Tackling 3D ToF Artifacts Through Learning and the FLAT Dataset

Qi Guo[a,b], Iuri Frosio[a], Orazio Gallo[a], Todd Zickler[b], Jan Kautz[a]

[a]NVIDIA, [b]Harvard SEAS

Abstract. Scene motion, multiple reflections, and sensor noise introduce artifacts in the depth reconstruction performed by time-of-flight cameras. We propose a two-stage, deep-learning approach to address all of these sources of artifacts simultaneously. We also introduce FLAT, a synthetic dataset of 2000 ToF measurements that capture all of these nonidealities, and allows to simulate different camera hardware. Using the Kinect 2 camera as a baseline, we show improved reconstruction errors over state-of-the-art methods, on both simulated and real data.

Keywords: Time-of-Flight, MPI artifacts, motion artifacts.

1 Introduction

Depth estimation is central to several computer vision applications. Among the many existing strategies for extracting a scene's 3D information, Time-of-Flight (ToF) cameras are particularly popular due to their robustness and affordability.

ToF cameras leverage the relation between an object's distance from the sensor and the amount of time required for photons to travel to that object and back. In particular, Amplitude-Modulated Continuous-Wave (AMCW) cameras emit a periodic light signal and measure its phase delay upon return: the phase delay offers an estimate of the time of flight and, in turn, of depth. Any implementation of this approach requires several careful considerations.

First and foremost, the phase delay wraps at distances that correspond to multiples of the modulation period. A common approach is to combine information from different modulation frequencies: longer modulation periods extend the unambiguous range, while shorter periods allow resolving finer details. Using multiple frequencies, however, is not without consequences. The dynamic parts of the scene may be displaced between sequential measurements at different frequencies, causing depth estimation errors[1] that are particularly strong along the depth discontinuities. These artifacts can be identified and removed, at the cost of missing depth information at the corresponding pixels.

Another consideration is multi-path light transport. In addition to the direct emitter-object-sensor path, light may follow other paths and bounce multiple times before being recorded by one pixel. This phenomenon, called multiple-path

[1] Some cameras use spatial multiplexing instead, and synchronize neighboring pixels with specific emission frequencies, thus sacrificing spatial resolution.

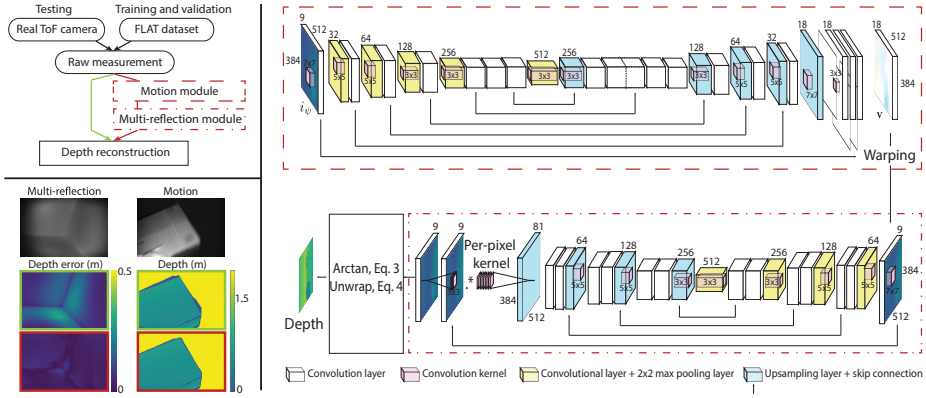


Fig. 1: The traditional ToF processing pipeline (green, libfreenect2 [1] as an example) and the proposed framework (red). The lower left panel shows artifacts generated by MPI and motion, that show up respectively as deformation close to corners and spikes or missing data close to the boundaries of moving objects in the traditional flowchart (green). These artifacts are greatly reduced by the proposed framework (red), which is based on the introduction of the motion and multi-reflection modules depicted in the right part.

interference (MPI), causes biased estimates of depth. Several methods attempt to attenuate the effects of MPI [2–6], some employing multiple modulation frequencies [2, 3], which, as discussed above, can introduce motion artifacts. These could be reduced in theory by increasing the capture speed, but shorter exposure times may lead to a lower signal-to-noise ratio because of shot noise.

The delicate trade-offs between phase-unwrapping, motion, sensor noise, and MPI, have been studied in a variety of ways, with several methods attempting to attenuate the effects of the different sources of artifacts *independently*. To the best of our knowledge, however, there have been no attempts to tackle them jointly by learning, possibly because of the lack of large datasets.

We introduce a learning-based approach to tackle dynamic scenes, MPI, and shot noise *simultaneously*. Our two-stage architecture (Fig. 1) operates directly on the raw measurements of a multi-frequency ToF sensor, and produces improved measurements that are compatible with standard equations for phase-unwrapping and conversion to depth. The first stage is an encoder-decoder architecture [7] that attenuates motion artifacts by spatially warping the raw measurements onto a common 2D reference frame. This increases the number of pixels whose depth can be reliably estimated, especially near the boundaries of moving objects. The second stage is a kernel predicting network [8] that attenuates MPI and shot noise. Because ground truth depth for real-world scenes is difficult to capture, we introduce FLAT (Flexible, Large, Augmentable, ToF dataset), a synthetic dataset for training and evaluation, which allows to accu-

rately simulate the raw measurements of different AMCW ToF cameras (including the Kinect 2) in the presence of MPI artifacts and shot noise; FLAT data can be augmented to approximate motion artifacts as well. Our contributions are:

(i) FLAT, a large, synthetic dataset of ToF measurements that simulates MPI, motion artifacts, shot noise, and different camera response functions.

(ii) A Deep Neural Network (DNN) architecture for attenuating motion, MPI, and shot noise artifacts that can be trained both in the raw measurement or depth domain.

(iii) A thorough validation, including an ablation study and a comparison with state-of-the-art algorithms for reducing MPI and motion artifacts.

(iv) A complete characterization of the Kinect 2 camera, including its camera response function and sensor noise characteristics.

Our DNN model, dataset and characterization of the Kinect 2 are available at `http://research.nvidia.com/publication/2018-09_Tackling-3D-ToF`.

2 Related Work

Several works separately reduce artifacts due to MPI, motion, or shot noise in AMCW ToF imaging. We group them into four categories.

Measurement noise reduction. Raw ToF measurements suffer from both systematic and random noise [9]. Systematic errors are often associated with imperfect sinusoidal modulations and can be reduced through calibration [9–11]. Shot noise and other types of random noise are typically addressed through bilateral filtering of the raw measurements, the depth map, or both sometimes using other images for guidance [12]. The performance of these approaches is generally satisfactory, and any system that intends to replace them, ours included, should not perform noticeably worse.

Motion artifacts reduction. Motion artifacts occur when objects move and ToF raw measurements are captured sequentially. Gottfried et al. [13] identify three ways to attenuate them: reduce the number of sequential measurements; detect and correct the regions affected by motion, both in the raw measurement domain and in the depth domain; or estimate the 2D motion fields between raw 2D measurement maps and apply corrective spatial warping. One way to detect affected pixels in raw measurements is by checking the *Plus* and *Minus rules* [14, 15] that derive from physical constraints on the light measured in a static scene. After detection, pixels affected by motion blur can be corrected, for example, by interpolation [14]. Object motion also affects the frequency of reflected signals due to Doppler effect; however, to use the frequency shift to measure object motion requires a higher number of measurements and extensive processing [16].

Physics-based MPI reduction. Algorithms for recovering depth from ToF correlations usually assume the pure measurement of direct, single-bounce (emitter-surface-sensor) light paths. In practice, many photons bounce multiple times, causing "erroneous" measurements [2, 10]. If multiple modulation

frequencies are used, the problem can be tackled by processing the temporal change of each pixel of the raw measurements in the Fourier domain. For instance, in the absence of noise, K interfering paths can be resolved by $2K + 1$ frequency measurements [3]. Other techniques for the per-pixel temporal processing of raw ToF measurements include Prony's method, the matrix pencil method, orthogonal matching pursuit, EPIRIT/MUSIC, and atomic norm regularization [2]. Freedman et al. propose a real-time temporal processing technique that uses per-pixel optimization based on a light transport model with sparse and low-rank components [17]. Phasor Imaging exploits the fact that the effects of MPI are diminished at much higher modulation frequencies, and shows that simple temporal processing can succeed with as few as two such frequencies, albeit with a reduced unambiguous working range [18].

Learning-based MPI reduction. The difficulty of modeling MPI analytically makes machine learning an enticing alternative for its reduction. However, one obstacle is the lack of large, physically-accurate datasets for training, which are difficult to capture [19] and prohibitively expensive to simulate, until recently. Marco et al. use an encoder to learn a mapping from captured ToF measurements to a representation of (MPI-corrupted) depth, and then combine this with a limited number of simulated, direct-only ToF measurements to train a decoder that produces MPI-corrected depth maps [4]. Mutny et al. focus on corners of different materials and use a dataset of such corners to train a random forest with hand-crafted features [5]. A different strategy is taken by Son et al., who use a robotic arm and structured light system to capture ToF measurements with registered ground-truth depth [6]. They then train two neural networks to correct depth and refine edges through geodesic filtering.

We leverage the availability of computational power and advances in transient rendering [20, 10] to synthesize a training dataset sufficiently large and diverse to explore a much larger class of learned models. In addition to physically-accurate MPI effects, the dataset provides realistic shot noise, supports augmentation with approximate motion models, and allows for efficient generation of raw measurements from AMCW ToF sensors with arbitrary modulation signals.

3 Time-of-Flight Camera Models

In this section we first review the theory of ToF reconstruction in the ideal case. We show that equations for depth reconstruction are differentiable, which allows for backpropagation. We then show the effect of MPI and motion on depth reconstruction, which helps framing the learning problem and defining the important factors for training. We leverage these elements to train a neural network, working in the domain of the raw measurements and before unwrapping, aimed at correcting these artifacts. The section is closed by an accurate characterization of the Kinect 2, which is our hardware testbed; this serves to produce accurate simulations for training and reduce the shift between synthetic and real data. All the math details for the section are in the Supplementary.

3.1 Ideal Camera Model

An AMCW ToF camera illuminates the scene with an amplitude-modulated, sinusoidal signal $g(t) = g_1 \cos(\omega t) + g_0$, [16]. If camera and light source are co-located, and the scene static, the signal coming back to a pixel is

$$s\left(t\right) = \int_{-\infty}^{t} a(\tau) \cos(\omega t - 2\omega\tau) d\tau, \tag{1}$$

where $a(\tau)$ is the scene response function, i.e. the signal reaching the pixel at time τ. In an ideal scenario, the light is reflected once and the scene response is an impulse, $a(\tau) = a\delta(\tau - \tau_0)$. The travel time τ_0 directly translates to depth.

A *homodyne* ToF camera modulates the incident signal with a phase-delayed reference signal at the same frequency, $b\cos\left(\omega t - \psi\right)$. The exposure time of the camera is usually set to $T >> 2\pi/\omega$. Simple trigonometry allows writing the raw correlation measurement $i_{\psi,\omega}$ as:

$$i_{\psi,\omega} = \int_{-T/2}^{T/2} s(t) b \cos\left(\omega t - \psi\right) dt \approx a(\tau_0) b \cos(\psi - 2\omega\tau_0) = a f_{\psi,\omega}(\tau_0), \tag{2}$$

where we denote $f_{\psi,\omega}(\tau) = b\cos(\psi - 2\omega\tau)$ as the *camera function*. Using raw measurements captured at a single frequency ω and $K \geq 2$ phases $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_K)$, the depth d can be recovered at each pixel as:

$$d = c/(2\omega) \ \arctan\left[(\sin \boldsymbol{\psi} \cdot \boldsymbol{i}_{\boldsymbol{\psi},\omega}) / (\cos \boldsymbol{\psi} \cdot \boldsymbol{i}_{\boldsymbol{\psi},\omega})\right], \tag{3}$$

where $\boldsymbol{i}_{\boldsymbol{\psi},\omega}$ is the K-vector of per-phase measurements.

However, d wraps for depths larger than $\pi c/\omega$, and additional measurements at $L \geq 2$ different frequencies $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_L)$ are required. Denoting the measurements at frequency ω_l as $\boldsymbol{i}_{\boldsymbol{\psi},\omega_l}$, an analytical expression for d was provided by Goshov and Solodkin [21] based on the Chinese remainder theorem:

$$d = \sum_{\boldsymbol{\omega}} A_l(\boldsymbol{\omega}) \arctan\left[(\sin \boldsymbol{\psi} \cdot \boldsymbol{i}_{\boldsymbol{\psi},\omega_l}) / (\cos \boldsymbol{\psi} \cdot \boldsymbol{i}_{\boldsymbol{\psi},\omega_l})\right] + B(\boldsymbol{\omega}), \tag{4}$$

where $\{A_l(\boldsymbol{\omega})\}_{l=1,\ldots,L}$ and $B(\boldsymbol{\omega})$ are constants. Based on Eq. 4, one can easily obtain the derivative $\partial d / \partial \boldsymbol{i}_{\boldsymbol{\psi},\omega_l}$, which makes it possible to backpropagate the error on d and to perform end-to-end training.

3.2 The Impact of Multiple Paths

In a realistic scenario, the signal that reaches the sensor is corrupted by multiple light paths that undergo different reflection events and have different path lengths. This means that the scene response $a(\tau)$ is not an impulse anymore, as it measures the arrival of the light reaching a pixel from all the possible light paths that connect it to the emitter. In this case, Eq. 2 becomes:

$$\begin{aligned} i_{\psi,\omega} &= \int_{-T/2}^{T/2} \left(\int_{-\infty}^{t} a(\tau) \cos(\omega t - 2\omega\tau) d\tau \right) b \cos\left(\omega t - \psi\right) dt \\ &\approx \int_{-\infty}^{t} a(\tau) b \cos(\psi - 2\omega\tau) d\tau = \int_{-\infty}^{t} a(\tau) f_{\psi,\omega}(\tau) d\tau. \end{aligned} \tag{5}$$

When sinusoidal reference signals with different frequencies ω_l and phases ψ_k are measured sequentially, one obtains multiple channels of raw measurements. The multi-channel measurements at any pixel $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}$ can be interpreted as a point in a multi-dimensional space, and while difficult to model analytically, there is structure in this space that can be exploited through learning.

In the ideal case of a single bounce, the set of all possible measurement vectors $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}$ forms a "single-bounce measurement manifold" defined by Eq. 2. If only one frequency is used, measurements affected by MPI lie on the same manifold, and it is therefore impossible to identify and correct them. On the other hand, in the case of multiple frequencies, the manifold becomes a non-linear subspace, and MPI-affected vectors do not lie on it anymore. The MPI problem can then be recast as one of mapping real measurements, possibly affected by MPI, to the ideal one-bounce manifold, which is also the idea behind many existing approaches for MPI correction.

3.3 The Impact of Motion

Real scenes are rarely static. Because of the lateral and axial motion of objects with respect to the camera, sequential correlation measurements $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}$ are misaligned. Moreover, the axial component of the motion also changes the scene response function $a(\tau)$: for example, even in the simple case of a single bounce, the term τ_0 in Eq. 2 changes with the axial motion of the object, whereas the measured intensity varies proportionally to the inverse-square of distance. In our indoor experimental setting we found both these phenomena to contribute significantly to the depth reconstruction error. Motion can even generate blur and Doppler within each raw correlation measurement, but we found these last effects negligible when compared to the previous ones.

3.4 Characterization of Kinect 2

The Kinect 2 is a well-documented, widely used ToF camera, with an open-source SDK (libfreenect2 [1]) that exposes raw correlation measurements and provides a baseline algorithm for benchmarking (indicated as LF2 in the following). We carefully characterize the camera functions, shot noise, vignetting, and per-pixel delay, to produce accurate simulations and mitigate the data shift between synthetic data from the FLAT dataset and real scenarios.

The Kinect 2 uses three modulation frequencies, each with three phases, for a total of nine camera functions $\boldsymbol{f}_{\boldsymbol{\psi},\boldsymbol{\omega}}(\tau)$. To calibrate the camera functions, we carefully align the optical axis to be normal to a Lambertian calibration plane. We place a light absorption tube in front of Kinect's light source to narrow down the beam emission angle and limit MPI. We translate the plane to known distances $\{d_j\}_{j=1..N}$ to obtain a series of raw measurements $\{\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}(d_j)\}_{j=1..N}$ that approximate $(d_j)^{-2}\boldsymbol{f}_{\boldsymbol{\psi},\boldsymbol{\omega}}(2d_j/c)$ up to a constant scale, for every pixel. After removing the squared-distance intensity decay $(d_j)^{-2}$, we have a series of observations of the camera functions $\{\boldsymbol{f}_{\boldsymbol{\psi},\boldsymbol{\omega}}(2d_j/c)\}$. Fig. 2 shows

three camera functions fitted by this method, parameterized as $b\cos(\psi - 2\omega\tau)$ (red and blue curve in Fig. 2), and $\max(\min(b_1\cos(\psi - 2\omega\tau), b_2), -b_2)$ (green curve in Fig. 2).

As for the shot noise, we assume each pixel to be independent from the others. We acquire data from 15 scenes; for each raw correlation measurement, we compute the per-pixel expected value as the average of 100 measurements. For any expected value, we collect the empirical distribution of the noisy samples in a lookup table, that is used to generate noisy data in simulation.

For the complete explanation of the calibration procedure, including vignetting and per-pixel time delay, we invite the reader to refer to the Supplementary, which also includes more experimental results.

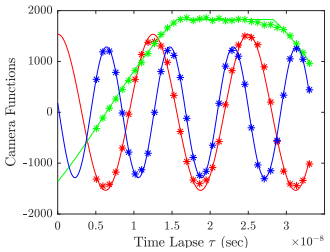


Fig. 2: Calibrated camera functions for Kinect 2; the points represent experimental data, the continuous line are the fitted camera functions.

4 The FLAT Dataset

An ideal dataset for training and validating ToF algorithms is large; allows simulating different camera response functions (like those in Section 3.4); allows including MPI, shot noise and motion; and exposes raw correlation measurements. We created the FLAT dataset with these principles in mind.

FLAT contains 2000 scene response functions, $\{a^j(\tau, x, y)\}_{j=1..2000}$, where we make the dependence on pixel (x, y) explicit. Each of these is computed through transient rendering [20], which simulates the irradiance received by the camera sensor at every time frame, after sending an impulse of light to the environment. The output of the renderer is a $n_\tau \times n_x \times n_y$ tensor, i.e. a discretized version of $a^j(\tau, x, y)$. The scenes in the dataset are generated from 70 object setups; each setup has 1 to 5 objects with lambertian surface and uniform albedo; their 3D models are from the Stanford 3D Scanning Repository [22] and the online collection [23]. We render each setup from approximately 30 point of views and orientations, at a spatial resolution of $(n_x = 424) \times (n_y = 512)$ pixels and for $n_\tau = 1000$ consecutive time intervals (each interval is $5e^{-11}$sec long); the horizontal field of view is 70 degrees (corresponding to the Kinect 2 camera). Since bi-directional path tracing is used to sample and simulate each light ray, $a^j(\tau, x, y)$ does simulate MPI. From the discretized version of $a^j(\tau, x, y)$, any raw measurement $i_{\psi,\omega}$ can be obtained as in Eq. 5, for any user-provided camera function $f_{\psi,\omega}(t)$ (like, for instance, the ones we estimated for Kinect 2).

The FLAT dataset offers the possibility to augment the raw measurement with shot noise, textures, vignetting, and motion. Within FLAT, we provide the code to apply any vignetting function and shot noise coherently with the simulated camera, while MPI and camera functions are handled as described in the previous paragraph. As a consequence, a physically correct simulation of differ-

ent camera functions, MPI, vignetting, and shot noise is a computationally light task within FLAT. On the other hand, texture and motion are more expensive to render exactly. Since each scene in the FLAT dataset takes on the average 10 CPU hours to render, creating a large set of scenes with different textures and motions would require tens to hundreds of times longer. We handle this by providing tools to approximate texture and motion in real time (as specialized forms of data augmentation), while still providing a small testing set within FLAT with exact motion, texture, and rendering.

We approximate textures on the training data, by pixel-by-pixel multiplication of the rendered $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}$ with texture maps from the CURET texture dataset [24]. This is an approximation that ignores the recursive impact that real textures have on MPI, but we have found that it is nonetheless useful as a form of data augmentation.

The FLAT dataset offers two different methods to augment the simulated raw measurements with approximate motion. To illustrate the first one, let us consider the Kinect 2, where nine correlation measurements of a static scene, $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}} = (i_{\psi_1,\omega_1}, \ldots, i_{\psi_9,\omega_9})$ are simulated. We generate a random 2D affine transform T and apply it to create a smooth motion as $i'_{\psi_j,\omega_j}(x,y) = i_{\psi_j,\omega_j}(T^{j-5}(x,y))$, where $T^n(x,y)$ is transforming (x,y) by T for n times, and $T^{-n}(T^n(x,y)) = (x,y)$. Notice that the first and last measurement will achieve the largest movement. To obtain a more complex movement, we simulate the motion of two or more objects with different affine transforms and composite the scene based on their depths. This approximate motion model is fast, but does not reproduce the MPI interaction between the objects in the scene. The second motion approximation method takes in input a rendered scene response function, $a(\tau,x,y)$. We generate then a random, 3D affine transformation and apply the corresponding displacement (v_x, v_y, v_z) to it, i.e., $a'(\tau,x,y) = a(\tau + v_z/c, x + v_x, y + v_y)$. Then we use Eq. 5 to compute one of the nine raw measurements. As in the previous method, the transform is applied multiple times to create a smooth motion between the nine measurements. This method is computationally more expensive compared to the previous one, but physically more accurate.

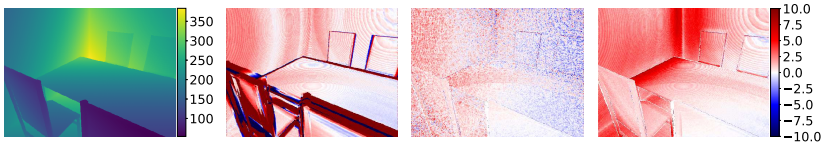
5 Network Architecture

We propose a two-module DNN (Fig. 1), to deal with MPI, motion, and shot noise in the domain of raw correlation measurements. We demonstrate the use of the modules on data from the Kinect 2. The two modules can be integrated into the LF2 [1] reconstruction pipeline, which is basically an implementation of Eq. 4 on Kinect 2. We leverage the differentiability of Eq. 4 to exploit, among other forms of training for the DNN, training using a loss function in the depth domain.

The first module (MOtion Module, MOM) is an encoder-decoder inspired by Flownet [7]. The aim of MOM is to output a velocity map to align the nine raw channels measured by Kinect2. Differently from the original design of Flownet, which computes the optical flow between two images in the same

Name	MOM + LF2	MOM-MRM + LF2*	MRM + LF2*
Training Data	Motion, MPI	Motion, MPI	Static, MPI
Architecture	Encoder-Decoder	Encoder-Decoder-KPN($3 \times 3 \times 1$)	KPN($1 \times 1 \times 9$)
Depth Reconstruction	LF2	LF2, no filter	LF2, no filter
Training Loss	L2, Velocity	L2, Raw Intensity	L2, Raw Intensity
Fine Tuning Loss		L2, Depth	L2, Depth

Table 1: Training specification.



(a) True depth (b) Motion (c) Shot noise (d) MPI

Fig. 3: Effect of non-idealities on the LF2 depth reconstruction error, in cm.

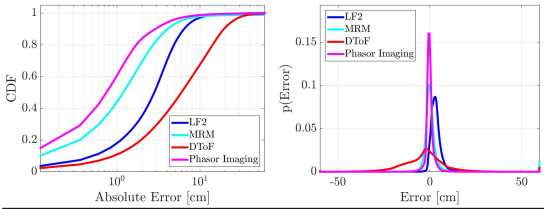
domain, MOM aligns raw correlation measurements taken with different camera functions, and therefore correlated, but visually different. Moreover, MOM takes in input nine misaligned channels and outputs eight optical flows at the same time, while Flownet deals with only one pair of images and one optical flow. The second module (Multi-Reflection Module, MRM) is based on a kernel-predicting network (KPN), that has been effectively used for burst denoising on shot noise [25]. MRM outputs nine spatially varying kernels for every pixel; each kernel is locally convolved with the input raw measurements, to produce a clean raw measurement by jointly removing shot and MPI noise on every channel.

Table 1 shows the details of different variations of the basic DNN architecture that we consider in our experiments. Notice that, when we use the MRM module, we modify the LF2 pipeline to remove bilateral filtering from it, because denoising is already performed by MRM; we indicate this variation of the LF2 pipeline as LF2*. The MOM-MRM (followed by LF2*) network inherits the encoder-decoder from MOM; training of MRM is performed with the output of MOM as input, the weights of MOM being fixed. We start training using the L2 error of the raw correlation measurements as loss. Then, we fine tune MRM using the L2 depth loss propagated through LF2*. Motion in the training data is generated using the first approximation method in the FLAT dataset. We also tried fine tuning using the second approximation, which is physically more accurate, obtaining similar results.

6 Experiments

To better illustrate the typical distribution of the artifacts introduced by motion, shot noise, and MPI, we show in Fig. 3 a scene from the FLAT dataset, rendered with motion, shot noise, or MPI only, and then reconstructed by the LF2 pipeline. Over/under shootings can be observed at the border of moving

objects, where raw measurements from the foreground and the background mix due to motion. Shot noise in $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}$ creates random noise in the depth map, especially in dark regions like background and image borders. MPI generates a low frequency noise in areas affected by light reflection, like the wall corner in Fig. 3.



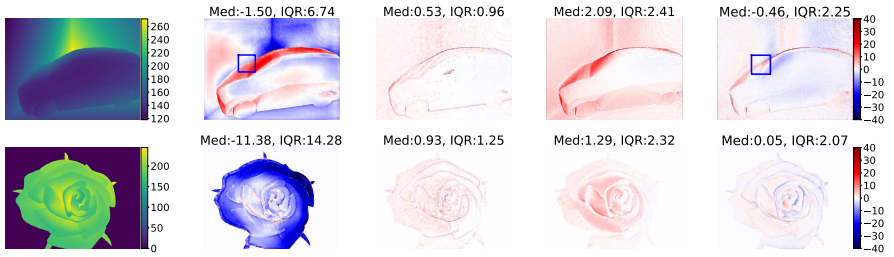
	Median (error)	IQR	Prctile (90^{th}) (absolute error)
LF2	2.92 (1.62%)	3.01 (2.34%)	6.55 (5.36%)
MRM	-0.01 (0.00%)	2.63 (1.70%)	4.19 (2.52%)
DToF	-2.48 (1.80%)	12.36 (7.13%)	19.56 (9.84%)
Phasor	-0.29 (0.12%)	1.62 (0.71%)	1.83 (1.16%)

Fig. 4: The upper left panel shows the CDF of the depth error for LF2, MRM, DToF, and Phasor, for simulated data from FLAT, affected by shot noise and MPI. The upper right panel shows the histogram of the error. All pixels whose real depth is in the [1.5m, 5m] range have been considered. Our MRM outperforms LF2 and DToF, and it comes much closer to the accuracy achieved using the MPI-dedicated, higher-frequency modulations of Phasor. The table shows corresponding median and Inter Quartile Range (IQR) of the depth error, and 90^{th} percentile of the absolute error, in cm. The numbers in the brackets indicate the relative errors.

6.1 MPI Correction

We first measure the effect of the MRM module on static scenes affected by MPI in the FLAT dataset, and compare it to LF2 [1], DToF [4], and Phasor [18], that are based respectively on multi-frequency, deep learning, and custom hardware. LF2 implements Eq. 4 on Kinect 2 and it constitutes our baseline to evaluate the improvement provided by our DNN on the same platform. DToF and Phasor require different sensor platforms than Kinect 2, but thanks to the flexibility of the FLAT dataset, we can simulate raw measurements using their specific modulation frequency and phase, and add the same level of noise for testing. As DToF and Phasor do not mask unreliable output pixels (like LF2 does), and Phasor's working range is limited to [1.5m, 5m], we compare the depth error for all those pixels in this range only. Furthermore, as Phasor does not deal with shot noise, we apply a bilateral filter to remove noise from its output depth.

Results reported in Fig. 4 show that DToF produces the less accurate depth map. The median error of LF2 is biased because of the presence of MPI in the raw data—in fact, the LF2 pipeline does not include any mechanism to correct such non-ideality. This bias is effectively removed by MRM. Our method approaches the Phasor's accuracy, without requiring expensive hardware to create very high modulation frequencies (1063.3MHz and 1034.1MHz): MRM works with Kinect 2, which uses frequencies below 50MHz. Fig. 5 shows the results of typical scenes from the FLAT dataset, where Phasor and MRM outperform other methods in



(a) True (b) DToF (c) Phasor (d) LF2 (e) MRM

Fig. 5: Depth error for scenes from the FLAT dataset, corrupted by shot noise and MPI and reconstructed by DToF [10], Phasor Imaging [18], LF2 [1] and MRM, in cm. Errors are computed only in the unanbiguous reconstruction range of Phasor Imaging, [1.5m, 5m]; no mask is used to remove unreliable pixels. The blue boxes in the first row show the receptive field for DToF and MRM.

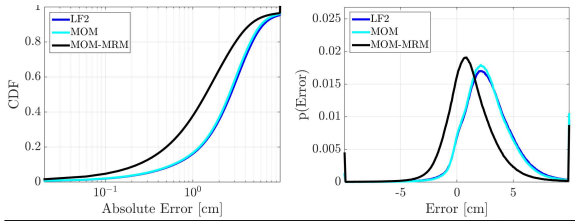
removing MPI. It is worth noticing that high frequency modulation signals, like those used in Phasor, are very susceptible to noise. In fact, we use a bilateral filter to reduce the effect of shot noise on the output of Phasor. Although this effectively reduces random noise, any misalignment on raw correlation measurements (like the one occurring in case of motion) creates a systematic noise that cannot be eliminated by bilateral filtering, which dramatically reduces the accuracy of Phasor (Fig. 7c). Our MRM appears much more reliable in this situation (Fig. 7e).

The case of a real scene of a corner, acquired in static conditions with a Kinect 2, is depicted in Fig. 8. The ground truth shape of the scene could be estimated by checkerboard calibration applied to each of the three planes of the corner. This figure shows that MRM can significantly reduce MPI artifacts (compared to LF2) not only in simulation, but also in realistic conditions.

6.2 Motion Artifact Correction and Ablation Study

We perform an ablation study to quantify the benefits of MOM and MOM-MRM, on a test set from the FLAT dataset corrupted by MPI, shot noise, and random motion. For this experiment, the motion between the nine correlations measurement is fully simulated by moving the objects in 3D space. This allows testing the MOM and MOM-MRM on simulated raw correlation measurements affected by a real motion field, even if the modules are trained on approximated motion data. We compare depth reconstruction through LF2, MOM, and MOM-MRM, each using the same masking method provided by LF2 to eliminate unreliable pixels, like those along misaligned object boundaries due to motion. The density (i.e., percentage of reconstructed pixels) reported in Fig. 6 is

therefore representative of how well objects boundaries are re-aligned by MOM. The depth accuracy is slightly higher for MOM compared to LF2, but the main advantage for MOM is a reduction of the unreliable pixels, as density increases from 93.56% to 95.50%. Red boxes in Fig. 9 demonstrate in simulation how introducing the MOM module can reduce the presence of holes in the reconstructed scene, especially close to object boundaries. The introduction of the MRM module further increases the density and reduces the bias in the depth error caused by MPI. Also this effect is clearly visible in the green boxes in Fig. 9, where the introduction of the MRM module leads to the reduction of the MPI artifact in the corner of the room.



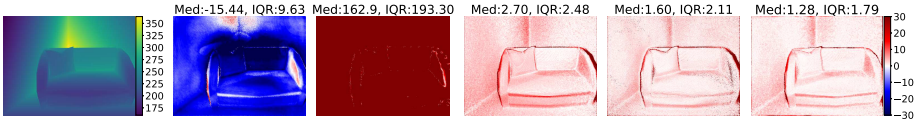
	Median (error)	IQR	Prctile (90^{th}) (absolute error)	Density
LF2	2.57 (1.11%)	2.65 (1.14%)	5.62 (2.51%)	93.56%
MOM	2.50 (1.08%)	2.55 (1.10%)	5.48 (2.44%)	95.50%
MOM-MRM	1.02 (0.45%)	2.43 (1.06%)	4.12 (1.82%)	97.67%

Fig. 6: The upper left panel shows the CDF of the depth error for LF2, MOM and MOM-MRM, for simulated data from the FLAT dataset, affected by shot noise, MPI, and motion. The upper right panel shows the histogram of the error. Only those pixels that have been reconstructed contribute to the statistics. The table shows the corresponding reconstruction errors (median, 90^{th} percentile, and Inter Quartile Range (IQR), in cm). The numbers in the brackets indicate the relative errors.

6.3 Putting Everything Together

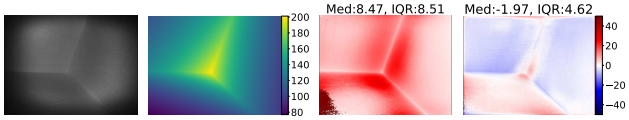
Fig. 7 shows a simulation from the FLAT dataset, where the scene has been corrupted by shot noise, MPI, and a small motion (an average of 10 pixels between the first and last raw correlation measurements). Phasor imaging cannot produce a reliable depth map in this case: even a small motion changes the measured phase significantly, because of the fast modulation frequency. MRM still outperforms LF2, reducing the MPI, whereas the architecture trained to correct both motion and MPI, MOM-MRM, performs best (lowest median and IQR). The reconstruction of the depth scene using the MOM-MRM approach takes approximately 90ms on a NVIDIA TitanV GPU.

Our DNN architecture, which operates on raw measurements, is the result of a thorough investigation. We tested several architectures, including one that directly outputs depth as suggested in the recent work by Su et al. [26], but the reconstruction error was consistently larger than that of the proposed DNN. We believe this is due to the fact a DNN that outputs the depth directly would be forced to learn the non-linear mapping from raw measurements to depth, instead of leveraging the fact that such relation is dictated by physics and known (see the inverse-tangent of Eq. 4). An additional benefit of working in the raw domain is



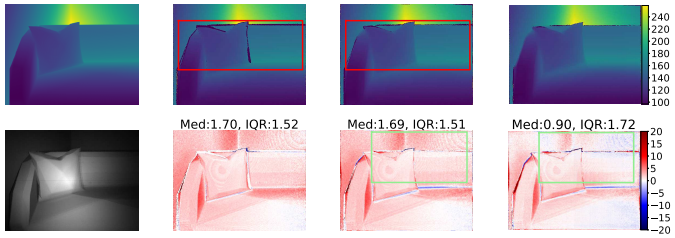
(a) True (b) DToF (c) Phasor (d) LF2 (e) MRM (f) MOM-MRM

Fig. 7: Depth error for different reconstruction methods and a simulated scene from FLAT, corrupted by shot noise, MPI, and small motion. Units in cm.



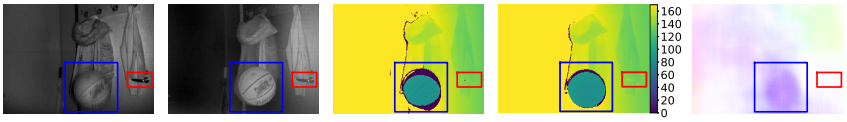
(a) Scene (b) True (c) LF2 (d) MRM

Fig. 8: A real scene of a corner captured by a Kinect 2 (a) with ground truth depth (b) and depth errors for LF2 (c), and MRM (d). MPI artifacts show up as lobes close to the corner in LF2, and are reduced by MRM. Units in cm.



(a) True (b) LF2 (c) MOM (d) MOM-MRM

Fig. 9: A simulated scene from FLAT, corrupted by shot noise, MPI, and motion. The upper row shows the depth maps. The lower left panel is the intensity image, other panels in the row are depth errors. MOM aligns object boundaries and allows a more dense reconstruction (red boxes in b, c). MRM mostly corrects MPI artifacts in the smooth areas (green boxes in c, d). Units in cm.



(a) Scene (b) Raw (c) LF2 (d) MOM-MRM (e) Flow

Fig. 10: Panel (a) shows the average of nine raw correlation measurements acquired by Kinect 2, with a moving ball (blue box); panel (b) shows one of the raw measurements. Our method (d) reduces the motion artifacts compared to LF2 (c). The optical flow generated from MOM is shown in panel (e). The red box highlights the reflective hand bar with specular reflections; if not masked out, our method fails on these pixels. Units in cm.

that, beyond depth, we can estimates uncertainty as in the LF2 pipeline, which is useful, for instance, to mask out bad depth values.

Fig. 10 shows an example of a real scene with moving objects (ball in the blue box), captured by Kinect 2. Notice that Fig. 10a is blurred as it averages nine raw measurements, whereas each individual raw measurement is still sharp, as in Fig. 10b. Coherently with simulations, the main effect of MOM is the reduction of the holes close to the boundaries of the moving object.

6.4 Method Limitations

Our method has some limitations that could be overcome with further development. The receptive field of MRM (blue box in Fig. 5) is 72×72 pixels, in theory not large enough to capture global geometric information and correct long-range MPI; furthermore, our loss function naturally emphasizes short-range MPI correction because the signal is substantially stronger for shorter traveling distances. Nonetheless, MRM does reduce many long-range MPI artifacts (see the windshield of the car in Fig. 5). This can be explained with the a priori information about object appearance learned by MRM, or assuming that MRM learns to project measurements corrupted by long-range MPI onto the single-bounce manifold (details in the Supplementary).

Another limitation is that FLAT only includes diffuse materials. Therefore MRM cannot reconstruct surfaces with strong specular reflections, as the door handle (red box) in Fig. 10.

Fig. 10 also highlights the limitations of MOM for large motions: while MOM effectively warps pixels of the foreground moving object, it is not designed to inpaint missing data of the partially occluded background. This results in only partial elimination of the motion artifacts for large motion fields.

As our experiments with Kinect 2 and rapid motion showed it to be negligible (Fig. 10b), we did not consider blur within a single raw measurement. In the case motion blur becomes significant for other platforms, approximate blur can be easily included when simulating measurements from FLAT.

Lastly, our method assumes constant ambient light (as in typical indoor conditions) to model the camera noise. Characterizing the noise induced by ambient light separately may lead to a more accurate noise model.

7 Conclusion

Motion, MPI and shot noise can significantly affect the accuracy of depth reconstruction in ToF imaging. We have shown that deep learning can be used to reduce motion and MPI artifacts jointly, on an off-the-shelf ToF camera, in the domain of raw correlation measurements. We demonstrated the effectiveness of our MOM and MRM modules through an ablation study, and reported results on both synthetic and real data. Alternative methods to tackle these artifacts are still to be explored; we believe that our flexible FLAT dataset represents a helpful instrument along this research line.

References

1. Xiang, L., Echtler, F., Kerl, C., Wiedemeyer, T., Lars, hanyazou, Gordon, R., Facioni, F., laborer2008, Wareham, R., Goldhoorn, M., alberth, gaborpapp, Fuchs, S., jmtatsch, Blake, J., Federico, Jungkurth, H., Mingze, Y., vinouz, Coleman, D., Burns, B., Rawat, R., Mokhov, S., Reynolds, P., Viau, P., Fraissinet-Tachet, M., Ludique, Billingham, J., Alistair: libfreenect2: Release 0.2 (April 2016)
2. Bhandari, A., Raskar, R.: Signal processing for time-of-flight imaging sensors: An introduction to inverse problems in computational 3-d imaging. IEEE Signal Processing Magazine (2016)
3. Feigin, M., Bhandari, A., Izadi, S., Rhemann, C., Schmidt, M., Raskar, R.: Resolving multipath interference in kinect: An inverse problem approach. sensors (2016)
4. Marco, J., Hernandez, Q., Muñoz, A., Dong, Y., Jarabo, A., Kim, M.H., Tong, X., Gutierrez, D.: DeepToF: Off-the-shelf real-time correction of multipath interference in time-of-flight imaging. In: ACM Transactions on Graphics (SIGGRAPH ASIA). (2017)
5. Mutny, M., Nair, R., Gottfried, J.: Learning the correction for multi-path deviations in time-of-flight cameras. arXiv preprint arXiv:1512.04077 (2015)
6. Son, K., Liu, M., Taguchi, Y.: Automatic learning to remove multipath distortions in time-of-flight range images for a robotic arm setup. arXiv preprint arXiv:1601.01750 (2016)
7. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., v.d. Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). (2015)
8. Bako, S., Vogels, T., McWilliams, B., Meyer, M., Novák, J., Harvill, A., Sen, P., DeRose, T., Rousselle, F.: Kernel-predicting convolutional networks for denoising monte carlo renderings. In: ACM Transactions on Graphics (SIGGRAPH). (2017)
9. Jiyoung Jung, Joon-Young Lee, I.S.K.: Noise aware depth denoising for a time-of-flight camera. In: Korea-Japan Joint Workshop on Frontiers of Computer Vision. (2014)
10. Jarabo, A., Masia, B., Marco, J., Gutierrez, D.: Recent advances in transient imaging: A computer graphics and vision perspective. Visual Informatics (2017)
11. Ferstl, D., Reinbacher, C., Riegler, G., Rther, M., Bischof, H.: Learning depth calibration of time-of-flight cameras. In: Proceedings of the British Machine Vision Conference (BMVC). (2015)
12. Lenzen, F., Kim, K.I., Schäfer, H., Nair, R., Meister, S., Becker, F., Garbe, C.S., Theobalt, C.: Denoising strategies for time-of-flight data. In: Time-of-Flight and Depth Imaging. (2013)
13. Gottfried, J.M., Nair, R., Meister, S., Garbe, C.S., Kondermann, D.: Time of flight motion compensation revisited. In: IEEE International Conference on Image Processing (ICIP). (2014)
14. Lee, S.: Time-of-flight depth camera motion blur detection and deblurring. IEEE Signal Processing Letters (2014)
15. Hansard, M., Lee, S., Choi, O., Horaud, R.: Time-of-Flight Cameras: Principles, Methods and Applications. Springer Publishing Company, Incorporated (2012)
16. Heide, F., Heidrich, W., Hullin, M., Wetzstein, G.: Doppler time-of-flight imaging. In: ACM Transactions on Graphics (SIGGRAPH). (2015)

17. Freedman, D., Krupka, E., Smolin, Y., Leichter, I., Schmidt, M.: SRA: fast removal of general multipath for tof sensors. arXiv preprint arXiv:1403.5919 (2014)
18. Gupta, M., Nayar, S.K., Hullin, M.B., Martin, J.: Phasor imaging: A generalization of correlation-based time-of-flight imaging. ACM Transactions on Graphics (2015)
19. Nair, R., Meister, S., Lambers, M., Balda, M., Hofmann, H., Kolb, A., Kondermann, D., Jähne, B.: Ground truth for evaluating time of flight imaging. Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications (2012)
20. Jarabo, A., Marco, J., Muñoz, A., Buisan, R., Jarosz, W., Gutierrez, D.: A framework for transient rendering. In: ACM Transactions on Graphics (SIGGRAPH ASIA). (2014)
21. Gushov, V., Solodkin, Y.N.: Automatic processing of fringe patterns in integer interferometers. Optics and Lasers in Engineering (1991)
22. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM (1996) 303–312
23. Burkardt, J.: Obj files: A 3d object format. `https://people.sc.fsu.edu/~jburkardt/data/obj/obj.html` (2016)
24. Dana, K.J., Van Ginneken, B., Nayar, S.K., Koenderink, J.J.: Reflectance and texture of real-world surfaces. (1999)
25. Mildenhall, B., Barron, J.T., Chen, J., Sharlet, D., Ng, R., Carroll, R.: Burst denoising with kernel prediction networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
26. Su, S., Heide, F., Wetzstein, G., Heidrich, W.: Deep end-to-end time-of-flight imaging. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2018)