Unsupervised Domain Adaptation for ToF Data Denoising with Adversarial Learning

Gianluca Agresti
University of Padova
agrestig@dei.unipd.it

Henrik Schaefer
Sony Europe B.V.
henrik.schaefer@sony.com

Piergiorgio Sartor
Sony Europe B.V.
sartor@sony.de

Pietro Zanuttigh
University of Padova
zanuttigh@dei.unipd.it

Abstract

Time-of-Flight data is typically affected by a high level of noise and by artifacts due to Multi-Path Interference (MPI). While various traditional approaches for ToF data improvement have been proposed, machine learning techniques have seldom been applied to this task, mostly due to the limited availability of real world training data with depth ground truth. In this paper, we avoid to rely on labeled real data in the learning framework. A Coarse-Fine CNN, able to exploit multi-frequency ToF data for MPI correction, is trained on synthetic data with ground truth in a supervised way. In parallel, an adversarial learning strategy, based on the Generative Adversarial Networks (GAN) framework, is used to perform an unsupervised pixel-level domain adaptation from synthetic to real world data, exploiting unlabeled real world acquisitions. Experimental results demonstrate that the proposed approach is able to effectively denoise real world data and to outperform state-of-the-art techniques.

1. Introduction

Among the various solutions for depth data acquisition, Time-of-Flight (ToF) sensors have attracted a large interest since they are able to reliably get the depth information at interactive frame rates. These sensors estimate the depth by illuminating a scene with a periodic, amplitude modulated light signal and by estimating the time taken by the signal to reach the scene points and come back from the phase displacement between the transmitted and received signal [33]. Even if the technology behind these sensors has improved a lot since their introduction, it is still affected by several critical issues, including limited spatial resolution, relatively high noise levels (specially on dark surfaces) and inaccuracies on the edges due to the mixed pixel effect. A particularly critical issue is the so-called Multi-Path Interference (MPI), due to the fact that the emitted light can bounce multiple times in the scene before reaching to the sensor. This leads to a depth overestimation that is scene dependent and related to both the geometry and the material properties.

Since the MPI error is related to the modulation frequency of the ToF signal, by using multi-frequency ToF (MF-ToF) sensors, useful clues for its estimation can be extracted and used for ToF data denoising. However, traditional approaches based on this idea do not have completely satisfactory performances. Recently, machine learning techniques and in particular Convolutional Neural Networks (CNN) have been exploited for this task [5, 30, 14]. Even if these approaches have obtained reasonable performances in some simple situations, they are not able to completely remove the MPI corruption and their generalization capabilities are limited. The main issue in using deep learning techniques for this task is the limited amount of available training data. There are no large public datasets, like for image classification or semantic segmentation. Furthermore, while ToF depth data can be easily acquired, getting the corresponding ground truth information is a very time consuming task, requiring the acquisition of the scene with highly accurate scanning equipments and the registration of ground truth data with the ToF acquisition.

A possible solution to overcome this issue, is to train the deep network with synthetic data produced by a Time-of-Flight simulator. The approach of [5] exploits this idea and obtains impressive performances on synthetic data. On the other hand, the differences between the real world and simulated data, reduce the performances on real sensors, where the approach is able to reduce the MPI corruption but does not completely remove it.

This paper introduces a novel transfer learning architecture able to properly denoise real world data by exploiting unlabeled real world ToF acquisitions (i.e., without the ground truth data) to adapt the training performed on synthetic data to the real world setting. More in detail, a Coarse-Fine CNN exploiting MF-ToF data derived from the one introduced in [5] is jointly trained in a supervised way on a dataset composed by synthetic scenes, for which the ground truth depth is known and in an unsupervised way on a real dataset, for which the depth ground truth is unknown. For the unsupervised training, we use an adversarial loss and a learning framework based on the Generative Adversarial Network (GAN) model. The generator network performing ToF data denoising is trained along with a discriminator network, implementing the adversarial loss used for the unsupervised domain adaptation of the generator. This method contains several novel contributions. First of all, it introduces a novel adversarial learning framework for domain adaptation in regression problems and it is the first to apply this technique to the denoising of depth data. In the proposed framework, couples of noisy depths and error maps, computed from ground truth or from generated data, are used to better capture the joint statistics of the noisy data and the denoised ones. This avoids the deviation of the generator from the input data. Furthermore, a novel data augmentation strategy based on *noise augmentation* is introduced. The effectiveness of the proposed approach is demonstrated by the evaluation on two different real world datasets.

2. Related Works

ToF sensors suffer from different noise sources [33, 20, 22, 19] as: thermal noise due to the electronics of the sensor; Photon Shot Noise (PSN) related to the random nature of the light; systematically distorted depth estimation due to the non-ideality of the emitted light signals; MPI due to the multiple reflections of the emitted light before coming back to the ToF sensor. Bilateral filtering or total variation techniques could be used to reduce the zero mean errors (PSN, thermal) [21, 5], while MPI is a more critical issue for ToF sensors. Many methods for MPI correction have been proposed [31], but it remains an open problem.

The methods based on single modulation frequency acquisitions employ a suitable reflection model and exploit the corrupted structure of the scene to estimate its true geometry [10, 11, 18].

The methods based on multi-frequency ToF acquisitions impose a hypothesis on the composition of the backscattered light, e.g. assuming it is composed by few specular rays. Freedman et al. [9] proposed an optimization based approach, using data acquired at 3 modulation frequencies. The method presented by Bhandari et al. [6] corrects MPI due to K interfering rays using $2K+1$ modulation frequencies with a closed form solution.

Other methods use a modified ToF light source that projects a sequence of spatial patterns onto the scene to separate the direct light, reflecting only once inside the scene, from the interfering rays, the so called global light [32, 4, 25, 2].

In order to avoid the use of an explicit reflection model, data-driven approaches correcting MPI have been presented recently. Son et al. in [29] use a deep neural network, trained on labeled real data, captured from a robotic arm on short range scenes. Since the acquisition of a dataset composed by ToF depth with a registered ground truth is challenging and expensive, Marco et al. in [24] trained an encoder-decoder CNN in an unsupervised way on real depth maps. Then they trained the decoder part only in a supervised way on a synthetic ToF dataset. Recently, deep learning techniques using end-to-end CNNs, taking raw ToF correlation samples as input and outputting the refined scene depth map, have been presented for general purpose ToF denoising [30, 14, 5]. In these methods, the CNNs have been trained on synthetic data, but the adaptation to real data has been investigated only from a qualitative point of view in [30] and on a single corner scene in [14]. In the method of Agresti et al. [5], a CNN for MPI correction is trained on multi-frequency synthetic data. By presenting a wider quantitative performance evaluation on real data, this paper shows some limitations due to the domain shift between the training and testing domain. Finally, a closely related field, where deep learning strategies have been successfully exploited, is the fusion of stereo and ToF data [3, 26].

This paper starts from the method introduced in [5], but uses an unsupervised domain adaptation technique to overcome the synthetic-real data domain shift. We acquired unlabeled real data, avoiding the expensive task of ground truth acquisition, and we used them in combination with the labeled synthetic data in the proposed domain adaptation framework.

Domain adaptation is a growing research area. A milestone of this field is the work of Ganin et al. [12], which reduced the domain shift of a classifier in an unsupervised way by using a *domain classifier* trained to decide if the input features of the classifier are coming from the source domain or from the target domain. The *domain classifier* is used to realize an adversarial loss, similar to GAN [13], to apply the domain adaptation. Similarly, unsupervised domain adaptation of a pre-trained classifier is proposed in [8]. In [27], feature adaptation is used to adapt a network trained for multi-task regression (normal, edge and depth from color image) from synthetic to real data. In [28, 8] a generator network is used to modify some labeled synthetic data to look similar to the real data, which are then used to train a classifier. A pixel-level cycle-consistent domain adaptation scheme, that also ensures the semantic consistency in the domain translation, is proposed in [17].

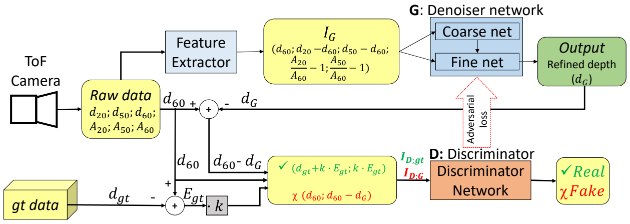

Figure 1: Architecture of the proposed approach.

3. Proposed Method

In order to denoise the acquired ToF depth data (i.e., remove both MPI and sensor noise), we use an improved version of the *Coarse-Fine* CNN of [5] that has been fitted into a novel adversarial learning framework, used to perform unsupervised domain adaptation from synthetic to real data.

The general architecture of the proposed machine learning strategy is shown in Fig. 1: the generator *Coarse-Fine* CNN (Section 4) takes different features extracted from the sensor raw data as input and produces an estimate of the noise-free depth map of the scene, while the discriminator network (Section 5) is used for the adversarial learning procedure of Section 6. In this section, we discuss the processing of input data, while the machine learning framework will be the subject of the next sections.

Recall that the phase offsets of the interfering rays causing MPI are frequency dependent and this *frequency diversity* can be used to understand if MPI is acting on MF-ToF cameras and can give cues for its correction [9, 7, 16].

Following this rationale, the scenes are captured with the ToF camera modulation frequency at $20,\ 50$ and $60\ \mathrm{MHz}$ and the depth maps have been phase unwrapped to extend the maximum unambiguous range up to $15\ \mathrm{m}$. The acquired information is pre-processed in order to extract a representation I_G that contains relevant information about the MPI presence and strength: 5 different feature channels have been extracted from the ToF data, thus obtaining the following input representation:

$$I_G = \left(d_{60}; d_{20} - d_{60}; d_{50} - d_{60}; \frac{A_{20}}{A_{60}} - 1; \frac{A_{50}}{A_{60}} - 1 \right) \quad (1)$$

where d_x and A_x are the ToF depth and amplitude maps, captured at x MHz. The idea is to exploit the frequency diversity and the fact that in presence of MPI, the modulated light interferes, resulting in a variation of the received signal amplitude when the modulation frequency changes (for more details see [5]) .

The input data I_G, without any pre-processing for denoising (differently from [5]), is fed to a generator network, i.e. a *Coarse-Fine* CNN architecture that produces the denoised depth map (Section 4). This network consists of two parts: a coarse network that takes I_G as input and estimates a low resolution version of the scene; a fine network that takes both I_G and the output of the coarse network as input in order to estimate the full resolution data. Note that the network directly produces the denoised data and not an estimation of the error as in [5], thus allowing to avoid the use of additional computationally intensive filtering steps, e.g. the bilateral filter used in other ToF denosing methods.

The discriminator CNN module (Section 5) instead is used to perform an unsupervised domain adaptation from synthetic to real data based on adversarial learning. This represents a key advancement of this work w.r.t. previous research on the topic [5, 14, 30, 24]. The discriminator is fed with a two channel input. The first channel is always the noisy ToF depth map. The second channel is an estimate of the ToF error (the deviation from the true depth).

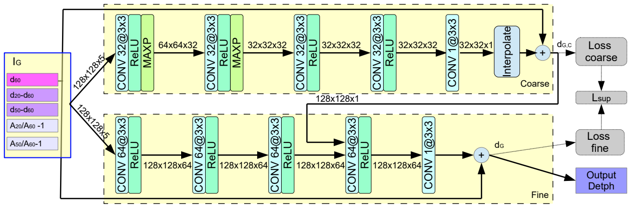

Figure 2: Architecture of the generator network G.

4. Generator Network Architecture

The architecture of the proposed generator network G is depicted in Fig. 2 and consists of a *Coarse-Fine CNN*. The choice of using 2 sub-networks is due to the fact that the reflections causing MPI can happen in different areas of the scene, thus requiring a wide field of view. On the other hand, wide convolutions and pooling layers cause blurring of edges and small details.

The coarse network allows to have a wide receptive field, since it applies downsampling with pooling layers after the first and second convolutional layers. It is made of a stack of 4 convolutional layers with 3×3 kernels and 32 filters. Each convolutional layer is followed by a ReLU. The last layer is a single 3×3 convolutional filter. The output of the coarse network is a low resolution estimate of the scene geometry (note that differently from [5], the network directly estimates the depth map and not the MPI corruption). The output of the last layer is finally up-sampled using a bilinear interpolation. The up-sampled output $d_{G,C} = G_C(I_G)$ is a coarse estimate of the refined depth map.

The fine network works at full resolution and allows to obtain an accurate representation of edges and details (the output of this network is denoted with $d_G = G(I_G)$ and is also the final output of the proposed method). The first 4 convolutional layers have 3×3 kernels, 64 filters, ReLU activation and no pooling. The output layer is a single 3×3 convolutional kernel. In order to exploit the global information, the up-sampled output of the coarse network ($d_{G,C}$) is given to the 4^{th} layer of the fine network as input.

5. Discriminator Network Architecture

In order to perform unsupervised domain adaptation, we use a discriminator Convolutional Neural Network (denoted with D). We want the discriminator to capture the relationships between the noisy depth data and the related noise image, in order to realize a discrimination of denoised depth maps produced from G from ground truth data. This will be used to drive the adversarial learning process in Section 6, that will force G to produce depth maps from synthetic and real data, that are correctly denoised and resemble the properties of ground truth data. As introduced in Section 3, the discriminator takes the noisy depth map d_n and the error map E (which can be the difference between the noisy depth map and the ground truth depth $E_{gt} = d_n - d_{gt}$ or between the noisy depth map and the generator output $E_G = d_n - d_G$) as input. The discriminator aims to capture the joint statistics of the couple $I_{D;gt} = (d_n; E_{gt})$, that is $(d_{gt} + E_{gt}; E_{gt})$ or equivalently $(d_n; d_n - d_{gt})$, giving output 1 if the input follows this distribution. Instead, we want the discriminator to discard all the data that does not follow the ground truth statistics and are generated by G. To clarify, the output of D should be 0, if the input is $I_{D;G} = (d_n, d_n - d_G) = (d_n, E_G)$ and 1 if the input is $I_{D;gt} = (d_n, d_n - d_{gt}) = (d_n, E_{gt})$. In an early version of the proposed work, we tried to use the standard approach of feeding D with d_{gt} as positive example, or the output of G, d_G, as negative example. After the domain adaptation, the generated data was not very close to the depth ground truth, since this approach left too much freedom to the generator. Thus, we employed the proposed two channel features. This choice forces D to focus on the raw ToF depth map and on how the estimated error is related to it, preventing the output of G to deviate from its input.

The architecture of the proposed discriminator network is shown in Fig. 3: it is made of a stack of 5 convolutional layers. The first 4 have 4×4 convolution kernel windows with a stride of 2 and 16, 32, 64 and 128 filters respectively. Each layer is followed by a batch normalization layer and a ReLU activation. The output layer has 1 filter and no ReLU and batch normalization. The discriminator can be trained by minimizing the following loss function:

$$L_D = -E\Big(\log\big(D(I_{D;gt})\big) + \log\big(1 - D(I_{D;G})\big)\Big) \quad (2)$$

Please note that we are using for the training of the whole system a synthetic dataset provided with the ground truth depth of the scenes (d^s_{gt}) and an unlabeled real dataset. In the rest of this paper, we will use the "*s*" and "*r*" apexes to distinguish between synthetic and real data.

In the *true* case $I_{D;gt}$ requires the ground truth d_{gt} and so it can be constructed only on the synthetic dataset. On the other hand, the *fake* data $I_{D;G}$ does not require ground truth information and can be constructed for both real and synthetic datasets.

In order to obtain better performance, we chose to train D on synthetic data only (note that real data will instead be used in the adversarial training procedure for G in Section 6). Otherwise, D would always recognize real data as fake, since they were always used as negative examples. This allows to avoid training the discriminator to distinguish between real and synthetic data instead of learning the statistics of $(d_n; E)$ in the correct way.

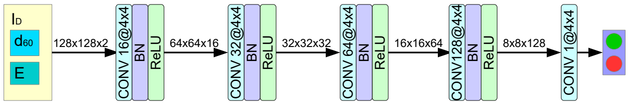

Figure 3: Architecture of the discriminator network D.

On the other side, the choice of using only synthetic data limits the capability of D to generalize to real data. One of the main causes for this is that the amount of noise on real data depends on several factors and can be slightly different from synthetic simulations. In order to better generalize and train a network that is able to adapt to different levels of noise, we apply a novel data augmentation strategy on $I^s_{D;gt}$. Using ground truth data we can separate data and noise on the training set and then produce different versions of the scene with slightly increased or decreased amounts of noise. The idea is to use as *true* input for D the couple

$$I^s_{D;gt} = (d^s_{gt} + E'_{gt}; E'_{gt}) \quad (3)$$

with E'_{gt} given by

$$E'_{gt} = k \cdot (d^s_{60} - d^s_{gt}) = k \cdot E^s_{gt}, \quad (4)$$

where k represents a uniform random variable in the range $[1 - \epsilon; 1 + \epsilon]$ that acts as a scaling factor for the noise on simulated data. The parameter ϵ has been set to 0.5 for optimal domain adaptation performance using k-fold validation. This data augmentation strategy leads to a wider and more general data distribution of which the synthetic statistics is a subset. It forces D to learn more generic pairs of *(noisy depth; error image)*, preventing it from focusing too much on synthetic ToF statistics. Doing so, D learns to judge how well the error map from G fits to the noisy ToF depth.

6. Adversarial Learning Strategy

The denoising network G is trained both with synthetic data in a supervised way and with unlabeled real data in an unsupervised way. The discriminator D is used to implement an adversarial loss to perform an unsupervised domain adaptation to real world scenes on G. More in detail, the supervised training is performed with the patches extracted from the synthetic dataset S_1 (see Section 7) and allows to obtain good performance on synthetic scenes, but the photometric differences between simulated and real world data makes this training not very effective on real data. For this reason, the unlabeled real dataset is used to train G by using the adversarial loss from the discriminator. G is trained by minimizing a loss function composed of 2 parts:

$$L_G = L_{sup} + w \cdot L_{adv}, \quad (5)$$

where

$$L_{sup} = E[|d^s_G - d^s_{gt}|] + E[|d^s_{G,C} - d^s_{gt}|] \tag{6}$$

$$L_{adv} = E[-\log\big(D(I^r_{D;G})\big)]. \tag{7}$$

The first term is optimized in a supervised way on synthetic data only (dataset S_1, Section 7). It is modeled as the sum of the l_1 distances between the outputs of G (i.e., the output $d^s_G = G(I^s_G)$ of the fine network and the output $d^s_{G,C} = G_C(I^s_G)$ of the coarse one) and the ground truth depth. Note that considering also the output of the coarse network allows to properly train also this module, that is fundamental to understand the general scene structure and consequently the behavior of MPI. The second part is trained in an unsupervised way on real data (dataset S_2, Section 7) without using ground truth information. By minimizing the loss of Eq. 7 we aim at fooling the discriminator by modifying the output of G in order to generate depth maps similar to the ground truth ones. This allows to obtain samples of $I^r_{D;G} = (d^r_n; d^r_n - d^r_G)$ (i.e. couples of noisy depth maps and related error images) similar to the ground truth data $I_{D;gt}$. With the proposed training approach, we can train G to adapt to and denoise real world data without capturing depth ground truth for real scenes.

The implementation of the loss functions given by Eq. 2 and Eq. 7 follows the LS-GAN structure proposed in [23], where the negative log likelihoods are replaced by least squared loss in order to stabilize the learning process.

At each step of the training phase, a batch of real data and a batch of synthetic data are sampled from the two training datasets S_1 and S_2. At first, the synthetic data are used to train the discriminator as mentioned in Section 5. By following the idea introduced in [28, 34], we exploited a buffer to collect examples of fake data $I^s_{D;G}$, produced by G when processing synthetic data in past training steps. Two different strategies can be selected with a 50% probability each. In the first, D is trained using data produced by G in the current training step. In the second, data collected in the buffer is extracted at random and used as fake examples for training while the buffer is filled with the data produced by G. This approach allows to avoid that D overfits on the current status of G. Thus, it stabilizes the training process and lets D focus also on fake data related to previous training steps, since these always have to be classified as fake. In this way, D captures the statistics of $I_{D;gt}$ better.

Simultaneously, G is trained on the unlabeled real data by minimizing the loss function of Eq. 7 and on the synthetic data by minimizing the loss in Eq. 6.

Since the exploited synthetic dataset is not too large, we have used *K-fold cross validation* with *K=5* on the synthetic training set to control and avoid over-fitting. Instead, the real dataset used for the adversarial training is completely unlabelled. For this reason, we used an additional real dataset provided with depth ground truth as validation set during the domain adaptation process. We have optimized the hyper-parameters of the CNN and of the training procedure, i.e., the learning rate, the weight of the adversarial loss L_{adv} and the structure of the discriminator network in order to reduce the most the average mean absolute error (MAE) on the real validation set S_3 (see Section 7) after the k-fold cross validation on the synthetic dataset.

The complete learning procedure is summarized in Algorithm 1 and in Fig. 4: we optimized the two neural networks using the TensorFlow framework [1] with the ADAM optimizer. The learning rate has been set to $5 \cdot 10^{-6}$, while the weight of the adversarial part has been set to $w = 5 \cdot 10^{-3}$. Each batch contains 4 samples and we trained the network for 10^5 training steps. Fig. 5 shows the mean behavior of the validation error (MAE) on the real world validation dataset S_3 (this dataset has depth ground truth, see Section 7) of the proposed architecture after k-fold cross validation. The figure compares the presented approach with the training curves obtained without using some of its components in order to allow for some ablation considerations.

The blue curve corresponds to the supervised training on synthetic data of the generator (i.e. without using the adversarial domain adaptation). It can be clearly seen that the validation error is higher than the proposed method, in particular the error initially decreases but after a certain point the accuracy does not improve, since the deep network is basically overfitting on the synthetic data.

The green curve corresponds to the baseline adversarial learning method without the history buffer and the data augmentation. The achieved minimum error is smaller than the supervised training, even if not as good as the complete version of our approach. On the other hand, the training looks unstable and after a certain point, the discriminator dominates on the generator and the validation error increases.

The purple plot corresponds to the use of data augmentation but no history buffer: the minimum error is similar to the previous case, but the curve is more stable and the problem of the discriminator saturation is more limited. The opposite case (history but no data augmentation) has a similar behavior with slightly better performance (in the final part the yellow curve has lower and more stable values).

Finally, by putting together all the components we can obtain very good performance with a small and stable validation error (red curve). In particular, note that even if the gap in terms of minimum error, obtained by adding data augmentation and history is not so large, the two techniques allow to obtain more stable training behavior and to avoid the unbalancing of the generator and discriminator after a certain point. This suggests that the full version of the approach has better generalization properties and can be applied on a wider set of different scenes and settings.

Algorithm 1 Domain Adaption Procedure

```
procedure TRAINING STEP
    $(I^s_G;\ d^s_{gt}) \leftarrow S_1$                          ▷ Get synthetic data
    $I^r_G \leftarrow S_2$                                       ▷ Get real world data
    $d^s_{60} \leftarrow I^s_G$ and $d^r_{60} \leftarrow I^r_G$
    $E^s_{gt} = d^s_{60} - d^s_{gt}$
    $k = rand.unif([1-\epsilon; 1+\epsilon])$                    ▷ For noise augm.
    $I^s_{D;gt} = (d^s_{gt} + k \cdot E^s_{gt}; k \cdot E^s_{gt})$
    $I^s_{D;G} = (d^s_{60}; d^s_{60} - G(I^s_G))$
    $I^r_{D;G} = (d^r_{60}; d^r_{60} - G(I^r_G))$
    if $rand.uniform([0;1]) > 0.5$ then
        $I^{s,\ curr}_{D;G} = I^s_{D;G}$
    else
        $I^{s\ curr}_{D;G} = queue.get\_sample()$
        $queue.push(I^s_{D;G})$
  ▷ Optimize the discriminator $(D)$
    minimize $L_D$ (Eq. 2) on $I^s_{D;gt}$ and $I^{s\ curr}_{D;G}$
  ▷ Optimize the generator $(G)$
    minimize $L_{sup}$ (Eq. 6) on $(I^s_G;\ d^s_{gt})$
    minimize $L_{adv}$ (Eq. 7) on $I^r_{D;G}$
```

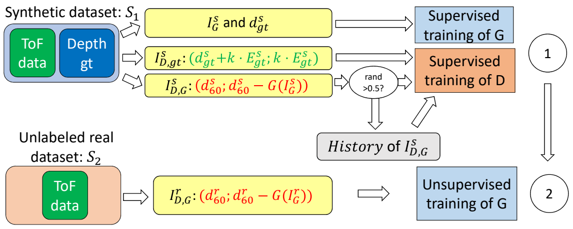
Synthetic dataset: S_1
ToF data
Depth gt
I^s_G and d^s_{gt}
$I^s_{D,gt}$: $(d^s_{gt}+k \cdot E^s_{gt}; k \cdot E^s_{gt})$
$I^s_{D,G}$: $(d^s_{60}; d^s_{60} - G(I^s_G))$
rand >0.5?
Supervised training of G
Supervised training of D
1
History of $I^s_{D,G}$
Unlabeled real dataset: S_2
ToF data
$I^r_{D,G}$: $(d^r_{60}; d^r_{60} - G(I^r_G))$
Unsupervised training of G
2

Figure 4: Schematic representation of a training step.

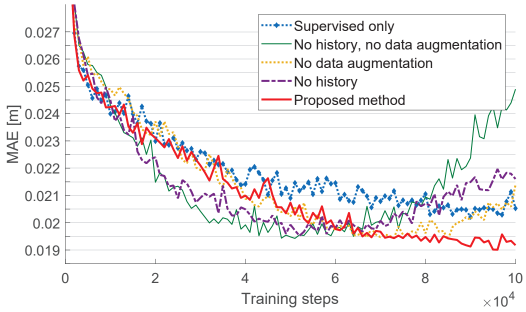

Figure 5: Validation error during the training procedure for different versions of the proposed approach.

7. Synthetic and Real World Datasets

We exploited 5 different datasets for the training and evaluation of this work.

For the supervised training with synthetic data we used the dataset introduced in [5]. This dataset (S_1) is composed of 40 synthetic scenes with multi-frequency data and ground truth depth. We performed data augmentation by extracting 10 random patches of size 128×128 [px] from each scene and by applying rotation and flipping of the patches.

In order to perform the adversarial training procedure, we acquired an unlabeled real world dataset (S_2), using a SoftKinetic Time-of-Flight camera in an office environment. The dataset is composed by 97 scenes (without ground truth depth). The scenes (some are depicted in the *additional material*) have different sizes and subjects and contain critical situations in which MPI is clearly visible.

We also acquired ground truth data for a small set of real world scenes for validation purposes. This smaller dataset, S_3, has only 8 scenes acquired with the SoftKinetic camera, but contains also ground truth information, acquired with an active stereo matching system using high frequency sinusoidal patterns, to reduce diffuse reflection distortions [15], and registered on the ToF sensor.

In order to evaluate the performances of the proposed approach, we used the real world dataset S_4 of [5]. This dataset contains 8 real world scenes with ground truth data and allows to perform the comparison with state-of-the-art approaches for the considered task.

Finally, we acquired another set of scenes (S_5) with ground truth information to ensure a more robust performance evaluation and comparison with competing approaches, focusing on current real world applications. Since ToF sensors can be used in logistics and manufacturing for inspection, handling and dimensioning of parcels, we acquired a set of 8 scenes with ground truth, containing boxes of different sizes. Since the boxes are arranged close or one over the other, there is strong MPI making the precise measurement of these simple geometries challenging.

The *additional material* describes the datasets in more detail and presents some visual examples of the data. Furthermore, the 3 datasets S_2, S_3 and S_5 that have been created for this work are available at `http://lttm.dei.unipd.it/paper_data/MPI_DA_CNN`.

8. Experimental Results

The proposed method has been evaluated using the real datasets S_4 (from [5]) and S_5 . Both the datasets contain 8 different scenes with the corresponding ground truth depth. The scenes contain objects of various sizes and materials and situation in which MPI can arise.

In order to evaluate the performance of the proposed approach we start by analyzing the impact of the proposed adversarial learning strategy and then we compared our method with some state-of-the-art approaches.

8.1. Denoising Properties of the Adversarial Scheme

First of all, we analyze how the adversarial learning strategy allows to perform denoising and MPI removal on real world data. Fig. 6 shows the output of the proposed approach on some sample scenes in the S_4 dataset and compares it with that obtained by the proposed generator network trained in a supervised way on synthetic data. Column 3 shows the error map for input data at 60 MHz. Note the large amount of MPI corruption on slanted surfaces and the issues close to the edges of the objects. Column 4 shows the error map corresponding to the usage of the proposed *Coarse-Fine* network in a supervised fashion (i.e. using only synthetic data for training but, differently from [5], by looking at the performance on the real validation set S_3 and not on the synthetic one to set the hyper-parameters). Note how it is possible to reduce the MPI corruption, but only by a small margin. A strong effect remains on the slanted surfaces, especially on the floor. Furthermore, there is a large amount of error in the proximity of the edges, probably due to the fact that edges are very sharp and well defined on synthetic data, while the mixed-pixel effect produces many artifacts in these regions in real world data. By applying the proposed adversarial learning strategy (last column), it is possible to obtain a noticeable improvement: the amount of MPI on the floor is further reduced, even if not completely removed and the accuracy in proximity of edges is much better than in the supervised case. The visual evaluation is also confirmed by numerical results: on the S_4 dataset, the average MAE on the input data (i.e., the ToF depth map at 60 Mhz) is 5.43 cm. By denoising the data with the network trained in a supervised way, the MAE can be reduced to 2.74 cm, i.e. about half of the original error. By applying the proposed domain adaptation approach, the average error is reduced to 2.36 cm, i.e. a further reduction of about 14% w.r.t. the error of the synthetic supervised approach.

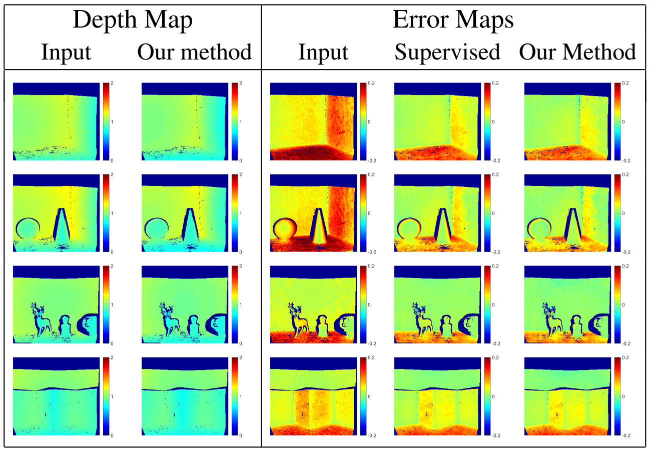

Figure 6: Output of the synthetic supervised and of the proposed domain adaptation approach on some sample scenes from the S_4 dataset. The values are measured in meters.

Method	S_4 Dataset ([5])		S_5 Dataset (*box*)	
	MAE (cm)	Relative error	MAE (cm)	Relative error
Input (60 Mhz)	5.43	-	3.62	-
Input (20 Mhz)	7.28	-	5.06	-
SRA [9]	5.11	94.1%	3.37	93.1%
DeepToF [24]	5.13	70.5%*	6.68	132%*
[24]+calibration	5.46	75%*	3.36	66.4%*
Agresti et al. [5]	3.19	58.7%	2.22	60.5%
Our Approach	**2.36**	**43.5%**	**1.66**	**46.1%**
DA-F (Ours+[12, 27])	2.6	47.9%	1.71	47.2%

Table 1: MAE and relative error on the S_4 and S_5 datasets. The relative error is the ratio between the MAE of each method and the MAE on input at 60 MHz, the highest employed frequency for all approaches, except [24] (*) that is compared with 20 MHz since it uses only this frequency.

8.2. Comparison with State-of-the-Art Approaches

The performance of the proposed method was compared with three state-of-the-art approaches for ToF data denoising, i.e. the multi-frequency scheme of Freedman et al. (SRA) [9], the deep learning based approaches of Marco et al. (DeepToF) [24] and of Agresti et al. [5].

Additionally, we also considered a combination of our model with the domain adaptation scheme of [12, 27] (we denote this idea as *DA-F*). In these approaches, the discriminator is trained to recognize if the features produced internally by the generator (we selected the output of the 4-th convolutional layer in the fine network) are originated from synthetic or real data, thus forcing G to produce similar features in the 2 domains and reducing the domain shift.

From the quantitative evaluation in Table 1, the analytical method of [9] is able to remove only a small part of about 6% of the noise and MPI in the scene. Deep learning based approaches have better performances: DeepToF [24] is able to remove about 30% of the error (w.r.t. the 20 Mhz data used by this approach), while the best competing approach is [5], which removes more than 40% of the corruption. Our approach outperforms all compared approaches with a large margin, removing more than 56% of the error and reducing it to just 2.36 cm. Also the variant with feature-based domain adaptation (DA-F) has good performances (even if lower than the proposed method) and removes about 52% of the error. Note that [5], sharing a similar denoiser CNN but without domain adaptation, obtains lower performance. The evaluation on the *box* dataset leads to very similar results. On this dataset, the initial amount of error is smaller (3.62 cm), mostly due to the simpler geometry of the objects and to the reduced amount of MPI. The SRA method [9] has roughly the same performance obtained on the other dataset, removing only 7% of the error. DeepToF [24] is affected by a systematic bias in the estimations on this dataset. For a fair comparison, we removed the bias by calibrating on a white wall scene, achieving an error reduction of 33%, confirming the results on S_4 also in this

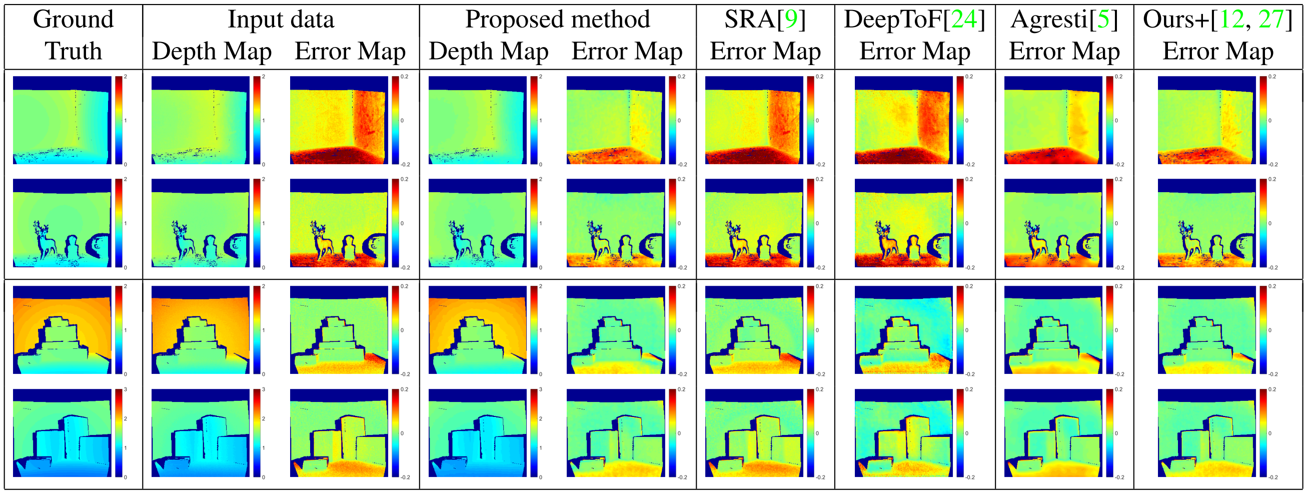

Figure 7: Comparison between the input depth at 60 MHz, the proposed method and the denoised depth maps obtained with some state-of-the-art methods. The figure shows the computed depths with the corresponding error maps for some sample scenes from S_4 (first and second row) and S_5 (third and fourth row). The values are measured in meters.

case. The method of Agresti et al. [5] is better performing and removes about 40% of the corruption. The proposed method reduces the MAE to 1.67 cm, removing 54% of the error, roughly confirming the results obtained on S_4. Again it clearly outperforms the compared approaches. Finally, the DA-F method outperforms [5] and get close to our approach, removing about 53% of the error.

Some visual results are shown in Fig. 7 for both datasets. It is possible to note how the proposed approach is able to remove most of the MPI corruption on the boxes and objects and a large part of the error on the floor (even if some MPI remains in this area). Also its variant DA-F (Ours+[12, 27]) looks visually good. The compared methods are able to remove a smaller amount of MPI. The best of the compared ones is [5], [9] have limited performances and [24] stays midway. Furthermore, edges are more accurately represented than the compared approaches and the zero mean noise is widely reduced. Complex or round shapes like the deer or the sphere are better preserved by the proposed approach while the competing ones introduce relevant artifacts on these objects. The *additional material* contains other visual results.

Fig. 8 shows the correction obtained with the different methods on a cross-section of a corner scene. Please note how the proposed method is able to reconstruct the corner shape more accurately reducing the distortion due to MPI.

9. Conclusions

In this paper, we presented a novel domain adaptation strategy for ToF data denosing. We solved the critical issue of the lack of real world ToF data with depth ground truth by using the combination of supervised learning on labeled synthetic data with an adversarial learning strategy for regression exploiting unlabeled real data. By feeding the discriminator with the combination of noisy data and error maps, it was possible to capture the relation between the structure of the scene and the error, while a novel data augmentation strategy allowed to make the approach more robust to differences between the real and simulated data. The adversarial learning strategy proved to be able to adapt the generator to real world data as demonstrated by the experimental results where the proposed method clearly outperforms all compared approaches.

Finally, note that the proposed approach has been introduced for ToF denoising but in principle it can be applied to any image or data denosing task, where learning on supervised data in one domain needs to be adapted to different domains. Further research will be devoted to the exploration of its applicability for general image denosing tasks.

Acknowledgment We thank Sony EuTEC for allowing us to use their *ToF Explorer* simulator, in particular Oliver Erdler and Markus Kamm. We thank prof. Calvagno for his support and gratefully acknowledge NVIDIA for the donation of the GPUs used for this research.

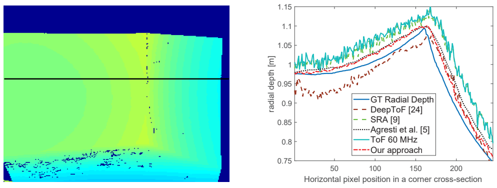

Figure 8: Comparison of different MPI correction methods on a cross-section of a corner scene.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. 5

[2] Supreeth Achar, Joseph R Bartels, William L Whittaker, Kiriakos N Kutulakos, and Srinivasa G Narasimhan. Epipolar time-of-flight imaging. *ACM Transactions on Graphics (ToG)*, 36(4):37, 2017. 2

[3] Gianluca Agresti, Ludovico Minto, Giulio Marin, and Pietro Zanuttigh. Deep learning for confidence information in stereo and tof data fusion. In *Geometry Meets Deep Learning ICCV Workshop*, pages 697–705, 2017. 2

[4] G. Agresti and P. Zanuttigh. Combination of spatially-modulated tof and structured light for mpi-free depth estimation. In *Proceedings of 3D Reconstruction in the Wild ECCV Workshop*, 2018. 2

[5] G. Agresti and P. Zanuttigh. Deep learning for multi-path error removal in tof sensors. In *Geometry Meets Deep Learning ECCV Workshop*, 2018. 1, 2, 3, 6, 7, 8

[6] Ayush Bhandari, Micha Feigin, Shahram Izadi, Christoph Rhemann, Mirko Schmidt, and Ramesh Raskar. Resolving multipath interference in kinect: An inverse problem approach. In *Sensors*, pages 614–617. IEEE, 2014. 2

[7] Ayush Bhandari, Achuta Kadambi, Refael Whyte, Christopher Barsi, Micha Feigin, Adrian Dorrington, and Ramesh Raskar. Resolving multipath interference in time-of-flight imaging via modulation frequency diversity and sparse regularization. *Optics letters*, 39(6):1705–1708, 2014. 3

[8] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 7, 2017. 2

[9] Daniel Freedman, Yoni Smolin, Eyal Krupka, Ido Leichter, and Mirko Schmidt. Sra: Fast removal of general multipath for tof sensors. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 234–249. Springer, 2014. 2, 3, 7, 8

[10] Stefan Fuchs. Multipath interference compensation in time-of-flight camera images. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3583–3586. IEEE, 2010. 2

[11] Stefan Fuchs, Michael Suppa, and Olaf Hellwich. Compensation for multipath in tof camera measurements supported by photometric calibration and environment integration. In *International Conference on Computer Vision Systems*, pages 31–41. Springer, 2013. 2

[12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014. 2, 7, 8

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[14] Qi Guo, Iuri Frosio, Orazio Gallo, Todd Zickler, and Jan Kautz. Tackling 3d tof artifacts through learning and the flat dataset. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 3

[15] Mohit Gupta and Shree K Nayar. Micro phase shifting. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 813–820. IEEE, 2012. 6

[16] Mohit Gupta, Shree K Nayar, Matthias B Hullin, and Jaime Martin. Phasor imaging: A generalization of correlation-based time-of-flight imaging. *ACM Trans. Gr.*, 34(5):156, 2015. 3

[17] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 2

[18] David Jiménez, Daniel Pizarro, Manuel Mazo, and Sira Palazuelos. Modeling and correction of multipath interference in time of flight cameras. *Image and Vision Computing*, 32(1):1–13, 2014. 2

[19] Jiyoung Jung, Joon-Young Lee, Yekeun Jeong, and In So Kweon. Time-of-flight sensor calibration for a color and depth camera pair. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1501–1513, 2015. 2

[20] Damien Lefloch, Rahul Nair, Frank Lenzen, Henrik Schäfer, Lee Streeter, Michael J Cree, Reinhard Koch, and Andreas Kolb. Technical foundation and calibration methods for time-of-flight cameras. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 3–24. Springer, 2013. 2

[21] Frank Lenzen, Henrik Schäfer, and Christoph Garbe. Denoising time-of-flight data with adaptive total variation. In *International Symposium on Visual Computing*, pages 337–346. Springer, 2011. 2

[22] Marvin Lindner, Ingo Schiller, Andreas Kolb, and Reinhard Koch. Time-of-flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding*, 114(12):1318–1328, 2010. 2

[23] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 2813–2821. IEEE, 2017. 5

[24] Julio Marco, Quercus Hernandez, Adolfo Muñoz, Yue Dong, Adrian Jarabo, Min H Kim, Xin Tong, and Diego Gutierrez. Deeptof: off-the-shelf real-time correction of multipath interference in time-of-flight imaging. *ACM Trans. Gr.*, 36(6):219, 2017. 2, 3, 7, 8

[25] Nikhil Naik, Achuta Kadambi, Christoph Rhemann, Shahram Izadi, Ramesh Raskar, and Sing Bing Kang. A light transport model for mitigating multipath interference in time-of-flight sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–81, 2015. 2

[26] Can Pu, Runzi Song, Nanbo Li, and Robert B. Fisher. Sdf-gan: Semi-supervised depth fusion with multi-scale adversarial networks. *CoRR*, abs/1803.06657, 2018. 2

[27] Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 7, 8

[28] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 5, 2017. 2, 5

[29] Kilho Son, Ming-Yu Liu, and Yuichi Taguchi. Learning to remove multipath distortions in time-of-flight range images for a robotic arm setup. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3390–3397, 2016. 2

[30] Shuochen Su, Felix Heide, Gordon Wetzstein, and Wolfgang Heidrich. Deep end-to-end time-of-flight imaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6383–6392, 2018. 1, 2, 3

[31] Refael Whyte, Lee Streeter, Michael J Cree, and Adrian A Dorrington. Review of methods for resolving multi-path interference in time-of-flight range cameras. In *IEEE Sensors*, pages 629–632. IEEE, 2014. 2

[32] Refael Whyte, Lee Streeter, Michael J Cree, and Adrian A Dorrington. Resolving multiple propagation paths in time of flight range cameras using direct and global separation methods. *Optical Engineering*, 54(11):113109, 2015. 2

[33] Pietro Zanuttigh, Giulio Marin, Carlo Dal Mutto, Fabio Dominio, Ludovico Minto, and Guido Maria Cortelazzo. *Time-of-Flight and Structured Light Depth Cameras*. Springer, 2016. 1, 2

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017. 5