid and $\varphi$ the phase delay due to the travel time. In a practical scenario 4 measurements are sufficient for recovering the 3 unknowns, and from the phase delay $\varphi$ we are then able to reconstruct the depth information $d$ from the well-known relation:

$$d = \frac{c\varphi}{4\pi f_m}, \tag{2}$$

where $c$ is the speed of light. If we take aside the intensity component, we can use the phasor notation as described in Gupta et al. [30] to represent the iToF measurement with

$$v = Ae^{i\varphi} = Ae^{i2\pi f_m \Delta t} \in \mathbb{C}, \tag{3}$$

where $\Delta t$ is the round trip time of the light signal. This notation can be used to mathematically describe the MPI phenomenon [30] in a real case where the light bounces multiple times inside a scene, meaning that the sensor will receive and integrate not one but multiple signals covering different, normally longer, paths. This effect can now be written as follows, as phasors are closed under summation:

$$v = \int_{t_{min}}^{t_{max}} x(t)e^{i2\pi f_m t}\mathrm{d}t, \tag{4}$$

with $t_{min}$ and $t_{max}$ respectively the minimum and maximum time of flights considered, and $x(t)$ the time dependent scene impulse response, also known as transient. A discretized transient can be seen in Figure 1. We can now discretize the integral in Equation (4),

$$v = \sum_{t=t_{min}}^{t_{max}} x(t)e^{i2\pi f_m t}, \tag{5}$$

consider multiple acquisition frequencies and rewrite it as

$$\boldsymbol{v} = \boldsymbol{\Phi}\boldsymbol{x}, \tag{6}$$

where $\boldsymbol{v}$ are the iToF measurements at multiple frequencies, $\boldsymbol{x}$ is the scene impulse response and $\boldsymbol{\Phi}$ the measurement model linked to the iToF camera.

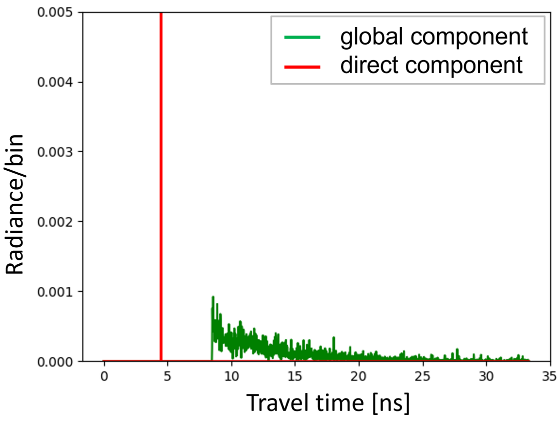


Fig. 1: A sample transient vector from a scene in the *walls* dataset. We highlighted the direct component in red and the global in green.

## IV. Method

In this section, we provide an exhaustive description of a novel method for MPI correction and transient image reconstruction. We start by describing the idea behind it and then go in detail through each of the three components of our modular architecture: the *Spatial Feature Extractor*, the *Direct Phasor Estimator* and the *Transient Reconstruction Module*. In the end of the section, we introduce the losses employed for training.

### A. Direct-Global Subdivision

Let's begin by considering the structure of a common transient vector (see the example in Figure 1); it is quite clear that it is composed of two quite distinct parts: one corresponding to the first peak, the other instead incorporating all the other incoming light rays. From now on, we will call the vector composed by the first peak alone *direct component* and will denote it with $\boldsymbol{x_d}$, while the *global component* will be composed of all the other reflections and will be represented as $\boldsymbol{x_g}$. We can now consider Equation (6) and write

$$\boldsymbol{v} = \boldsymbol{\Phi}\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{x_d} + \boldsymbol{x_g}) = \boldsymbol{v_d} + \boldsymbol{v_g}, \tag{7}$$

where we exploited the linearity of the model to extract the $\boldsymbol{v_d}$ and $\boldsymbol{v_g}$ vectors. What follows from this derivation is that the subdivision of the transient vector into direct and global components can be translated also onto the iToF domain. In practice, we now have a vector $\boldsymbol{v_d}$ which corresponds to ideal iToF measurements, the ones that would be produced by the direct peak alone, while $\boldsymbol{v_g}$ are the measurements corresponding to all reflections but the first.

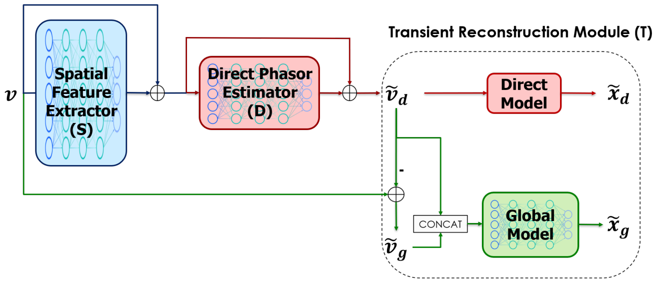

$\boldsymbol{v}$
Spatial Feature Extractor (S)
Direct Phasor Estimator (D)
Transient Reconstruction Module (T)
$\tilde{\boldsymbol{v}}_{\boldsymbol{d}}$
Direct Model
$\tilde{\boldsymbol{x}}_{\boldsymbol{d}}$
-
CONCAT
Global Model
$\tilde{\boldsymbol{x}}_{\boldsymbol{g}}$
$\tilde{\boldsymbol{v}}_{\boldsymbol{g}}$

Fig. 2: High level structure of our training architecture

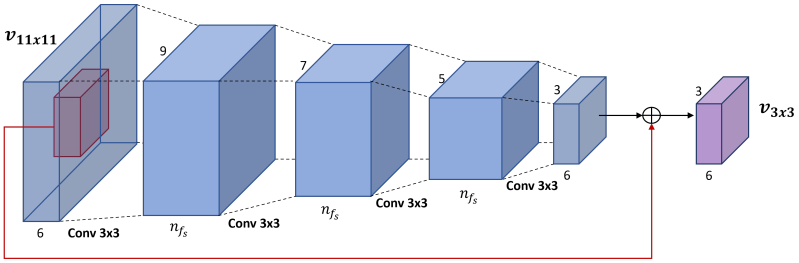


Fig. 3: Representation of the Spatial Feature Extractor module for 3 input frequencies and an input patch size of $11 \times 11$. The number of feature maps $n_{f_s}$ is equal to 32 for all experiments.

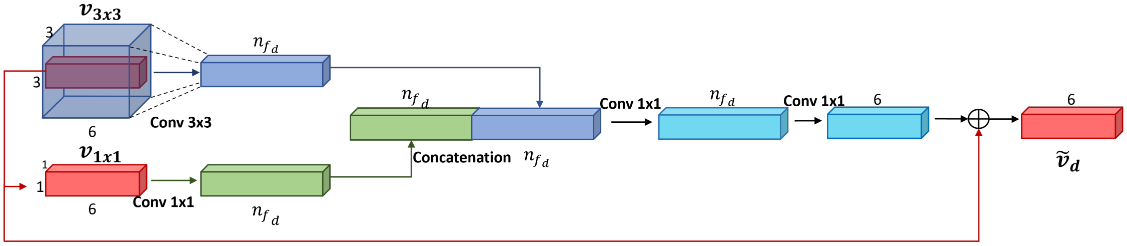


Fig. 4: Representation of the Direct Phasor Estimator module (upper part) for 3 input frequencies. The number of feature maps $n_{f_d}$ is equal to 8 when both the $S$ and $D$ models are used, and to 32 when the $D$ model is employed alone.

### *B. Deep Learning Architecture*

The different modules of our network are shown in Figure 2. As input the model takes in the real and imaginary components of the raw iToF measurements $\boldsymbol{v}$ at different modulation frequencies. First, they go through the *Spatial Feature Extractor*, which exploits the spatial information to produce an intermediate representation of the data (it proved to be very useful for handling zero-mean noise). Its output is then processed by the *Direct Phasor Estimator*, that predicts the iToF measurements corresponding to the direct component, which, subtracted from the original input, gives us also the iToF measurements corresponding to the global component. The two predictions are in the end fed to the *Transient Reconstruction Module* that has the task of reconstructing the whole transient vector. As we will see, this module is further split into the *Direct Model* which is a deterministic function computing the direct component, and the *Global Model* that instead consists of a deep learning architecture predicting the global component. For the construction of the learning model we used as a starting point the network introduced in [15], where the raw iToF input was directly mapped into an oversimplified encoded version of a transient vector, consisting of two peaks. We kept a narrow receptive field claiming that the information in the transient dimension is enough for MPI correction, but differently from [15], we introduce an intermediate training target between the input $\mathbf{v}$ and the transient prediction $\tilde{\mathbf{x}}$ (i.e., the subdivision into direct and global components). Moreover, we introduce a more complex and realistic model for the backscattering vector itself.

*1) Spatial Feature Extractor (S):* The main task of this module is providing an encoded version of spatial information to the following stages. As we will see, the *Direct Phasor Estimator* has a very narrow receptive field (i.e. $3 \times 3$), which limits its capability of managing noise sources such as shot noise. The *Spatial Feature Extractor* is a fully convolutional architecture with a $9 \times 9$ receptive field. It consists of 4 layers, each with $n_{f_s} = 32$ feature maps and a residual connection links the central $3 \times 3$ part of the input to the output. A visual representation of this network can be found in Figure 3.

*2) Direct Phasor Estimator (D):* This module estimates $\boldsymbol{v_d}$, the direct component of the raw phasor. The raw measurements coming from the *Spatial Feature Extractor* are fed to two branches with receptive field $3 \times 3$ and $1 \times 1$ respectively, whose outputs are then concatenated and used for the prediction of $\tilde{\boldsymbol{v}}_{\boldsymbol{d}}$. More in detail, as depicted in Figure 4, it takes in input both a $3 \times 3$ patch and its central pixel; they go through a convolutional layer with an output of size $1 \times 1$ and are then concatenated. The information is then processed by two other convolutional layers before producing the $\tilde{\boldsymbol{v}}_{\boldsymbol{d}}$ prediction. Each convolutional layer has $n_{f_d}$ feature maps and there is a residual connection between input and output. From the prediction of $\tilde{\boldsymbol{v}}_{\boldsymbol{d}}$ we then compute the corresponding depth for each of the modulation frequencies, using the smallest frequency for solving ambiguity range uncertainty on the higher ones. The output depth maps are then passed through a bilateral filter and the final depth prediction will be the pixel-wise minimum of the output depths. The reason for this is that the MPI, that is the major cause of error, leads to an overestimation of the distance. Considering Equation (7) we can then retrieve also the iToF measurements corresponding to the global component by simply subtracting the direct component, i.e., $\tilde{\boldsymbol{v}}_{\boldsymbol{g}} = \boldsymbol{v} - \tilde{\boldsymbol{v}}_{\boldsymbol{d}}$. Notice that this module is tackling the MPI removal task, as if the direct-global subdivision is successful, we are able to recover an MPI-free estimate of our input from the $\tilde{\boldsymbol{v}}_{\boldsymbol{d}}$ component. The number of feature maps is set to $n_{f_d} = 8$ when the $S$ and $D$ models are used together and to $n_{f_d} = 32$ when the $D$ model is used alone.

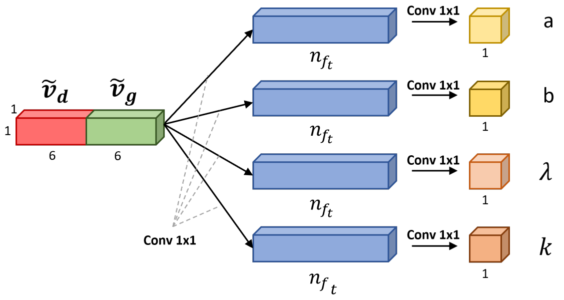


Fig. 5: Representation of the *Global Model*. The number of feature maps $n_{f_t}$ has been set to 32 for all experiments.

*3) Transient Reconstruction Module (T):* Retrieving the transient information from iToF leads to some serious challenges, not only linked to the difficulty of the task itself, but also to the dimensionality of our output. As remarked in Section I, we want to map the raw iToF measurements, that correspond to a handful of values, into a vector with thousands of entries. The complexity of the matter makes an encoding of the ground truth a necessity. Since the one proposed in [15] is way too simplistic, and the one by Liang et al. [31] computationally heavy, we propose a novel approximation of the transient vector $\boldsymbol{x_g}$ with just 6 parameters, 2 needed for the direct component, and the other 4 for the global. Therefore, the *Transient Reconstruction Module* is further split into two components, the *Direct Model* that takes care of the reconstruction of the direct component and the *Global Model*, which instead predicts the global component.

*a) Direct Model:* Similarly to [15], each direct component $\boldsymbol{x_d}$ gets encoded by its magnitude $E_d$ and time position $t_d$. As a matter of fact, no learnable parameters are needed for the prediction of the direct component $\boldsymbol{x_d}$, since the time position $t_d$ is directly proportional to the phase $\varphi_d$ through Equation (3), and can be retrieved directly from $\tilde{\boldsymbol{v}}_{\boldsymbol{d}}$. At the same time, the magnitude of the first peak $E_d$ is strictly related to the amplitude of the raw iToF measurements of the direct component; this is true since in the case of a single peak, the magnitude is the value of the peak itself, while the amplitude $A_d$ can be written as follows,

$$\begin{aligned} A_d &= \frac{1}{2}\sqrt{\boldsymbol{v_{d,\Re}}^2+\boldsymbol{v_{d,\Im}}^2}\\ &= \frac{1}{2}\sqrt{\left(\sum_{t=0}^{T}\Phi_{\Re,t}\boldsymbol{x_t}\right)^2+\left(\sum_{t=0}^{T}\Phi_{\Im,t}\boldsymbol{x_t}\right)^2} = \\ &= \frac{1}{2}\sqrt{\left(\Phi_{\Re,t_d}x_{t_d}\right)^2+\left(\Phi_{\Im,t_d}x_{t_d}\right)^2} = \\ &= \frac{1}{2}x_{t_d} = \frac{1}{2}E_d, \end{aligned} \tag{8}$$

where we used the Pythagorean identity and with the fact that only one element of the sum is non-zero (the one at time index $t_d$). $\boldsymbol{v_{d,\Re}}$ and $\boldsymbol{v_{d,\Im}}$ are the real and imaginary components following the phasor notation in Equation (3).

*b) Global Model:* For the encoding of $\boldsymbol{x_g}$ we chose instead the following parametric function $\tilde{\boldsymbol{x}}_{\boldsymbol{g}}(t)$ inspired by the Weibull distribution [32]

$$\tilde{\boldsymbol{x}}_{\boldsymbol{g}}(t) = \mathbf{a}(t-\mathbf{b})^{\mathbf{k}-1}\exp\left(-\frac{t-\mathbf{b}}{\boldsymbol{\lambda}}\right)^{\mathbf{k}}, \tag{9}$$

where $t$ ranges from 0 to $T$ (the maximum acceptable travel time), $\mathbf{a}$ takes care of the scale, $\mathbf{b}$ of the shift, and $\mathbf{k}$ and $\boldsymbol{\lambda}$ of the shape. For the choice of this function we took inspiration from the topic of multipath interference related to radio signals, where distributions such as the Rayleigh or the Weibull are usually employed [32]. In the end we decided to employ the Weibull distribution since it is a generalization of the Rayleigh and shows a good resemblance with common shapes of transient vectors. Predicting the parameters of the global component $\tilde{\boldsymbol{x}}_{\boldsymbol{g}}$ expressed in Equation (9) from $\tilde{\boldsymbol{v}}_{\boldsymbol{g}}$ and $\tilde{\boldsymbol{v}}_{\boldsymbol{d}}$ is a quite complex task, which is handled by an additional deep learning architecture. The *Global Model* is composed of 4 parallel branches with a $1\times1$ receptive field, each predicting one of the 4 parameters of the parametric function. Each branch is composed of a stack of 2 convolutional layers with a total of 32 feature maps. It takes $\tilde{\boldsymbol{v}}_{\boldsymbol{g}}$ and $\tilde{\boldsymbol{v}}_{\boldsymbol{d}}$ in input, estimates from it the 4 parameters of the function described in Equation (9), and finally compares it to the ground truth $\boldsymbol{x_g}$. The *Global Model* can be seen in Figure 5. The proposed model for global prediction is a clear improvement w.r.t. the competitors as the Weibull function provides a much better fitting of a distribution such as the one in Figure 1 than the single peak prediction of [15], which compacts all the information in a single bin.
Combining together the outputs of the *Direct Model* and of the *Global Model*, we obtain an estimate of the transient vector. Notice that while the proposed reconstruction of the global component of light is more advanced than the ones of previous works, still it only estimates a single Weibull function and therefore assumes any secondary reflection to come from a single surface. While this is a quite coarse simplification, it can still provide a sufficiently good reconstruction for simple tasks such as tracking NLOS objects or material estimation.

### C. Training Targets

The losses used for training our architecture are the Mean Absolute Error (MAE), and the Earth Mover's Distance (EMD) [33]. The *Direct Phasor Estimator* uses as guidance a simple MAE on the target values, while the EMD guides the training of the *Global Model*. The ground truth $\boldsymbol{v_d}$ can only be retrieved from the transient information and for this reason it is not available when using common iToF datasets. For this reason, we employed two different training methodologies according to the input data:

1) When the transient data is available we can directly compute the loss between the ground truth $\boldsymbol{v_d}$ and our prediction $\tilde{\boldsymbol{v}}_{\boldsymbol{d}}$ as

$$\mathcal{L}_{MAE_{v_d}} = \mathbb{E}\left[\|\boldsymbol{v_d}-\tilde{\boldsymbol{v}}_{\boldsymbol{d}}\|_1\right]. \tag{10}$$

2) When instead the dataset only offers the depth ground truth, we are unable to recover $\boldsymbol{v_d}$, but we are able to compute the ground truth phase delay $\boldsymbol{\varphi_d}$ following Equation (2). From the network prediction $\tilde{\boldsymbol{v}}_{\boldsymbol{d}}$ we can

| Dataset | # of images | Image type | Noise type | Ground truth |
|---|---|---|---|---|
| $S_1$ [14] | 54 | Synthetic | Shot | Depth |
| $S_3$ [21] | 8 | Real | Real | Depth |
| $S_4$ [14] | 8 | Real | Real | Depth |
| $S_5$ [21] | 8 | Real | Real | Depth |
| iToF2dToF [23] | 5000 | Synthetic | Shot,read | Depth* |
| Walls[1] | 222 | Synthetic | None | Transient |

TABLE I: Comparison between the employed datasets. (*) transient available only for a small amount of data. Samples are shown in Table II and Figure 6.

thus compute the predicted $\tilde{\boldsymbol{\varphi}}_{\boldsymbol{d}}$ through an arctangent operation, and finally compute the loss as:

$$\mathcal{L}_{MAE_{\varphi_d}} = \mathbb{E}\left[\|\boldsymbol{\varphi_d} - \tilde{\boldsymbol{\varphi}}_{\boldsymbol{d}}\|_1\right]. \qquad (11)$$

Furthermore, if the training dataset instead contains not only the depth ground truth, but also images both with and without zero-mean noise, it is possible to extend the interpretability of the architecture, by pinning the output of the *Spatial Feature Extractor* with an additional loss. The *Spatial Feature Extractor* would therefore be dealing only with zero-mean noise, while the *Direct Phasor Estimator* would take care exclusively of MPI. The output of the *Global model* is guided instead by the EMD, which we had already employed in [15], and is defined as

$$\mathcal{L}_{EMD} = \mathbb{E}\left[\left\|\boldsymbol{X_g} - \tilde{\boldsymbol{X}}_{\boldsymbol{g}}\right\|_1\right], \qquad (12)$$

where $\boldsymbol{X_g}$ and $\tilde{\boldsymbol{X}}_{\boldsymbol{g}}$ are the cumulative sums of $\boldsymbol{x_g}$ and $\tilde{\boldsymbol{x}}_{\boldsymbol{g}}$ respectively. What this distance measure captures is the dissimilarity between the two distributions, i.e., the minimum amount of work needed to convert one into the other [33]. The performance of the different losses and the modularity of the approach will be thoroughly investigated in Section VI.

## V. Datasets

In this section we will introduce the main datasets employed for the training and evaluation of our model. We will employ both depth and transient datasets, due to the close relation between the two topics and the lack of datasets of the second kind. In particular, we will introduce the *Walls* dataset: a novel synthetic transient dataset based on simple structures which will be used for training and for evaluating the *Transient Reconstruction Module*. In Table I we show a comparison between the datasets.

### A. iToF Datasets

Regarding the iToF data, we will mainly focus on the synthetic and real datasets introduced in [14], [21]. These datasets come with amplitude and phase information at three different modulation frequencies: $20, 50$ and 60 MHz; the scenes depicted are simple indoor scenes, with a maximum distance smaller than 7.5 $m$ (the ambiguity range of the 20 MHz component) and high amounts of MPI. In particular, the synthetic dataset $S_1$ is composed of 54 scenes (40 for training and 14 for testing), has a high degree of shot noise and a spatial resolution of $240 \times 320$, while $S_3$, $S_4$ and $S_5$ are all real datasets with 8 images each, a limited amount of shot noise and a spatial resolution of $239 \times 320$. Dataset $S_1$ will be employed for training, $S_3$ for validation and $S_4$ and $S_5$ will be the main test sets for benchmarking the MPI correction capabilities of our network. The iToF2dToF dataset [23] will instead be used for some additional studies regarding the resilience to shot noise and MPI correction. The dataset is composed of a total of 5000 images with a spatial resolution of $120 \times 160$ and with all iToF measurements ranging from 20 to 600 MHz with a step of 20. The dataset presents an extremely high amount of shot noise (around $80\%$ of the total noise) and will be used as a stress test for our architecture.

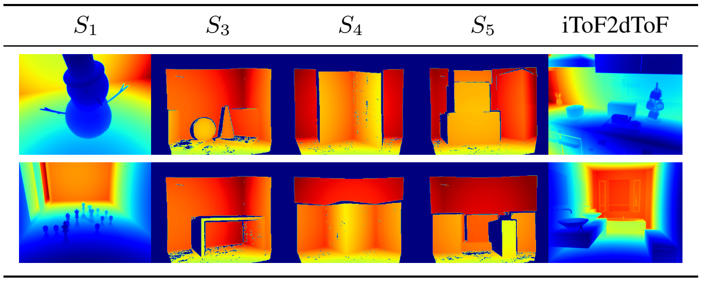


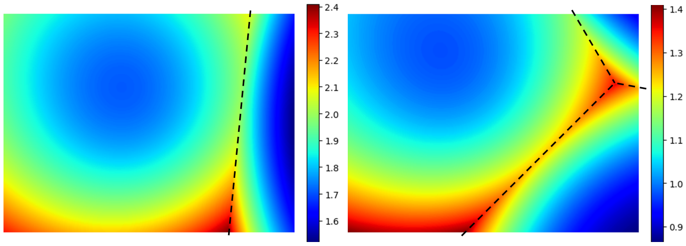
TABLE II: Sample images from the employed datasets

### B. The Walls Transient Dataset

The task of transient reconstruction is quite new in the literature and this is also due to the difficulties in acquiring a reliable transient dataset for the task. No real transient datasets are available and only few synthetic ones are freely accessible such as the FLAT dataset [25] and the Zaragoza [34] one.
In this paper we introduce the *Walls* dataset: a novel synthetic transient dataset based on simple geometries. The dataset has been simulated using the Microsoft ToF Tracer [35] with a maximum depth set to 5 $m$. The simulated scenes consist of one to three walls with varying angles between them. The point is that the dataset has been built as a template case for MPI. The scenes, while very simple, still capture some of the most common MPI scenarios where the overestimation is due to at maximum a couple of reflecting surfaces. This assumption may not be true in general, but it is a good approximation for most practical cases, as the light intensity is inversely proportional to the square of the travelled distance, making the contributions of longer paths mostly negligible. In total, the dataset is composed of 222 images, 53 with a single wall, 95 with two, and 74 with three. A couple of samples can be seen in Figure 6.

The spatial resolution of our images has been set to of $480 \times 640$, to match that of some of the most recent ToF cameras, while the temporal dimension has been divided into 2000 bins; keeping into account that the maximum depth is 5 $m$, this means that the depth quantization step consists of 2.5 $mm$, a desirable property for indoor settings. The dataset has no noise sources other than MPI and rendering noise.

[1] https://lttm.dei.unipd.it/paper_data/transientMPI/



(a) Two walls

(b) Three walls

Fig. 6: Depth images from our transient dataset

As the maximum depth of the *Walls* dataset amounts to 5 meters, while the one of $S_1$ gets to 7.5 meters, we decided to perform some data augmentation on the *Walls* dataset in order to cover the wider range. In practice, we added a shift to each of the transient vectors, randomly picking it from a uniform distribution in the $[0, 5]m$ range and from these we then recomputed the iToF measurements. The dataset can be found at https://lttm.dei.unipd.it/paper_data/transientMPI/.

In the following Section we will benchmark our model for the task of MPI correction and provide qualitative results both for this task and for the one of transient reconstruction.

## VI. Results

This section is devoted to the experimental evaluation of our method and to the comparison with the existing state-of-the-art. The quantitative evaluation and comparison will be carried out on the MPI correction task, while the transient reconstruction part will be evaluated only qualitatively.

### A. Training Details

The proposed neural networks have been trained using the $S_1$ and *Walls* datasets. From the first one we took the original training set of 40 images, while from the *Walls* dataset we randomly picked 134 images. The validation set consists instead of the 8 images from the real dataset $S_3$. In particular, the training data has been cut in patches of size $11 \times 11$, randomly chosen inside the images, while the validation set has been kept at full resolution. The models have been developed in Tensorflow 2.1, the trainings of our architecture have been performed on an NVIDIA 2080 Ti GPU, with ADAM as optimizer with a learning rate of $10^{-4}$ and a batch size of 2048. We will focus our evaluation for MPI correction on two models: the first one comprised of the first two modules introduced in Section IV which we will abbreviate *SD*, and the second a lighter architecture without the *Spatial Feature Extractor*. In this case the number of feature maps of the *Direct Phasor Estimator* was changed from 8 to 32 to provide the network with additional learning parameters. We will abbreviate this second model *D*. In all cases, each input patch has been normalized by the mean amplitude of its 20 MHz component to help generalizing on real data.

### B. Results on MPI correction

We will now compare our approach with some of the best performing MPI correction methods. The comparison will be made with SRA [12], an algorithmic approach, with DeepToF [20], one of the first deep learning approaches for MPI correction, with Buratto et al. [15], which was the starting point for our current architecture and with the approach from Agresti et al. [14], together with the subsequent domain adaptation approaches *in*-DA, *feat*-DA and *out*-DA proposed in [22].

| Approach | $S_4$ | | $S_5$ | | # of |
|---|---|---|---|---|---|
| | MAE $[cm]$ | Relative error | MAE $[cm]$ | Relative error | param. |
| Input (60 MHz) | 5.43 | - | 3.62 | - | - |
| Input (20 MHz) | 7.28 | - | 5.06 | - | - |
| SRA [18] | 5.11 | 94.1% | 3.37 | 93.1% | - |
| DeepToF [20] | 5.13 | 70.5%* | 6.68 | 132%* | 330k |
| + calibration | 5.46 | 75%* | 3.36 | 66.4%* | 330k |
| Agresti et al. [14] | 3.19 | 58.7% | 2.22 | 60.5% | 150k |
| +in-DA [22] | 2.40 | 44.2% | 1.74 | 48.1% | 150k |
| +feat-DA [22] | 2.37 | 43.6% | 1.66 | 45.8% | 150k |
| +output-DA [22] | 2.31 | 42.5% | **1.64** | **45.3%** | 150k |
| Buratto et al [15] | 2.60 | 47.9% | 2.12 | 58.6% | 22k |
| *D* (*Walls*) | 2.46 | 45.3% | 1.98 | 54.7% | **3k** |
| *D* (*Walls*) | 2.40 | 44.2% | 1.88 | 51.9% | 25k |
| *SD* (*Walls*+$S_1$) | **2.06** | **37.9%** | 1.87 | 51.7% | 23k |

TABLE III: Quantitative comparison between several state-of-the-art MPI correction algorithms on the real datasets $S_4$ and $S_5$. The evaluation metrics are the MAE and the relative error compared to the highest input frequency. * is compared to the 20 MHz input as it is single frequency. The complexity of each method is also displayed.

| Approach | $S_1$ $[cm]$ | # of parameters |
|---|---|---|
| Single freq. (60 MHz) | 16.7 | - |
| SRA [18] | 15.0 | - |
| DeepToF [20] + calibration | 26.1 | 330k |
| Agresti et al. [14] | 7.49 | 150k |
| Buratto et al [15] (original) | 30.5 | 22k |
| Buratto et al [15] (Walls) | 20.0 | 22k |
| Ours: *D* | 12.2 | **3k** |
| Ours: *SD* | **6.17** | 23k |

TABLE IV: Quantitative comparison between several state-of-the-art MPI correction algorithms on the test set of the synthetic dataset $S_1$. The evaluation metric is the MAE. The complexity of each method is also displayed.

In Table III we show the overall comparison between the cited approaches and the two proposed architectures. The first two columns show the MAE on the two real datasets $S_4$ and $S_5$, while the last one shows the network complexity of each approach; SRA has no entry as it isn't deep learning based. Regarding our approaches, the parameters of the *Transient Reconstruction Network* were not included in the total amount as it is not needed for MPI correction. Moreover, note that while the *SD* model has been trained on both the $S_1$ and *Walls* datasets, *D* has been trained on *Walls* alone, since as we will see it is not able to deal with shot noise due to its very narrow receptive field. The real datasets present a

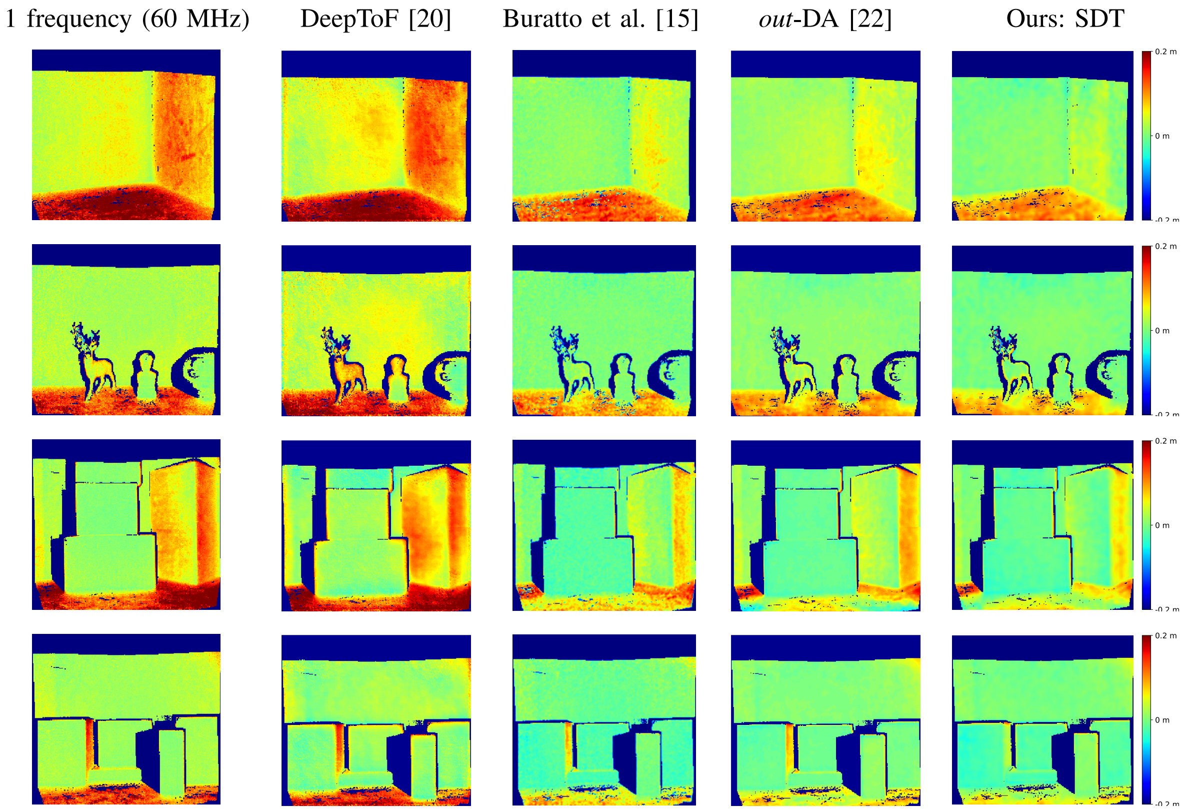


Fig. 7: Qualitative comparison between some of the best approaches for MPI correction. The first two images come from the $S_4$ dataset, while the other two from $S_5$. All the images display the reconstruction error w.r.t. the ground truth, where green means good reconstruction and red an overestimation.

clear challenge due to the domain shift between synthetic and real data. In practice, the resemblance between training and test data is strictly limited by the accuracy of the simulation, which can mimic a real scenario only up to a certain extent. As we can see from the table, our approach not only clearly outperforms both the architectures proposed in [15] and [14], but also beats the results from [22] on $S_4$, while falling shortly behind on $S_5$, all by using just $\frac{1}{7}$ of the parameters from [22]. This is particularly striking as differently from [22] we only rely on synthetic data for our prediction. Given this, we have clear reasons to expect an even better performance by using some unsupervised domain adaptation techniques as was done in [22]. Another interesting outcome is the fact that the $D$ model shows quite competitive results w.r.t. *SD* and [22] and at the same time outperforms other architectures such as [14] and [15]. The $D$ model is extremely light, with just around 3k learnable parameters, but still gets close to state-of-the-art results. A second version of the $D$ model with 25k parameters has also been trained but the improvement is not significant w.r.t. the lighter version.

Figure 7 shows a qualitative comparison on a few images from the $S_4$ and $S_5$ datasets. The approach we propose shows a clear improvement on the competitors, providing a good reconstruction also on regions highly corrupted by MPI such as the floor and other steeply sloped scene elements. As an example we can consider the first row of Figure 7, where not only the floor shows a better reconstruction, but the MPI artefacts on the wall on the right are almost completely corrected. Similar considerations can be made on the last image row, where our approach is the only one able to clear the right face of the box on the left side, a particularly difficult surface due to its tilt.

In Table IV we report instead the comparison made on the $S_1$ dataset. This is interesting due to the abundant presence of shot noise which can hinder the performance of some approaches. The problem is that, while MPI correction can be performed using only information along the transient dimension, that is not possible for shot noise removal. Networks such as that of Buratto et al. [15] and our $D$ model have a receptive field of size $3 \times 3$, which hampers the performance on $S_1$. We tested the approach of [15] first by using the original pretrained model, which was trained on the FLAT dataset [25], and then by retraining it on the *Walls* dataset (since training it on $S_1$ is not possible due to the lack of transient information). In both cases, as expected, the performance is not satisfactory. Similar conclusions can be drawn when training the $D$ model alone. Its performance is better than that of [15] as $D$ can also be trained on $S_1$ directly, but still far from optimal. In this case, as shown in the table, the addition of the $S$ module was crucial, as the gap between *SD* and $D$ is much wider than before. At the same time however, with the *SD* architecture we are still able to outperform approaches relying on much more complex networks, and in particular, the one from Agresti et al. [14] (that introduced the dataset $S_1$), by more than 1 cm.

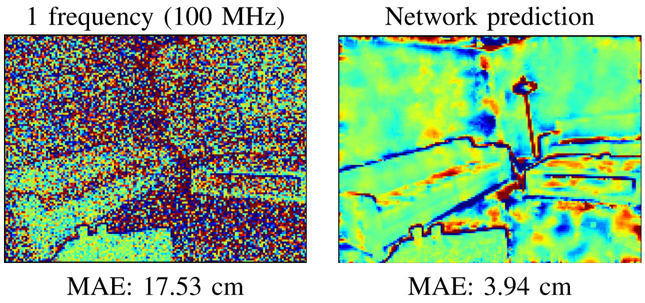


Fig. 8: Qualitative results on a particularly noisy image from the test set of the iToF2dToF dataset [23]. On the left side the single frequency reconstruction at 100 MHz, on the right side the network prediction.

To conclude, we will now see how our model fares in the presence of extremely high amounts of shot noise. To this aim, we trained our approach on the iToF2dToF dataset [23], which, as described in Section V, has a very high amount of zero-mean error. To put things into perspective, the single frequency reconstruction at 100 MHz of the test set from the measurements with shot noise leads to a MAE of 7.24 cm, while the same computation done on images with only MPI, gives an error of 1.45 cm, meaning that MPI accounts only for 20% of the total reconstruction error.

Following the setup from [23], we used two input frequencies, 20 and 100 MHz, masked the edges using a Canny edge detector during testing and did not consider the highest 1% of errors for the final computation. Our approach shows some remarkable denoising capabilities even in this scenario as it can be seen in Figure 8. Quantitatively, our approach reaches a test error of 1.97 cm, removing around 75% of the noise, that is behind the performances of iToF2dToF, which removes around 82% of the noise, but this is to be expected considering the very different scenario. Our approach consists of a very light architecture with extremely good MPI denoising capabilities, which is also able to deal with relatively high amount of shot noise, but the removal of zero-mean noise sources is not its primary objective, while iToF2dToF mostly focuses on this task. Moreover, the difference in complexity between the two architectures is striking: iToF2dToF needs almost two million parameters, while our network is still able to remove three quarters of the total noise using 100 times less parameters.

### C. *Transient Reconstruction*

We will now provide some qualitative results on the performance of the *Transient Reconstruction Module*, highlighting its pros and current limitations, and then make a qualitative comparison with iToF2dToF [23], the only other data-driven model that tries to reconstruct transient information. In Figure 9 we can see a comparison between the transient ground truth and the reconstruction of our network for 4 pixels from our transient dataset. On the top row we show a pair of good examples, where both the direct and the global components are captured quite well; on the bottom row instead we can see some of the limitations of our model. The direct component still shows a good reconstruction, while the global is more challenging to be reconstructed. The y axis has been logarithmically scaled to show both the direct and global components. In Figure 10 we show the performance of our approach on two pixels from the iToF2dToF dataset. Since the direct component of the transient pixel is very spread in this case, and our method only predicts a single peak, we show also an edited version of the ground truth for a better comparison. We substituted the original direct component with a single peak whose magnitude corresponds to the sum of all elements of the original direct, and whose position is taken from the weighted sum of all direct elements' positions, with each weight consisting of the element value itself. We can see that our method shows promising performances also on a previously unseen dataset, capturing very precisely the direct component, and reconstructing reasonably well the global. It is also clear that our model proposes a much more convincing reconstruction w.r.t. that of iToF2dToF for both components, as the competitor has a much worse estimate, especially for the global component.

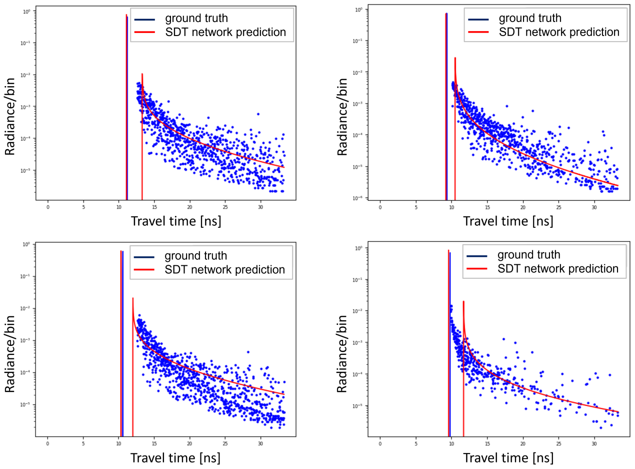


Fig. 9: Qualitative examples showing the transient reconstruction capabilities of our approach. On the top row we show a pair of good examples, while on the bottom one a pair of less accurate ones. All the plots have a logarithmic scaling.

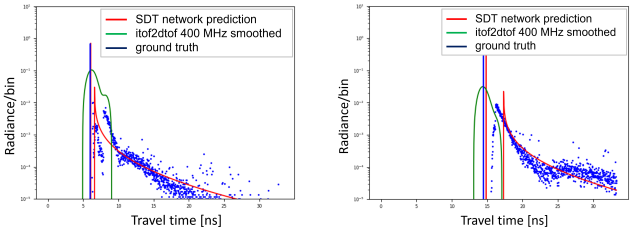


Fig. 10: Qualitative comparison on two pixels from the iToF2dToF dataset. The direct ground truth has been substituted by a peak whose magnitude consists of the sum of the whole direct, and its position of the weighted average of the direct elements.

| Training dataset | # of images $S_1$ | # of images *Walls* | $S_1$ $[cm]$ | $S_4$ $[cm]$ | $S_5$ $[cm]$ |
|---|---|---|---|---|---|
| $S_1$ ($\boldsymbol{\varphi_d}$) | 40 | - | $\mathbf{5.93}$ | 3.65 | 2.33 |
| *Walls* ($\boldsymbol{\varphi_d}$) | - | 45 | 19.2 | 2.46 | 2.39 |
| *Walls* ($\boldsymbol{\varphi_d}$) | - | 134 | 18.4 | 2.34 | 2.40 |
| *Walls* ($\boldsymbol{v_d}$) | - | 134 | 12.2 | 2.26 | 2.44 |
| $S_1$ ($\boldsymbol{\varphi_d}$) + *Walls* ($\boldsymbol{\varphi_d}$) | 40 | 134 | 6.99 | 2.26 | 2.04 |
| $S_1$ ($\boldsymbol{\varphi_d}$) + *Walls* ($\boldsymbol{v_d}$) | 40 | 134 | 6.17 | $\mathbf{2.06}$ | $\mathbf{1.87}$ |

TABLE V: Quantitative comparison of the performance of different training datasets on the synthetic dataset $S_1$ and on the real datasets $S_4$ and $S_5$ for different training datasets and losses. The evaluation metric is the MAE. All trainings have been performed on the *SD* network.

| Input frequencies | $S_1$ dataset $[cm]$ | $S_4$ dataset $[cm]$ | $S_5$ dataset $[cm]$ |
|---|---|---|---|
| Single frequency 50 MHz | 18.0 | 5.31 | 3.82 |
| $20, 50$ MHz | 6.56 | 3.44 | 2.31 |
| $20, 50, 60$ MHz | $\mathbf{6.17}$ | $\mathbf{2.06}$ | $\mathbf{1.87}$ |

TABLE VI: Quantitative comparison of the performance of a different number training frequencies on the synthetic dataset $S_1$ and on the real datasets $S_4$ and $S_5$. The evaluation metric is the MAE. The first row shows the baseline error at 50 MHz without any processing.

### D. Ablation studies

As follows, we show some ablation studies which explain the choices behind our network architecture. Among other variations, we study how the training datasets, the receptive field and the number of input frequencies influence the final performance.

*a) Training dataset and number of frequencies:* The prediction quality of any data-driven technique heavily depends on the goodness of the dataset used to train it in the first place. For this reason, we decided to test three different scenarios: in the first one we trained our *SD* model on the $S_1$ dataset alone, in the second one the *Walls* dataset was the only input of the network, and in the final one we used both for supervision, as we did for the results in the previous section. In order to make the comparison fair, we also decided to perform different trainings on the *Walls* dataset supervising either on $\boldsymbol{v_d}$ or on $\boldsymbol{\varphi_d}$ (i.e., the phase of the direct component). Finally, as the two datasets have a different size, we also added one entry where our method has been trained on a reduced version of our dataset (45 training images instead of 134). The outcome of this study is shown in Table V. Considering first the results on $S_1$, we can see that the training performed on $S_1$ itself leads to the best performance, followed at a short distance by using both datasets; training on *Walls* alone instead falls behind by a large margin. This is not surprising as the images from *Walls* have no shot noise, thus explaining the poor performance. What's more remarkable instead is the prediction on real data, as the $S_1$ dataset yields a significantly poorer performance when compared to the training on *Walls*. The dissimilarity between the two datasets is striking: the former is made of much more complex scenes, shows a wide range of textures and has a good amount of simulated noise; the latter instead focuses on extremely simple structures, no changes in texture and its only noise source is MPI. What seems to be happening is that the added complexity of the dataset and some issues it presents (some scene elements of the $S_1$ dataset present very unreliable information), make the prediction harder for the network. Moreover, our approach relies mostly on the information in the transient direction, making in this way the complex structures from $S_1$ less relevant than the cleaner phasor data from *Walls*. In the end, combining the two datasets leads to the best overall solution, with a competitive performance on $S_1$, and the best prediction on both real datasets. It is also useful to point out that using $\boldsymbol{v_d}$ for supervision improves on training only on the phase, showing how a transient dataset can be useful for MPI correction.

| Receptive field | $S_1$ dataset $[cm]$ | $S_4$ dataset $[cm]$ | $S_5$ dataset $[cm]$ |
|---|---|---|---|
| $7 \times 7$ | 8.08 | 2.44 | $\mathbf{1.82}$ |
| $11 \times 11$ | $\mathbf{6.17}$ | 2.06 | 1.87 |
| $15 \times 15$ | 6.35 | $\mathbf{2.00}$ | 2.10 |
| $21 \times 21$ | 8.19 | 2.42 | 2.45 |

TABLE VII: Quantitative comparison of the performance of different receptive fields for the *SD* network on the synthetic dataset $S_1$ and on the real datasets $S_4$ and $S_5$. The evaluation metric is the MAE.

Another point of interest concerns the ability of the model in dealing with a different number of input frequencies. In particular, we decided to train our model with two frequencies, 20 and 50 MHz, and see how it coped in comparison to the three frequencies input. In Table VI we can see that the lack of the 60 MHz component (depth estimated from higher frequencies has typically a smaller error) has indeed a toll on the overall performance, but the model is still able to clean a noteworthy amount of MPI. To put things into perspective, it is enough to look at Table V, where we can see how a 2 frequencies training on *Walls* still provides a better prediction than a 3 frequencies training on $S_1$ on the real datasets.

*b) Receptive field and network complexity:* As we have seen in Table III, there is no clear correlation between the number of parameters of the architecture and its actual performance. This is due to multiple factors, such as the high risk of overfitting due to the domain difference between training and test data, the relatively small sizes of the datasets (even the *Walls* one only comprises around 200 images) and the main focus of the model itself, which can be centered on the use of spatial features (e.g. [14], [21]) or on the transient dimension ([15] and ours). In our case, we have to consider an additional factor which are the characteristics of our training datasets. In particular, while a relatively large receptive field would be better in order to deal with shot noise, we cannot enlarge it too much as our main tool, the *Walls* dataset, is made mostly of flat surfaces and there is a risk of overfitting its structure. Learning too much from this dataset geometry, as it can be seen in Table VII, decreases the performance of the model. From the Table we can see that the overall best performance on the datasets arises from a receptive field of either $11 \times 11$ or $15 \times 15$, while

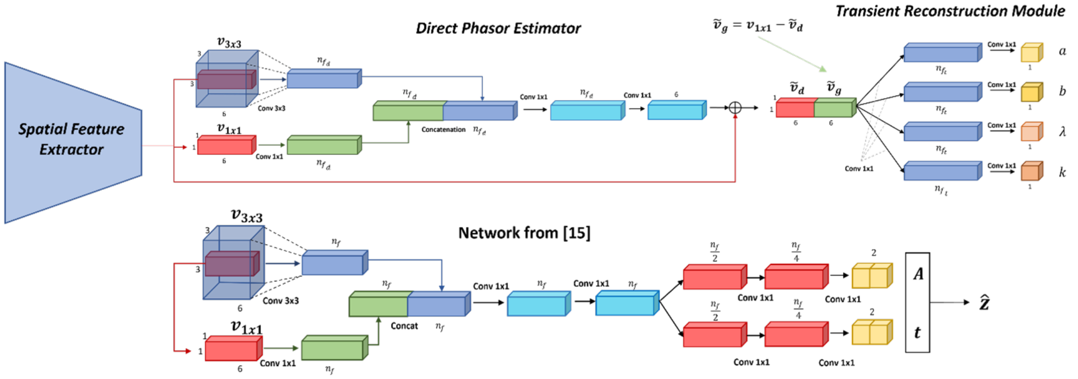


Fig. 11: Comparison between the network structure we propose and the one introduced in [15]

it clearly degrades for smaller or bigger sizes. We decided to employ a receptive field of $11 \times 11$ due to its slightly better performance and the reduced network complexity.

## VII. Conclusions

In this work we have introduced a novel network for MPI denoising and transient reconstruction. The architecture is modular: the *Spatial Feature Extractor* is useful to deal with zero mean errors, the *Direct Phasor Estimator* deals with MPI and the *Transient Reconstruction Module* reconstructs the transient information from the previous. We have proposed two very compact networks, and compared their performance against some of the best models in the literature. Our *SD* architecture reaches state-of-the-art performance both on synthetic and real data, while the *D* one shows comparable performance, while only needing 3k network weights. The model shows also promising results regarding the reconstruction of transient information, but still has a few limitations that we plan to address in future works. Some key points we need to address are the shape of the function itself, since the Weibull function is able to predict only a single reflection, and the usage of a larger receptive field in our learning process, as currently the *Transient* network relies only on the output of the *SD* structure, which still has a very small receptive field. A key challenge that we will explore is how to find an accurate model for the global component that can be represented with a few parameters. Moreover, we plan to substitute the parametric functions with a network in order to learn more complex global component shapes. Finally, we also plan to investigate applications of our method, such as non-line-of-sight (NLOS) imaging. It has been shown [36] that from the information carried by the global component it is possible to completely reconstruct a NLOS pixel; while the cited work relies SPAD sensors, our plan is to perform the same task starting from iToF information.

## Appendix A

In Figure 11 we highlight the differences between the proposed architecture and the model from [15]. The output space of the model from [15] had been built with the idea of a two-peaks transient vector. The job of the network is to predict an encoding vector $\hat{\mathbf{z}}$ which consists of the amplitude and time position of the two peaks. This 4 elements output is then mapped back into a transient vector of 1000 elements, all of which are 0 apart from the two predicted ones. In practice, the space where $\hat{\mathbf{z}}$ resides is useful as it keeps the number of network outputs limited but has no direct connection to the input space and is not directly used for the loss computation. When we consider the proposed $D$ model instead, we have an intermediate estimation $v_d$ which belongs to the same space as the input and that is directly used for the loss calculation. This structural difference makes it possible to train the $D$ model uniquely for MPI correction, without going to the transient domain. This is a clear improvement w.r.t. [15] as this allows to train our model on datasets with only depth ground truth but no transient information; the same thing is impossible using the network from [15] as it needs the transient ground truth. Notice that datasets with depth information can be acquired with widespread tools while the acquisition of the transient ground truth for real data is extremely challenging and nowadays only synthetic datasets with this type of data are available.

Another novelty is the *Spatial Feature Extractor*, a convolutional structure used for dealing with shot noise. Both the $D$ network and the structure from [15] have a very narrow receptive field (i.e. 3x3), and show very poor performances when employed on noisy datasets (see Table IV). Adding the $S$ network halves the error when compared to $D$ alone.

Finally, the global light model proposed in this work is significantly more advanced; in [15] the global component is approximated by a Dirac peak while here we use a Weibull distribution which is a much more accurate approximation.

## References

[1] Q. Zhu, L. Chen, Q. Li, M. Li, A. Nüchter, and J. Wang, “3d lidar point cloud based intersection recognition for autonomous driving,” in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 456–461.

[2] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, “Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[3] E. Bostanci, N. Kanwal, and A. F. Clark, “Augmented reality applications for cultural heritage using kinect,” *Human-centric Computing and Information Sciences*, vol. 5, no. 1, pp. 1–18, 2015.

[4] Y. M. Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Miscusik, and S. Thrun, “Multi-view image and tof sensor fusion for dense 3d reconstruction,” in *Proceedings of International Conference on Computer Vision Workshops (ICCVW)*, 2009, pp. 1542–1549.

[5] C. Kerl, M. Souiai, J. Sturm, and D. Cremers, “Towards illumination-invariant 3d reconstruction using tof rgb-d cameras,” in *2014 2nd International Conference on 3D Vision*, vol. 1, 2014, pp. 39–46.

[6] F. Amzajerdian, D. Pierrottet, L. Petway, G. Hines, and V. Roback, “Lidar systems for precision navigation and safe landing on planetary bodies,” in *International Symposium on Photoelectronic Detection and Imaging 2011: Laser Sensing and Imaging; and Biological and Medical Applications of Photonics Sensing and Imaging*, F. Amzajerdian, W. Chen, C. Gao, and T. Xie, Eds., vol. 8192, International Society for Optics and Photonics. SPIE, 2011, pp. 27 – 33. [Online]. Available: https://doi.org/10.1117/12.904062

[7] U. R. Dhond and J. K. Aggarwal, “Structure from stereo-a review,” *IEEE transactions on systems, man, and cybernetics*, vol. 19, no. 6, pp. 1489–1510, 1989.

[8] R. Horaud, M. Hansard, G. Evangelidis, and M. Clément, “An overview of depth cameras and range scanners based on time-of-flight technologies,” *Machine Vision and Applications*, vol. 27, 10 2016.

[9] R. O. Dubayah and J. B. Drake, “Lidar remote sensing for forestry,” *Journal of Forestry*, vol. 98, no. 6, pp. 44–46, 2000.

[10] P. Zanuttigh, G. Marin, C. Dal Mutto, F. Dominio, L. Minto, and G. M. Cortelazzo, “Time-of-flight and structured light depth cameras,” *Technology and Applications*, pp. 978–3, 2016.

[11] https://www.sony.de/electronics/smartphones/xperia-1m2.

[12] D. Freedman, Y. Smolin, E. Krupka, I. Leichter, and M. Schmidt, “Sra: Fast removal of general multipath for tof sensors,” in *Proceedings of European Conference on Computer Vision (ECCV)*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 234–249.

[13] S. Su, F. Heide, G. Wetzstein, and W. Heidrich, “Deep end-to-end time-of-flight imaging,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[14] G. Agresti and P. Zanuttigh, “Deep learning for multi-path error removal in tof sensors,” in *Proceedings of European Conference on Computer Vision Workshops (ECCVW)*, September 2018.

[15] E. Buratto, A. Simonetto, G. Agresti, H. Schäfer, and P. Zanuttigh, “Deep learning for transient image reconstruction from tof data,” *Sensors*, vol. 21, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/6/1962

[16] S. Fuchs, “Multipath interference compensation in time-of-flight camera images,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3583–3586.

[17] S. Fuchs, M. Suppa, and O. Hellwich, “Compensation for multipath in tof camera measurements supported by photometric calibration and environment integration,” in *Computer Vision Systems*, M. Chen, B. Leibe, and B. Neumann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 31–41.

[18] D. Freedman, Y. Smolin, E. Krupka, I. Leichter, and M. Schmidt, “Sra: Fast removal of general multipath for tof sensors,” in *Proceedings of European Conference on Computer Vision (ECCV)*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 234–249.

[19] A. Bhandari, M. Feigin, S. Izadi, C. Rhemann, M. Schmidt, and R. Raskar, “Resolving multipath interference in kinect: An inverse problem approach,” in *SENSORS, 2014 IEEE*, 2014, pp. 614–617.

[20] J. Marco, Q. Hernandez, A. Muñoz, 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 Trans. Graph.*, vol. 36, no. 6, 2017. [Online]. Available: https://doi.org/10.1145/3130800.3130884

[21] G. Agresti, H. Schaefer, P. Sartor, and P. Zanuttigh, “Unsupervised domain adaptation for tof data denoising with adversarial learning,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[22] G. Agresti, H. Schafer, P. Sartor, Y. Incesu, and P. Zanuttigh, “Unsupervised domain adaptation of deep networks for tof depth refinement,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 01, pp. 1–1, oct 5555.

[23] F. Gutierrez-Barragan, H. G. Chen, M. Gupta, A. Velten, and J. Gu, “itof2dtof: A robust and flexible representation for data-driven time-of-flight imaging,” *CoRR*, vol. abs/2103.07087, 2021. [Online]. Available: https://arxiv.org/abs/2103.07087

[24] 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.

[25] Q. Guo, I. Frosio, O. Gallo, T. Zickler, and J. Kautz, “Tackling 3d tof artifacts through learning and the flat dataset,” in *Proceedings of European Conference on Computer Vision (ECCV)*, September 2018.

[26] G. Dong, Y. Zhang, and Z. Xiong, “Spatial hierarchy aware residual pyramid network for time-of-flight depth denoising,” in *Proceedings of European Conference on Computer Vision (ECCV)*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 35–50.

[27] F. Heide, L. Xiao, W. Heidrich, and M. B. Hullin, “Diffuse mirrors: 3d reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[28] J. Lin, Y. Liu, M. B. Hullin, and Q. Dai, “Fourier analysis on transient imaging with a multifrequency time-of-flight camera,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[29] Y. Liang, M. Chen, Z. Huang, D. Gutierrez, A. Muñoz, and J. Marco, “A data-driven compression method for transient rendering,” in *ACM SIGGRAPH 2019 Posters*, ser. SIGGRAPH ’19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3306214.3338582

[30] M. Gupta, S. K. Nayar, M. B. Hullin, and J. Martin, “Phasor imaging: A generalization of correlation-based time-of-flight imaging,” *ACM Trans. Graph.*, vol. 34, no. 5, 2015. [Online]. Available: https://doi.org/10.1145/2735702

[31] Y. Liang, M. Chen, Z. Huang, D. Gutierrez, A. Muñoz, and J. Marco, “A data-driven compression method for transient rendering,” in *ACM SIGGRAPH 2019 Posters*, ser. SIGGRAPH ’19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3306214.3338582

[32] W. Weibull *et al.*, “A statistical distribution function of wide applicability,” *Journal of applied mechanics*, vol. 18, no. 3, pp. 293–297, 1951.

[33] S. Cohen and L. Guibas, “The earth mover’s distance: Lower bounds and invariance under translation,” Stanford University CA Dept. of Computer Science, Tech. Rep., 1997.

[34] M. Galindo, J. Marco, M. O’Toole, G. Wetzstein, D. Gutierrez, and A. Jarabo, “A dataset for benchmarking time-resolved non-line-of-sight imaging,” 2019. [Online]. Available: https://graphics.unizar.es/nlos

[35] M. Pharr, W. Jakob, and G. Humphreys, *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[36] S. Xin, S. Nousias, K. N. Kutulakos, A. C. Sankaranarayanan, S. G. Narasimhan, and I. Gkioulekas, “A theory of fermat paths for non-line-of-sight shape reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6800–6809.