DeepToF: Off-the-Shelf Real-Time Correction of Multipath Interference in Time-of-Flight Imaging

JULIO MARCO*, QUERCUS HERNANDEZ, and ADOLFO MUÑOZ, Universidad de Zaragoza, I3A
YUE DONG, Microsoft Research Asia
ADRIAN JARABO, Universidad de Zaragoza, I3A
MIN H. KIM, KAIST
XIN TONG, Microsoft Research Asia
DIEGO GUTIERREZ, Universidad de Zaragoza, I3A

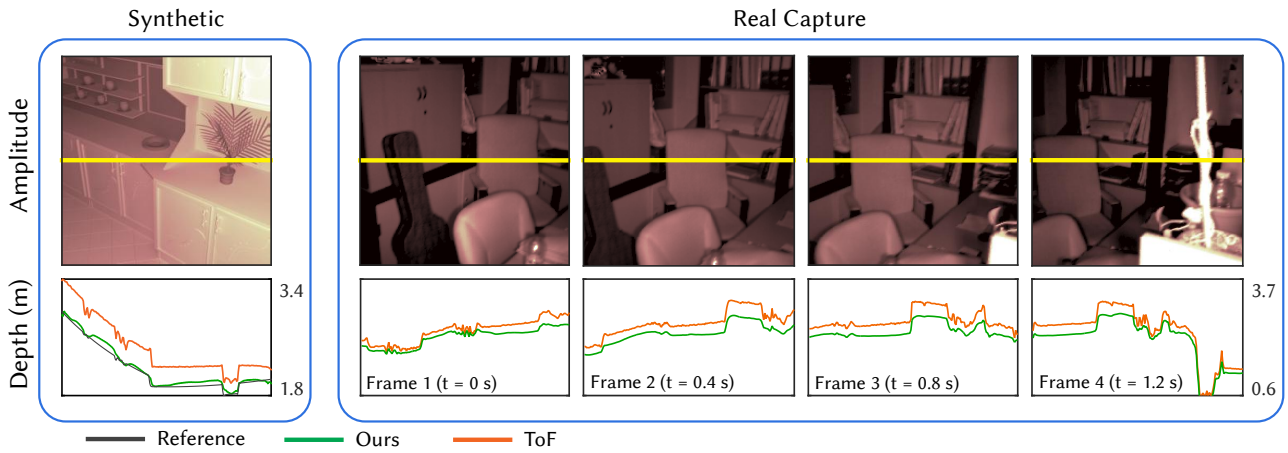

Fig. 1. We propose a convolutional approach to correct multipath interference (MPI) errors, in real-time and on complex scenes captured with off-the-shelf, time-of-flight (ToF) cameras. We introduce a novel two-stage learning scheme: we first pre-initialize our network to learn real depth representations, and then re-train these to learn MPI corrections from synthetic data. The leftmost image shows one of our synthetic validation scenes: our method (green) provides a better fit to ground truth data than ToF depth from the camera. In the sequence to the right, we illustrate how our system allows for real-time MPI compensation (10 milliseconds per frame) in real scenes captured with ToF devices, preserving temporal consistency. Please refer to the supplemental video for more results.

Time-of-flight (ToF) imaging has become a widespread technique for depth estimation, allowing affordable off-the-shelf cameras to provide depth maps in real time. However, multipath interference (MPI) resulting from indirect illumination significantly degrades the captured depth. Most previous works have tried to solve this problem by means of complex hardware modifications or costly computations. In this work, we avoid these approaches and propose a new technique to correct errors in depth caused by MPI, which requires no camera modifications and takes just 10 milliseconds per frame. Our observations about the nature of MPI suggest that most of its information is available in image space; this allows us to formulate the depth imaging process as a spatially-varying convolution and use a convolutional neural network to correct MPI errors. Since the input and output data present similar structure, we base our network on an autoencoder, which we train in two stages. First, we use the encoder (convolution filters) to learn a suitable basis to represent MPI-corrupted depth images; then, we train the decoder (deconvolution filters) to correct depth from synthetic scenes, generated by using a physically-based, time-resolved renderer. This approach allows us to tackle a key problem in ToF, the lack of ground-truth data, by using a large-scale captured training set with MPI-corrupted depth to train the encoder, and a smaller synthetic training set with ground truth depth to train the decoder stage of the network. We demonstrate and validate our method on both synthetic and real complex scenarios, using an off-the-shelf ToF camera, and with only the captured, incorrect depth as input.

CCS Concepts: • **Computing methodologies → Computer vision**; **Computational photography**; **3D imaging**;

Additional Key Words and Phrases: Time-of-Flight, depth cameras, multipath interference, learning

ACM Reference format:
Julio Marco, Quercus Hernandez, Adolfo Muñoz, Yue Dong, Adrian Jarabo, Min H. Kim, Xin Tong, and Diego Gutierrez. 2017. DeepToF: Off-the-Shelf Real-Time Correction of Multipath Interference in Time-of-Flight Imaging. *ACM Trans. Graph.* 36, 6, Article 219 (November 2017), 12 pages.
https://doi.org/10.1145/3130800.3130884

*Part of this work was done during Julio Marco's internship at Microsoft Research Asia.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2017 Association for Computing Machinery.
0730-0301/2017/11-ART219 $15.00
https://doi.org/10.1145/3130800.3130884

1 INTRODUCTION

Time-of-flight (ToF) imaging, and in particular continuous-wave ToF cameras, have become a standard technique for capturing depth maps. Such devices compute depth of visible geometry by emitting modulated infrared light towards the scene, and correlating different phase-shifted measurements at the sensor. However, ToF devices suffer from multipath intereference (MPI): a single pixel records multiple light reflections, but it is assumed that all light reaching it has followed a direct path (see [Jarabo et al. 2017] for details). This introduces an error on the captured depth, which reduces the applicability of ToF cameras.

To compensate for MPI, most previous works leverage additional sources of information, such as coded illumination or multiple modulation frequencies that lead to different phase-shifts, from which indirect light might be disambiguated. This requires either hardware changes (e.g. modifying the built-in illumination, or using sensors that can handle multiple modulated frequencies), or multiple passes with a standard ToF camera. Other single-modulation approaches simulate an estimation of the ground-truth light transport, then compensate the captured MPI using information from such simulation. Such approaches, while working with any out-of-the-box ToF sensor, typically require several minutes for a single frame, and might lead to errors when the simulation is not accurate.

Our work aims to lift these limitations: We present a novel technique to correct MPI using an *unmodified, off-the-shelf camera*, with a *single frequency*, and in *real time*. A key observation is that, since both the camera and the infrared emitter are co-located and share almost the same visibility frustrum, most of the MPI information is actually present in image space. Furthermore, from the discretized geometry given by a depth map, light transport at each pixel can only be estimated as a linear combination (with unknown weights) of the contributions from the rest of the pixels. This linear process can be represented as a spatially-varying convolution in image space, with unknown convolution filters. This motivates our design of a convolutional neural network (CNN) to obtain such filters.

However, suitable ToF datasets that include depth with MPI and its corresponding ground-truth reference do not exist, and capturing such dataset is not possible with current devices. To overcome this, we synthesize such data using an existing physically-based transient light transport renderer (Section 5), extending it with a ToF camera model. Using this model, we introduce MPI in the simulated depth estimation, and compare it with the reference depth. Our network uses the synthetic data for training, takes depth with MPI as input, and returns a corrected depth map. In particular, since the input and output have the same resolution, we design an encoder-decoder network. While amplitude or phase-shifted images could provide additional information to the network, they are often uncalibrated and highly dependent on the device characteristics. We therefore use depth as the only reliable input to our network, providing a real-time solution that is robust even for single-frequency ToF devices.

This approach introduces two new challenges. First, generating synthetic data is time-consuming, and thus the simulated dataset is unlikely to be large and diverse enough to avoid overfitting. Second, although we carefully analyze our synthetic data to make sure that it is statistically similar to real-world data, it might still be too perfect, lacking for instance subtle differences due to imperfections in the sensor or the emitter. We address this by leveraging the fact that the input and output depths must be structurally similar, and devising a two-stage training. The first stage is a convolutional autoencoder (CAE) from real-world captured data, which requires no ground-truth reference. This tackles both challenges, since it allows us to use large datasets with real-world imperfect data for training. This first stage thus trains the encoding filters of the network as a feature dictionary learned from structural properties of ToF depth images (Section 6.1).

The second stage provides supervised learning for the regression with the synthetic dataset as reference, which accounts for the effect of MPI (input and output are now different), and feeds the decoder (while the encoder remains unmodified). We treat the effect of MPI as a residue, and therefore model this second stage as supervised residual learning (Section 6.2).

We analyze the performance of our approach in synthetic and captured ground-truth data, and compare against previous works, showing favorable results while being significantly faster. Finally, we demonstrate our technique in-the-wild, correcting MPI from depth maps captured by a ToF camera in real time. In summary, we make the following contributions:

- A two-stage training strategy, with a convolutional autoencoder plus a residual learning approach. It leverages statistical knowledge from real captured data with no ground truth, and then compensates the error from synthetic data (which includes ground truth).
- A synthetic ToF dataset of scenes sharing similar statistical properties as real-world scenes. For each scene, we provide both MPI-corrupted and ground-truth depth. We believe this data is much needed, and we hope our dataset can help future works.
- A trained network that compensates multipath interference from a single ToF depth image in real time, which outperforms previous algorithms even using such minimal input.

Our training dataset and trained network are publicly available online at http://webdiis.unizar.es/~juliom/pubs/2017SIGA-DeepToF/

2 RELATED WORK

Convolutional neural networks (CNNs) have been widely used for many image-based reconstruction tasks, such as intrinsic images, normal estimation, or depth recovery. Here we only focus on CNN-based depth reconstruction methods that are closely related to our work. We refer to Jarabo et al.'s recent survey [Jarabo et al. 2017] for a complete overview on transient and ToF imaging, and to Goodfellow et. al.'s book [2016] for other deep learning techniques and their applications.

CNN-based depth reconstruction. A set of methods derive depth from multi-view images. Zbontar et al. [2015] trained a CNN for computing the matching cost of stereo image pairs. Kalantari et. al. [2016] exploited a CNN to estimate the disparity between sparse light-field views, and fed the result to another CNN to interpolate the light-field views for novel view synthesis. Different from these

multi-view methods, we reconstruct a depth image from a single snapshot captured by a monocular ToF camera.

Other methods estimate depth from a single RGB image. Eigen et. al. [2015; 2014] proposed an end-to-end CNN in which a coarse depth image is first recovered, then progressively refined. In each step, the coarse depth is upsampled and combined with fine scale image features. Based on this approach, several works formulate the generated depth as a conditional random field (CRF), and then refine it with the help of color image segmentation [Li et al. 2015; Liu et al. 2015; Wang et al. 2015], or multi-resolution depth information generated by intermediate CNN layers [Xu et al. 2017]. Although these methods improve the accuracy of the result, the CRF optimization is expensive and slow. Most recently, Su et. al. [2017] trained a CNN with a synthetic dataset rendered from a large dataset for reconstructing a low-resolution 3D shape from a single RGB or RGB-D image. Different from these methods, we do not rely on additional sources of information. Moreover, we have also developed an efficient scheme that combines both unlabeled real ToF images and labeled synthetic data for CNN training. This can efficiently generate a full-resolution depth map at a rate of up to 100 frames/second.

Multipath interference. Several works take advantage of inputs with several amplitudes and phase images through multiple modulation frequencies. This input can be translated into multipath interference correction through optimization [Dorrington et al. 2011; Freedman et al. 2014], closed-form solutions with inverse attenuation polynomials [Godbaz et al. 2012], spectral methods [Feigin et al. 2016; Kirmani et al. 2013], sparse regularization [Bhandari et al. 2014], or through modeling indirect lighting as phasor interactions in frequency space [Gupta et al. 2015]. Although these techniques are efficient for some devices that can capture a few frequencies simultaneously, such as Kinect V2 [Feigin et al. 2016], they require multiple passes for single-frequency ToF cameras.

Other approaches deal with multipath interference by adding or modifying hardware. Wu and colleagues [2014] decompose global light transport into direct, subsurface scattering, and interreflection components, leveraging the extremely high temporal resolution of the femto-photography technique [Velten et al. 2013]. Modified ToF sensors allow to reconstruct a transient image from multiple frequencies [Heide et al. 2013; Peters et al. 2015]. Other techniques include custom coding [Kadambi et al. 2013], or combining ToF sensors with structured light projection [Naik et al. 2015; O'Toole et al. 2014]. In contrast, our method works with just an out-of-the-box phase ToF sensor, without any modifications.

Some techniques remove multipath interference from a single frequency (a single amplitude and depth image) by estimating light transport from the approximated depth. One of the approximations considers a single indirect diffuse bounce (assuming constant albedo) connecting all pairs of pixels on the scene [Fuchs 2010]. This has been later extended to multiple diffuse bounces and multiple albedos, by adding some user input [Fuchs et al. 2013]. Last, an optimization algorithm over depth space with path tracing has also been presented [Jiménez et al. 2014]. While these approaches manage to compensate multipath interference from input obtained with any ToF device, they are very time-consuming; moreover, given the sparse input and the assumptions simplifying the underlying light transport model, they are unable to completely disambiguate indirect light in all cases. In contrast, given even less information (a single depth map, without amplitude) our work is able to compensate multipath interference for varied and complex geometries, with different albedos, and in real time.

Table 1 summarizes these approaches, and compares them to our method. We list the required input, the variability of the tested scenes (including albedo and geometry) and the execution time. Our method works on a large range of scenes, with a just a single depth map as input, and yields real time performance.

3 PROBLEM STATEMENT

ToF depth errors. Using four phase-shifted measurements $\mathbf{c}_{1\ldots4}$, ToF devices compute the depth z at every pixel p as

$$z = \frac{c\,\phi}{4\pi f_{\omega_R}} \tag{1}$$

$$\phi = \arctan\left(\frac{\mathbf{c}_4 - \mathbf{c}_2}{\mathbf{c}_1 - \mathbf{c}_3}\right), \tag{2}$$

where f_{ω_R} is the device modulation frequency, c is the speed of light in a vacuum, and ϕ is the phase of the wave reaching a pixel p. This model works under the assumption of a single impulse response from the scene, therefore assuming

$$\mathbf{c}_i(p) = x(p)e^{2\pi\, j\, \phi(p)\frac{f_{\omega_R}}{c} + \theta_i}, \tag{3}$$

with $j = \sqrt{-1}$, $x(p)$ the amplitude of the wave, and θ_i the phase shift of measurement $\mathbf{c}_i$. However, given indirect illumination, the observed pixel may receive light from paths other than single-bounce direct light, which leads to

$$\hat{\mathbf{c}}_i(p) = \mathbf{c}_i(p) + \int_P x(\overline{p})\, e^{2\pi\, j\, \phi(\overline{p})\frac{f_{\omega_R}}{c} + \theta_i}\, d\overline{p}, \tag{4}$$

where P is the space of all the light paths $\overline{p}$ reaching pixel p from more than one bounce; the amplitude $x(\overline{p})$ and phase delay due to light time-of-flight $\phi(\overline{p})$ are now functions of the path $\overline{p}$. The effect of multiple bounce paths if often ignored in ToF sensors obtaining approximate measures $\mathbf{c}_i(p) = \hat{\mathbf{c}}_i(p)$ and therefore leading to a depth estimation error, the multipath interference (MPI).

MPI observations. In ToF range devices, the light source is typically co-located with the camera, sharing a similar visibility frustum. As we illustrate in Figure 2 (camera facing the table), this implies that most of the MPI due to second-bounce indirect illumination comes from actual visible geometry. Previous works have leveraged this by taking into account only second-bounce illumination [Dorrington et al. 2011; Fuchs 2010], or by ignoring non-visible geometry [Jiménez et al. 2014]. Higher-order indirect illumination might come from non-visible geometry, but due to the exponential decay of scattering events and quadratic attenuation with distance, light paths of more than two bounces interfere mainly in the local neighborhood of a point (see Figure 2, camera facing the plants). These observations suggest that most of the information on multipath interference from a scene is available in *image space*, where Equation (4) is discretized into a summatory and can be modeled as a spatially-varying convolution. Please refer to Appendix A for a more detailed derivation of such spatially-varying convolution

Table 1. Comparison between the different existing techniques that address the problem of multipath interference, including required input, tests shown in the paper (material types and variability of scenes) and execution time. Our approach performs in real time while requiring the most reduced input. Furthermore, we show a greater scene variability in terms of materials and geometries.

Work	Input	Tested materials	Scenes	Reported time
[Fuchs 2010]	Single frequency	Single albedo	Few planes	≈ 10 minutes
[Dorrington et al. 2011]	Multiple frequencies	Multiple albedos	Few planes	Unreported
[Godbaz et al. 2012]	Multiple frequencies	Multiple albedos	Few planes	Unreported
[Fuchs et al. 2013]	Single frequency, user input	Multiple albedos	Few planes	≈ 150 seconds
[Heide et al. 2013]	Multiple freqs. hardware mods.	Multiple albedos	Several	≈ 90 seconds
[Kadambi et al. 2013]	Single freq., custom code	Multiple albedos	Several	≈ 4 seconds
[Kirmani et al. 2013]	Multiple frequencies	Single albedo	Few planes	Real time
[Freedman et al. 2014]	Multiple frequencies	Mult. albedos, specular	Few scenes	31.2 ms
[Bhandari et al. 2014]	Multiple frequencies	Transparency	Few planes	Unreported
[O'Toole et al. 2014]	Multiple freqs., structured light	Multiple albedos	Several	Unreported
[Jiménez et al. 2014]	Single frequency	Multiple albedos	Many	Several minutes
[Peters et al. 2015]	Multiple frequencies	Mult. albedos, specular	Scene, video	18.6 fps
[Gupta et al. 2015]	Multiple frequencies	Multiple albedos	Few scenes	Unreported
[Naik et al. 2015]	Single freq. structured light	Multiple albedos	Few planes	"Not real time"
[Feigin et al. 2016]	Multiple frequencies	Transparency	Several	"Efficient"
Our work	Single depth	Multiple albedos	Many	10ms

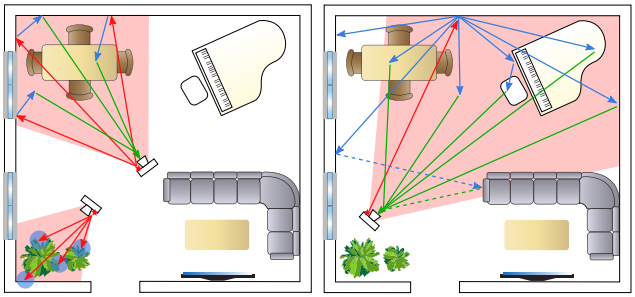

Fig. 2. ToF MPI at different scales, showing IR emission (red), indirect bounces (blue) and observed radiance (green). Left: Observed second-bounce illumination occurs mostly from reflections on visible geometry due to shared light-camera visibility frustum (camera facing the table), while higher-order bounces usually have a significant impact in the locality of the observed points (e.g. camera facing the plants). Right: Large objects such as a wall may cast a significant indirect component over the whole scene when captured from afar, while longer paths from out-of-sight geometry (discontinuous) create negligible MPI due to attenuation.

model. Last, since the effect of multipath interference does not eliminate major structural features of a depth map, the incorrect depth and the reference depth are structurally similar.

4 OUR APPROACH

Given that MPI can be expressed as a spatially-varying convolution, MPI compensation could be modeled as a set of convolutions and deconvolutions in depth space. This in turn suggests that MPI errors could in principle be solved designing a convolutional neural network (CNN). Specifically, since incorrect and correct depths are only slightly different (but structurally similar), using a convolutional autoencoder (CAE) would be a tempting solution. A convolutional autoencoder is a powerful tool which takes the same input and output to learn hidden representations of lower-dimensional feature vectors by *unsupervised* learning, resulting in two symmetric networks: an encoder and a decoder. This allows to build a deeper network architecture, and preserves spatial locality when building these representations [Masci et al. 2011]. The lower-dimensional feature vectors retain the relevant structural information on the input and eliminate existing errors, effectively returning the restored (reference) image. Recently, CAEs have been successfully used in many vision and imaging tasks (e.g. [Choi et al. 2017; Du et al. 2016]).

A straight-forward CAE, nevertheless, cannot be applied to our particular problem: as the errors introduced by MPI are highly correlated with the reference depth to be recovered, we need such ground-truth reference for training. However, a large enough labeled dataset (i.e., pairs of MPI-corrupted depth and its corresponding ground-truth depth), needed for training, does not exist. Although real-world, ToF depth images are widely available, measuring their ground-truth depth maps is a non-trivial task. On the other hand, rendering time-resolved images from synthetic scenes is extremely time-consuming, and the results would only cover a small portion of real-world scene variations.

We propose a two-step training scheme to infer our convolutional neural network from both *unlabeled* real depth images, and labeled synthetic depth image pairs (with and without MPI). Our method is inspired by super-resolution methods based on overcomplete dictionaries and sparse coding [Yang et al. 2010]. Figure 6 shows an overview of our network: We first learn an encoder network as a depth prior through traditional unsupervised CAE training, in which unlabeled real depth images (with unknown errors) are used as both input and output. The resulting encoder allows us to obtain lower-dimensional feature vectors of incorrect depth images. In our second step, different from sparse coding where original signals are reconstructed as a linear product of dictionary atoms, we train a decoder that can reconstruct the reference depth map from such feature vectors. To this end, we keep the encoder network unchanged and cascade it with a residual decoder network. The weights of the decoder network are learned from the synthetic depth pairs via supervised CNN training.

Our network therefore only takes the depth image with MPI as input, and outputs a depth map without the effects of MPI. We do not feed our network with any other ToF information, such as pixel amplitudes or phase images, because these properties highly depend

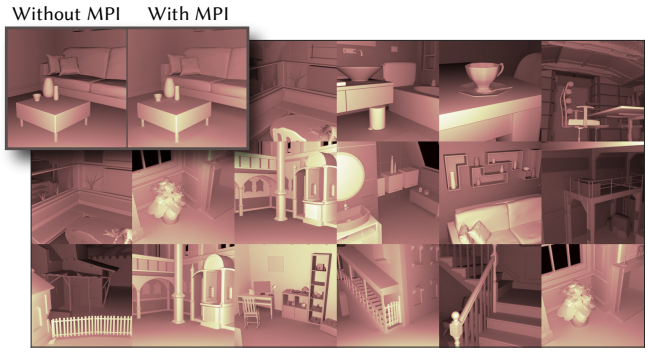

Fig. 3. Representative sample of the scenes (amplitude) rendered with the ToF model to generate depth images with MPI. The top-left image shows the same scene with and without MPI.

on specific ToF camera settings, and are unstable. By using only depth as input, our solution is robust using off-the-shelf ToF devices that operate with a single frequency.

In the following two sections, we first introduce our dataset for training (Section 5), then describe in detail our network and our two-step training scheme (Section 6).

5 TRAINING DATA

To obtain accurate pairs of incorrect (due to MPI) and ground-truth reference depth maps, we simulate the response of the ToF lighting model using the publicly-available, time-resolved bidirectional path tracer of Jarabo et al. [2014]. This allows us to obtain four phase images [Equation (2)], and from those the MPI-distorted depth estimation of a ToF camera [Equation (1)].

Existing ToF devices generally use square modulation functions instead of perfect sinusoidal ones; this introduces additional errors on the depth estimation (*wiggling*), although ToF cameras do account for these errors and compensate them in the final depth image. The wiggling effect and its correction are specific for each camera, and in general information is not provided by manufacturers. We therefore use ideal sinusoidal functions to avoid introducing non-MPI-related error sources. Ground-truth depths are straightforward to obtain from simulation.

Dataset. We simulated 25 different scenes with varying materials, using six different albedo combinations between 0.3 and 0.8, and rendered from seven different viewpoints, at 256×256 resolution—similar to what ToF cameras yield—, computed with up to 20 bounces of indirect lighting. From these we obtained 1050 depth images with MPI (which we flip and rotate to generate a total of 8400) and their respective reference depths. The geometric models of the scenes were obtained from three different free repositories[1]. Some examples can be seen in Figure 3. In the remaining of this paper we refer to this as the **synthetic dataset**.

Training a network from scratch requires a sufficiently large labeled dataset. However, generating it is very time-consuming, so

[1]https://benedikt-bitterli.me/resources/
http://www.blendswap.com/
https://free3d.com/

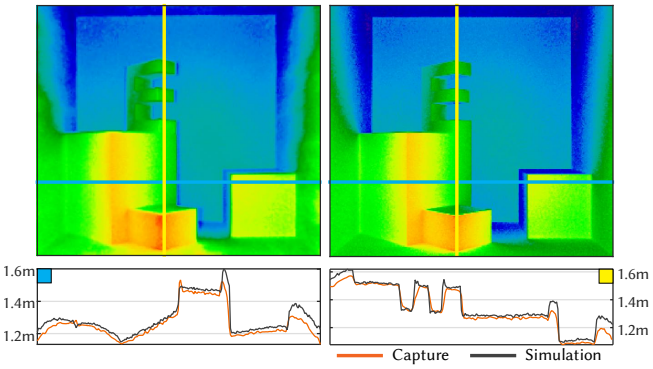

Fig. 4. Depth map with MPI of a real scene (top left) captured by a ToF camera, and its corresponding synthetic model (top right). The bottom row shows the depth profiles for the horizontal (left) and vertical (right) lines, showing a good agreement.

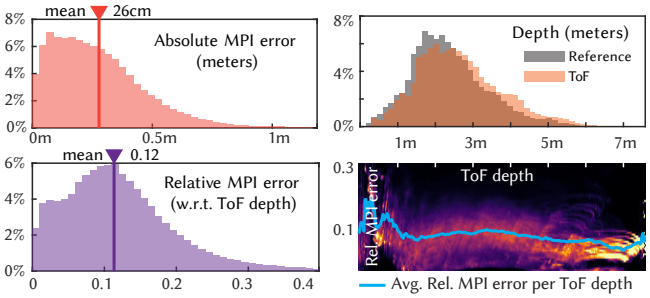

Fig. 5. In reading order: Absolute MPI error; depth distribution for reference and ToF depths, at a modulation of 20MHz (i.e. maximum unambiguous distance of 7.5m); relative error with respect to measured ToF depth; and bivariate histogram showing relative MPI error density per measured ToF depth. Measured ToF depths contain an average error of 12%. The blue line in the bivariate histogram shows that the average relative MPI error per ToF depth remains around 10% for most measured depths.

using only synthetic data for our purposes is highly unrealistic. We therefore gather an additional dataset of 6000 *unlabeled*, real depth images (48000 with flips and rotations) from public repositories [Karayev et al. 2011; Silberman et al. 2012; Xiao et al. 2013], and use them to pre-initialize our network; we refer to this as the **real dataset**. Learning representations of unlabeled real depths will later improve depth corrections from our smaller synthetic labeled dataset (Section 6).

Figure 4 compares a real scene captured with a ToF camera (thus including MPI) with our ToF simulation, showing a good match. Additionally, in Appendix B we perform a statistical analysis on both datasets, to assess their similarity.

Quantitative analysis of MPI errors. Figure 5 shows the error distributions across our entire synthetic dataset. Note that while previous works have addressed local errors of just a few centimeters in small scenes, our data indicates that the global component can introduce much larger errors, with an average error of 26 cm (red line in the top-left histogram) for scenes up to 7.5 m. On average, 12% of the

measured ToF depth corresponds to MPI (bottom-left histogram). The bivariate histogram relating relative error and observed depth (bottom-right) additionally shows that the average remains constant at around 10% for most measured depths, being larger for smaller depths.

6 NETWORK ARCHITECTURE

We now describe how to train our network, following our two-stage scheme with the real dataset (during autoencoding) and our synthetic dataset (during supervised decoding).

6.1 Stage One: Autoencoder

We train our convolutional autoencoder using the real dataset, containing 48000 depth images with unknown errors. We use this incorrect depth as input and output for this unsupervised training, and use the synthetic dataset (with MPI) as its validation set. With this stage we pre-initialize the network so the encoder (Figure 6, top, gray blocks) is able to generate lower-dimensional feature vectors (Figure 6, top, blue block) for both real and synthetic depth maps. Training and validation curves of this stage are shown in Figure 7. Once we train the parameters of the encoder, we freeze its convolutional layers and update only the decoder layers in the second stage.

Network Parameters. In the encoding stage, we apply sets of two 5×5 convolutions with a two-pixel padding, and a stride of two pixels to progressively reduce the size of the convolutional inputs to each layer. This helps to effectively combine and find features at different scales. We perform this operation at six scales, applying pairs of convolutions over features of 256×256 pixels (input), down to 8×8 (innermost convolution pair, last encoding layer). In the decoding stage, we perform upsampling and 5×5 convolutions with two-pixel padding, starting from the encoder output (see Figure 6 top, red arrows) until we reach the output resolution 256×256.

We have tested our network without this pre-initialization step, feeding it directly with synthetic labeled data. The results show that the network loses the ability to generalize, arbitrarily decreasing the accuracy in the validation dataset, as presented in Section 7.

6.2 Stage Two: Supervised Decoder

In the second stage, we freeze the encoder layers, and train the decoder through supervised learning using our synthetic dataset. We introduce incorrect depth (with MPI) as input, and target reference depth (without MPI) as output. We use 80% of our synthetic dataset for training, and the remaining 20% for validation. Given that the encoder performs downsampling operations to detect features at multiple scale levels, full resolution outputs (256×256) are significantly blurred. We thus add symmetric skip connections to mix detailed features of the encoding convolutions (which stay unchanged) to their symmetric outputs in the decoder (Figure 6, bottom). Since we observed that the difference between depth with MPI (input) and reference depth (output) is on average 12%, we treat MPI as a residue [He et al. 2016] by performing element-wise additions between the upsampled features and the skipped ones. Training and validation curves of this stage are shown in Figure 7.

In principle, concatenation of skipped features (instead of simpler element-wise additions) could create more complex combinations with upsampled features using additional learned filters. However, in our results we observed that our residual approach performs equally well (even slightly better, see Section 7.1) while yielding a 30% smaller network model, reducing also execution time.

6.3 Implementation Details

We have implemented our network in Caffe, and trained it on an NVIDIA GTX 1080. Our network takes the input depth without applying any normalization. Following previous works using CNNs, all convolutional layers are followed by a batch normalization layer, a scale and bias layers, and a ReLU activation layer, in that order. For training, we use the Adam solver [Kingma and Ba 2014] for gradient propagation. The learning rate was set to $1 \cdot 10^{-4}$, and adjusted in a stepped fashion in steps between $1 \cdot 10^{-3}$ and $1 \cdot 10^{-5}$, to avoid getting stuck in a plateau, while our batch size is set to 16 to maximize memory usage. Our resulting network performs MPI corrections for a single frame in 10 milliseconds. Additional details on the definition of both the network and training, including the input sources, can be found in the project webpage.

7 RESULTS AND VALIDATION

In this section, we first analyze other alternative, simpler networks, showing how they yield inferior results. We then compare our results against existing methods using off-the-shelf cameras, and thoroughly validate the performance of our approach in both synthetic and real scenes, including video in real time. Real scenes were captured with a PMD CamCube 3.0, which provides depth images at 200×200 resolution, and operates at 20MHz with 7.5 meters of maximum unambiguous depth.

7.1 Alternative Networks

We test three alternatives to our CNN: (1) suppressing the pre-initialization autoencoder stage by directly training an encoder-decoder with synthetic labeled data; (2) removing the residual skip connections; (3) substituting residual connections by concatenated connections. Figure 8 shows how our autoencoder with the residual learning approach yields better results with respect to these other alternatives, with better generalization and smaller network size. By computing R^2 scores between the depth predicted by each network and the target depths across the whole set of images, we observe that, without the autoencoding stage, the images present a lower average score than our results (see Figure 8, top-right table). Also, the variance of the per-image mean absolute error without pre-initialization triplicates the variance of our residual network errors, leading to more unstable accuracy. Concatenating skip connections worsens results slightly, while additionally making the network about 30% larger, due to the need to learn more parameters to combine the additional features. Last, removing residual skip connections avoids enriching upsampled features with high resolution features from the encoder layers, producing blurry outputs and therefore a much higher error.

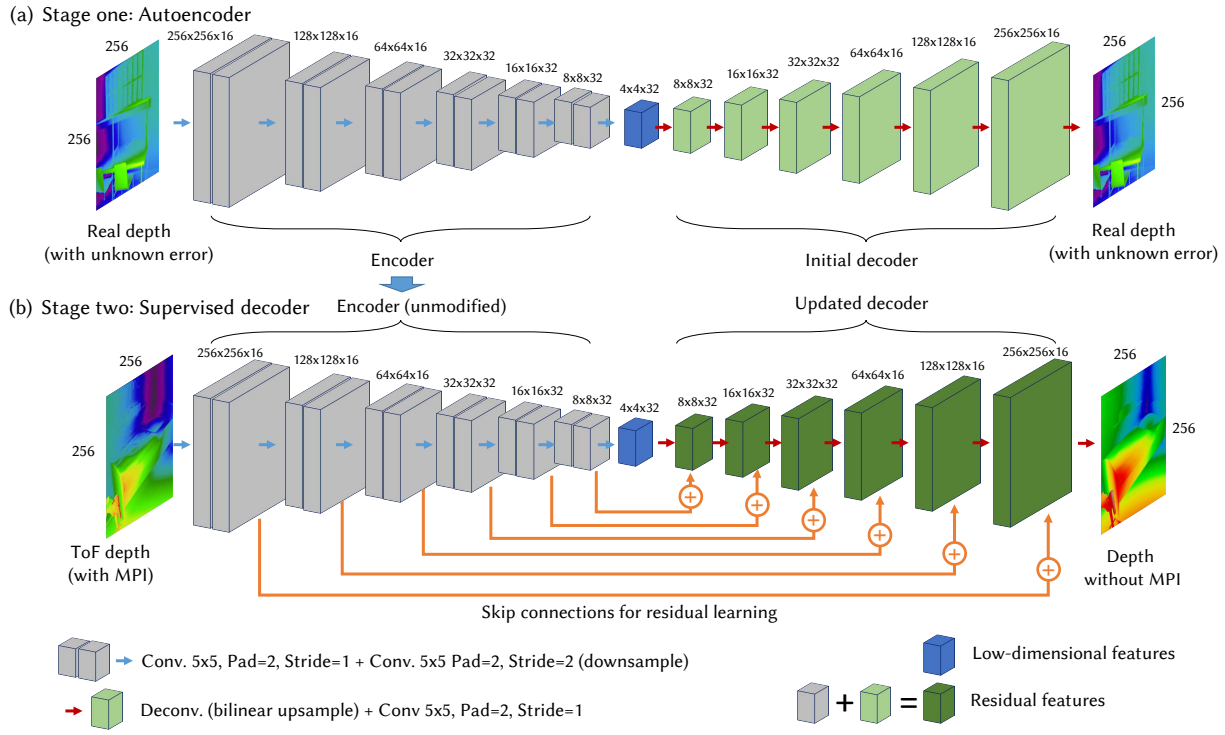

Fig. 6. Our two-stage training process for our regression network architecture. The first training stage learns a convolutional autocoder (an encoder-decoder pair) to learn lower-dimensional representations of depth. We use incorrect (with MPI) depth as both input and output in this unsupervised training. In the second, supervised training stage we input both incorrect depth with MPI and depth without MPI, and add skip connections for residual learning while fixing the weights of the encoder. This supervised training allows us to update the weights of a reconstruction decoder. Our final network is the pair of the original encoder and the updated decoder, working as a regression network. Please refer to the text for a more complete description.

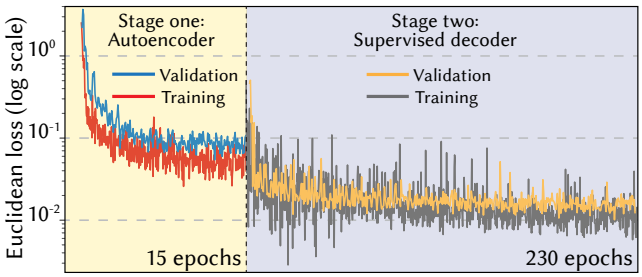

Fig. 7. Learning curves for our two-stage scheme. The first stage (left) learns lower-dimensional representations of real depths using an autoencoder. The second stage (right) learns MPI corrections in the decoder by training with our labeled synthetic data. Note that the first stage converges quickly and provides a good starting point for the second stage, which uses a different training set.

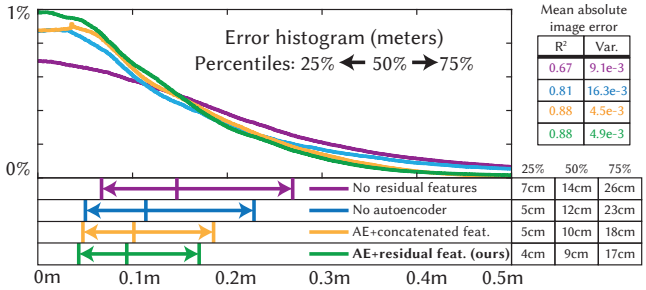

Fig. 8. Error distributions for different network alternatives: without residual connections (purple), without the autoencoding pre-initialization (blue), with concatenated skip connections (yellow), and our residual approach (green). Average R^2 across all predicted depth images shows that our residual learning with autoencoding pre-initialization reaches the best error distributions in the results, in terms of accuracy and low variance. The percentiles show that our approach presents also the best error distribution.

7.2 Comparison with Previous Work

In Figure 9 we compare our solution to previous works requiring no hardware modifications, and using a single frequency [Fuchs 2010; Jiménez et al. 2014]. We use both a synthetic and a real scene. Fuchs' approach [2010] results in a noisy estimation due to the discretization of light transport, taking around 10 minutes to compute. Jimenez et. al's technique [2014] is hindered by the geometrical complexity of the scenes, taking around one hour for a lower resolution image of 100×100; we were unable to compute larger images due to high memory consumption (about 60GB). Moreover, the results are very close in general to the input ToF captures, including MPI errors

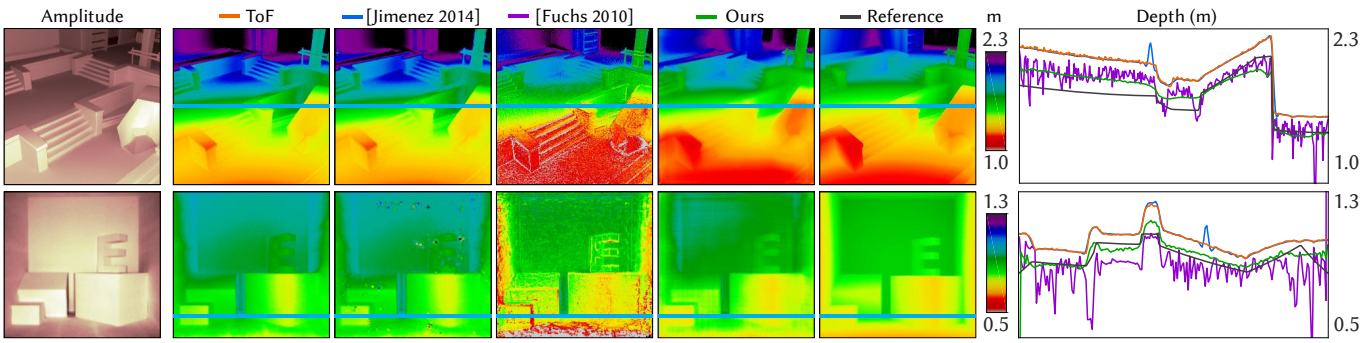

Fig. 9. Comparison with previous works. Top row: synthetic scene. Bottom row: real scene. Jimenez et. al' approach [2014] is hindered by the geometrical complexity of the scenes, and fails to correct MPI since the optimization converges to a local minimum. Fuchs' technique [2010] is closer to the reference, but is very noisy and greatly diverges in some regions. Our approach is the closest to the reference, as shown both in the depth maps and the graphs plotting the blue scanlines.

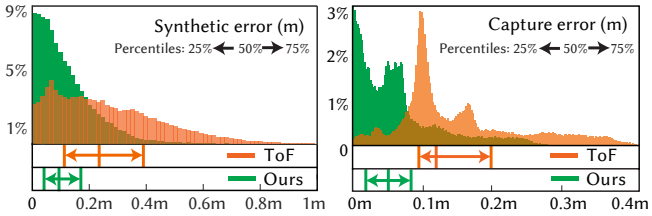

Fig. 10. Per-pixel distributions of absolute error for the synthetic validation dataset (left), and real dataset with measured ground truth (right). For each distribution, three percentiles (25%, 50% and 75%) are marked below. Our results clearly present a better error distribution.

and several outlier pixels. Our results are significantly closer to the reference, eliminating MPI errors, while being orders of magnitude faster.

7.3 Synthetic Scenes

From the synthetic dataset, a total of 213 scenes were used for the validation set (augmented to 1704 with flips and rotations). As Figure 10 (left) shows, our method yields a much better error distribution. Figure 11 shows a comparison of simulated ToF depth, our MPI-corrected depth, and the reference depth images. Our CNN preserves details while significantly mitigating MPI errors.

Additionally, in Figure 12 we compare the errors for five albedo combinations of three different scenes by randomly varying each object's reflectance between 0.3 and 0.8. It can be observed how our network is robust to these variations, consistently correcting depth errors due to MPI. Note how even under strong albedo changes on large flat objects (e.g., the cabinet in the first row, or the tabletop in the second) our network successfully recovers the correct depth.

7.4 Real Scenes

We now analyze the performance of our method in real scenes, captured with a PMD camera. We first show results on controlled scenarios with combinations of V-shapes, panels and a Cornell box, and then more challenging captures in the wild. The lens distortion of all the captures was corrected using a standard calibration of the intrinsic camera parameters, using a checkerboard pattern and captures at different distances [Bouguet 2004; Lindner et al. 2010].

Cornell box and V-shapes. We created different setups combining a Cornell box structure and V-shapes with flat panels (see Figure 13, left column). We accurately measured the geometry of these scenes, to create corresponding synthetic reference images for a quantitative analysis. The Cornell box dimensions were 600×500×640 mm, with additional panels from 400 mm to 1200 mm. The PMD camera was placed at multiple distances from 0.5 to 2.4 meters. We added several geometric elements to the scenes: three prisms with different dimensions, and a cardboard letter E, in order to add extra sources of MPI error. The surfaces of the Cornell box, two panels, and the smaller shapes were painted twice with a 50-50 mixture of barium sulfate and white matte paint, providing a good trade-off between durability and high-reflectance diffuse surfaces [Knighton and Bugbee 2005; Patterson et al. 1977]. Note that this mixture has an albedo of approximately 0.85, leading to large MPI and thus ensuring very challenging scenarios. Figure 13 (middle and right) shows the results of the captured depth and our corrected result. We compensate most of the MPI errors in both scenes, approximating depth much closer to the reference than the ToF camera. We can observe on the error distributions (right histogram in Figure 10) how our CNN manages to keep 50% of the per-pixel errors under 5 cm, while 75% of the errors in the PMD captures are *over* 9 cm.

Scenes in the wild. We now analyze several in-the-wild scenes, to illustrate the benefits of our approach in non-controlled conditions. The results are shown in Figure 14. We can see how our network successfully suppresses MPI in all cases, while still preserving details thanks to our residual learning approach. The magnitude of our MPI corrections is proportional to the measured ToF distance. This follows our observations in the error analysis (see Figure 5), where larger distances tend to yield larger errors since the relative error oscillates at around 10%.

7.5 Video in Real Time

Given the speed of our approach we are also able to process depth videos in real time. Regarding temporal coherence, we leverage the

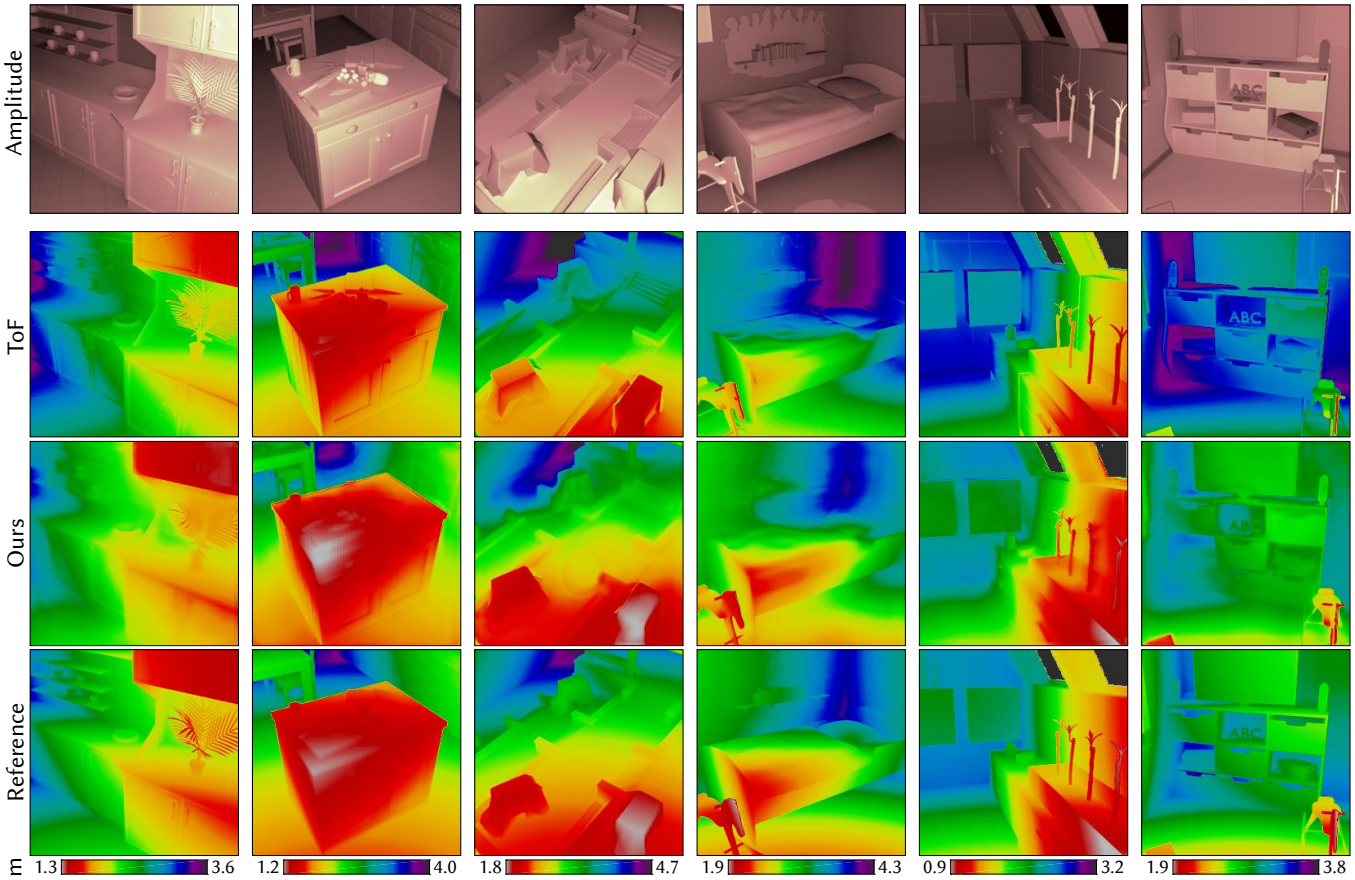

Fig. 11. Validation results for synthetic scenes (validation set) at varying distances between 0.5 and 7 meters. Top row shows ToF amplitude. Second, third and fourth rows show ToF depth with MPI, our estimated depth, and reference depth (without MPI). Our solution manages to correct MPI errors in a wide range of scenes while preserving details.

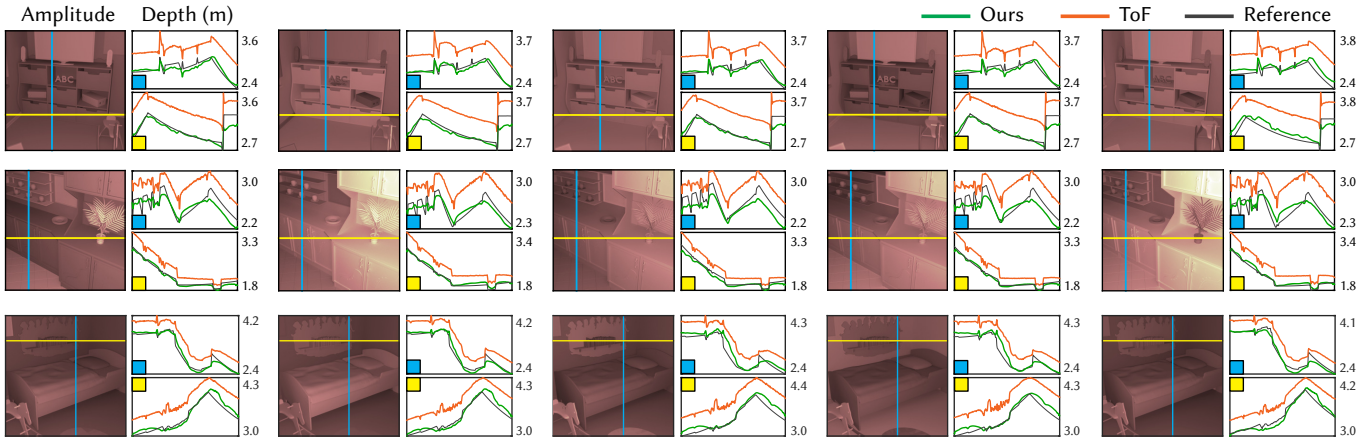

Fig. 12. Validation results for three scenes (one per row) with randomly varying albedos (one albedo set per column). Images show amplitude, vertical and horizontal plots show depth. Our approach is robust under arbitrary albedo variations.

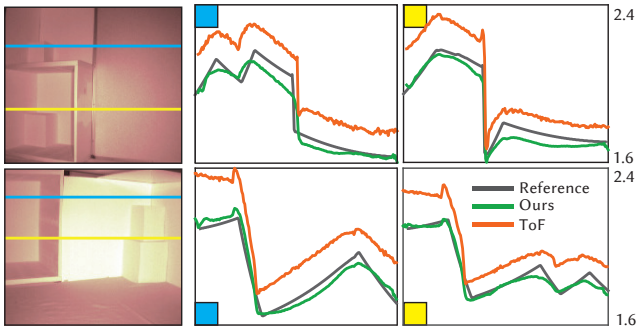

Fig. 13. Error comparison for different captured combinations of the Cornell box, panels and prisms. Reference depth solutions were obtained replicating the scenes in simulation. We significantly decrease MPI errors in all the captured scenes, yielding errors under 5 cm for the 50% of the pixels, as we demonstrate in the error histograms (Figure 10).

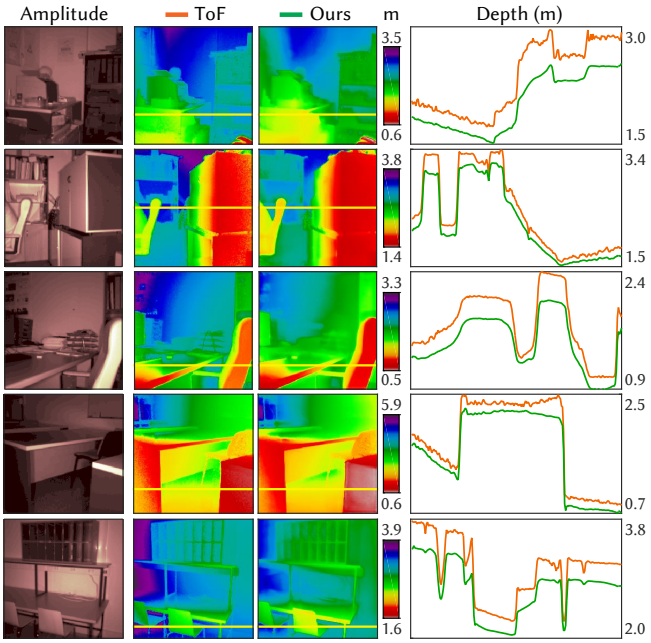

Fig. 14. Comparison of conventional ToF and our corrected depth, in real world scenes. In accordance with our analysis of MPI errors from our synthetic dataset, longer depths yield higher errors (about 10% of the measured distance).

fact that our input (incorrect depth) is quite stable between frames, so our network produces temporally coherent results without explicitly enforcing it. Other inputs such as amplitude and/or phase could show less stability, compromising this temporal coherence. We show some frames in Figures 1 and 15. Full sequences can be found in the supplemental video.

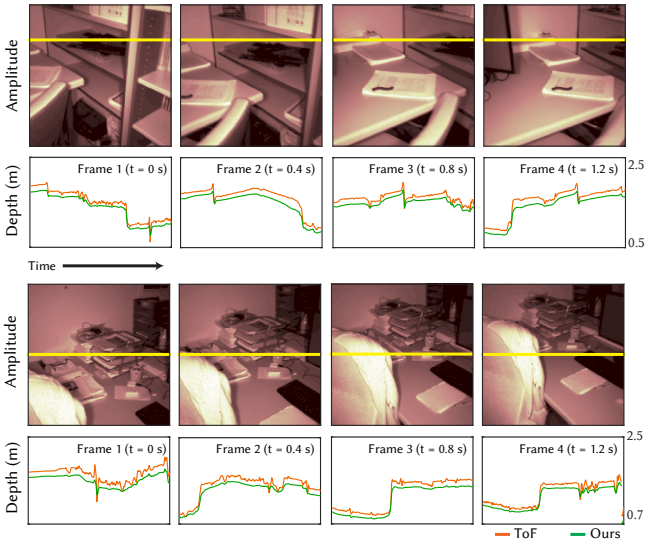

Fig. 15. Our approach can also be applied to correct MPI errors of in-the-wild videos of real scenes, in real time and keeping temporal consistency. Here we show the depth profiles of a few frames of two of our videos. The complete sequences can be found in the supplemental video.

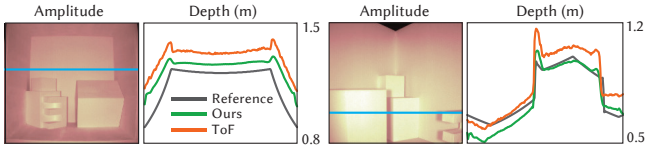

Fig. 16. Due to the high albedo of our barium-mixed diffuse paint (≈ 0.85), some specific camera-light configurations may yield large MPI errors (left). Objects very close to the camera (right, bottom-left box) yield a higher error since most of our training dataset has depths from 1.5m to 4m (see Figure 5). Still, our approach manages to improve MPI errors in both cases, providing results significantly closer to the reference depth.

8 DISCUSSION AND FUTURE WORK

We have presented a new approach for ToF imaging, to compensate the effect of multipath interference in real time, using an unmodified, off-the-shelf camera with a single frequency, and just the incorrect depth map as input. This is possible due to our carefully designed encoder-decoder (convolutional-deconvolutional) neural network, with a two-stage training process both from captured and synthetic data. With this paper, we provide our synthetic time-of-flight dataset that includes pairs of incorrect depth (affected by MPI) and its corresponding correct depth maps, as well as the trained network, for public use.

Several avenues of future work exist. First, as discussed in Section 5, we do not consider the wiggling error due to non-perfectly sinusoidal waves in our training dataset, since it is partially compensated by ToF cameras, and manufacturers do not provide information on this. If this information were available, we could incorporate the full camera pipeline (including non-sinusoidal waves and wiggling

correction) into our training dataset, and re-train our CNN accounting for these residual errors. In addition, there are still challenging scenarios where results could be improved, as shown in Figure 16. The very high albedo of barium-painted surfaces creates large MPI errors, specially under specific camera-light configurations (left). Although our MPI correction provides better results than the captured ToF depth, there is still some residual error of about 10 cm in average. Also, our network fails to correct MPI in the presence of objects which are very close to the camera, such as the bottom-left box in the second example (Figure 16, right). This is most likely because most of our synthetic dataset contains depths between 1.5m and 4m. Despite this, as we showed in Figure 10, per-pixel error distributions are significantly better than captured ToF depth.

Our work assumes diffuse (or nearly diffuse) reflectance. Although we have shown that it works well in several real-world scenarios with more general reflectances, it presents some problems in the presence of highly glossy materials. While incorporating such reflectances into our training dataset would help, our approach is likely to fail for extremely glossy or transparent surfaces; in such scenarios, other multi-frequency approaches [Kadambi et al. 2013; Qiao et al. 2015] could be better suited.

ACKNOWLEDGEMENTS

We want to thank the anonymous reviewers for their insightful comments, Belen Masia for proofreading the manuscript, and the members of the Graphics & Imaging Lab for helpful discussions. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (CHAMELEON project, grant agreement No 682080), DARPA (project REVEAL), and the Spanish Ministerio de Economía y Competitividad (projects TIN2016-78753-P and TIN2014-61696-EXP). Min H. Kim acknowledges Korea NRF grants (2016R1A2B2013031, 2013M3A6A6073718), Giga KOREA Project (GK17P0200) and KOCCA in MCST of Korea. Julio Marco was additionally funded by a grant from the Gobierno de Aragón.

APPENDICES

A LIGHT TRANSPORT IN IMAGE SPACE

Section 3 shows how most of the information on multipath interference from a scene is available in image space. This allows us to approximate Equation (4) by limiting the integration domain P to the differential paths $\overline{p}$ that reach pixel p from visible geometry. Moreover, given the discretized domain of an image, we can model Equation (4) using the transport matrix $\mathbf{T}_i$ [Ng et al. 2003; O'Toole et al. 2014], which relates the ideal response in pixel p_v with the outgoing response at pixel p_u, for a measurement phase shift i. Thus

$$\begin{aligned}\hat{\mathbf{c}}_i(p_u) &= \mathbf{c}_i(p_u) + \sum_v \mathbf{T}_i(p_u, p_v)\, \mathbf{c}_i(p_v) \\ &= \mathbf{c}_i(p_u) + \mathbf{T}_i * \mathbf{c}_i, \end{aligned} \tag{5}$$

where $*$ is the convolution in p_v, and $\mathbf{c}_i$ is the full phase-shifted image. While ideally this means that we can compute the correct phase-shifted image $\mathbf{c}_i$ by applying a deconvolution on the captured $\hat{\mathbf{c}}_i$, as $\mathbf{c}_i = \hat{\mathbf{c}}_i *_v (I + \mathbf{T}_i)^{-1}$ with I the identity matrix, in practice this is not possible since the transport matrix $\mathbf{T}_i$ is unknown. Capturing it is an expensive process, and we cannot make strong simplifying assumptions on the locality of light transport (i.e. sparsity in $\mathbf{T}_i$), since light reflected from far away pixels might have an important contribution on pixel p. However, as we show in Section 4, we can learn the resulting deconvolution operator by means of a convolutional neural network.

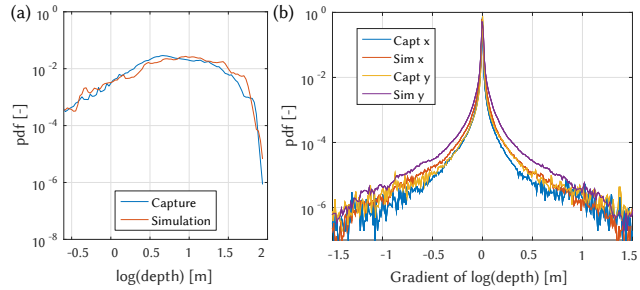

Fig. 17. (a) Logarithmic depth histogram of real and simulated depths, showing a good match between both sets. (b) Derivative in x and y directions of real and simulated depths, also showing a similar trend. Please refer to the text for quantitative data and other statistical analyses.

B DEPTH STATISTICS

To validate our synthetic dataset (Section 5), we follow previous works on depth image statistics [Huang et al. 2000; Huang and Mumford 1999] on both real and synthetic datasets. In particular, we analyze single-pixel, derivative and bivariate statistics, as well as joint statistics of Haar wavelet coefficients. To compare the results we use three different metrics: chi-squared error [Pele and Werman 2010], the Jensen-Shannon distance [Endres and Schindelin 2003; Lin 1991], and Pearson's correlation coefficient. The first is a weighted Euclidean error ranging from 0 to ∞ (less is better); the second one measures the similarity between two distributions, ranging from 0 to $\sqrt{ln(2)} = 0.833$ (less is better); the last measures correlation, where a value of 0 indicates two independent variables, and ± 1 indicates a perfect linear direct or inverse relationship.

Figure 17a shows the depth histograms of both sets of images. They are very similar, with a chi-squared error of 0.032, Jensen-Shannon distance of 0.129, and a correlation of 0.90. Figure 17b shows how the vertical and horizontal gradients of both datasets also follow a similar trend, with a chi-squared error of 0.051 and 0.028, Jensen-Shannon distance of 0.162 and 0.118, and correlation coefficients of 0.948 and 0.969 for the vertical and horizontal gradients respectively. Both sets have high kurtosis values, as reported by Huang [2000].

Next, we carried out a bivariate statistical analysis to pairs of pixels at a fixed separation distance, following the co-ocurrence equation [Huang et al. 2000; Lee et al. 2000]. Capture and simulation match with a chi-squared error of 0.094, Jensen-Shannon distance of 0.245, and correlation of 0.915. Last, we selected three Haar filters (horizontal, vertical, and diagonal) in order to analyze joint statistics in the wavelet domain. Capture and simulation match with a chi-squared error of 0.152 and 0.172, Jensen-Shannon distance of 0.314 and 0.336, and correlation coefficients of 0.892 and 0.896, for the horizontal-vertical and horizontal-diagonal pairs respectively.

These values indicate that our simulated images share very similar depth statistics with existing real-world depth images, so they can be used as reliable input for our network.

REFERENCES

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

Jean-Yves Bouguet. 2004. Camera calibration toolbox for Matlab. (2004).

Inchang Choi, Daniel S. Jeon, Giljoo Nam, Diego Gutierrez, and Min H. Kim. 2017. High-Quality Hyperspectral Reconstruction Using a Spectral Prior. *ACM Transactions on Graphics (SIGGRAPH Asia 2017)* 36, 6 (2017).

A. A. Dorrington, J. P. Godbaz, M. J. Cree, A. D. Payne, and L. V. Streeter. 2011. Separating true range measurements from multi-path and scattering interference in commercial range cameras. In *Proceedings of SPIE*, Vol. 7864. 786404–786404–10.

B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao. 2016. Stacked Convolutional Denoising Auto-Encoders for Feature Representation. *IEEE Trans. Cybernetics* 99 (2016), 1–11.

David Eigen and Rob Fergus. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of ICCV*.

David Eigen, Christian Puhrsch, and Rob Fergus. 2014. Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of NIPS*.

Dominik Maria Endres and Johannes E Schindelin. 2003. A new metric for probability distributions. *IEEE Transactions on Information theory* 49, 7 (2003), 1858–1860.

M. Feigin, A. Bhandari, S. Izadi, C. Rhemann, M. Schmidt, and R. Raskar. 2016. Resolving Multipath Interference in Kinect: An Inverse Problem Approach. *IEEE Sensors Journal* 16, 10 (May 2016), 3419–3427.

Daniel Freedman, Yoni Smolin, Eyal Krupka, Ido Leichter, and Mirko Schmidt. 2014. SRA: Fast removal of general multipath for ToF sensors. In *Proceedings of ECCV*. Springer, 234–249.

Stefan Fuchs. 2010. Multipath Interference Compensation in Time-of-Flight Camera Images. In *Proceedings of the International Conference on Pattern Recognition*. 3583–3586.

Stefan Fuchs, Michael Suppa, and Olaf Hellwich. 2013. Compensation for Multipath in ToF Camera Measurements Supported by Photometric Calibration and Environment Integration. In *Proceedings of the International Conference on Computer Vision Systems (ICVS'13)*. Springer-Verlag, Berlin, Heidelberg, 31–41.

John P. Godbaz, Michael J. Cree, and Adrian A. Dorrington. 2012. Closed-form inverses for the mixed pixel/multipath interference problem in AMCW lidar. In *Proceedings of SPIE*, Vol. 8296. 829618–829618–15.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Mohit Gupta, Shree K. Nayar, Matthias B. Hullin, and Jaime Martin. 2015. Phasor Imaging: A Generalization of Correlation-Based Time-of-Flight Imaging. *ACM Trans. Graph.* 34, 5, Article 156 (Nov. 2015), 18 pages.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of CVPR*. 770–778.

Felix Heide, Matthias B. Hullin, James Gregson, and Wolfgang Heidrich. 2013. Low-budget Transient Imaging Using Photonic Mixer Devices. *ACM Trans. Graph.* 32, 4, Article 45 (July 2013), 10 pages.

Jinggang Huang, Ann B Lee, and David Mumford. 2000. Statistics of range images. In *Proceedings of CVPR*, Vol. 1. IEEE, 324–331.

Jinggang Huang and David Mumford. 1999. Statistics of natural images and models. In *Proceedings of CVPR*, Vol. 1. IEEE, 541–547.

Adrian Jarabo, Julio Marco, Adolfo Muñoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez. 2014. A Framework for Transient Rendering. *ACM Trans. Graph.* 33, 6, Article 177 (2014).

Adrian Jarabo, Belen Masia, Julio Marco, and Diego Gutierrez. 2017. Recent Advances in Transient Imaging: A Computer Graphics and Vision Perspective. *Visual Informatics* 1, 1 (2017).

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

Achuta Kadambi, Refael Whyte, Ayush Bhandari, Lee Streeter, Christopher Barsi, Adrian Dorrington, and Ramesh Raskar. 2013. Coded Time of Flight Cameras: Sparse Deconvolution to Address Multipath Interference and Recover Time Profiles. *ACM Trans. Graph.* 32, 6, Article 167 (Nov. 2013), 10 pages.

Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. 2016. Learning-based View Synthesis for Light Field Cameras. *ACM Trans. Graph.* 35, 6, Article 193 (Nov. 2016), 10 pages.

S Karayev, Y Jia, J Barron, M Fritz, K Saenko, and T Darrell. 2011. A category-level 3-D object dataset: putting the Kinect to work. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 1167–1174.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Ahmed Kirmani, Arrigo Benedetti, and Philip A Chou. 2013. Spumic: Simultaneous phase unwrapping and multipath interference cancellation in time-of-flight cameras using spectral methods. In *Proceedings of IEEE International Conference on Multimedia and Expo*. 1–6.

Nick Knighton and Bruce Bugbee. 2005. A mixture of barium sulfate and white paint is a low-cost substitute reflectance standard for Spectralon®. (2005).

Ann B Lee, JG Huang, and DB Mumford. 2000. Random collage model for natural images. *Int. J. of Computer Vision* (2000).

Bo Li, Chunhua Shen, Yuchao Dai, A. van den Hengel, and Mingyi He. 2015. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In *Proceedings of CVPR*.

Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory* 37, 1 (1991), 145–151.

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

Fayao Liu, Chunhua Shen, and Guosheng Lin. 2015. Deep Convolutional Neural Fields for Depth Estimation from a Single Image. In *Proceedings of CVPR*.

Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Proceedings of Int. Conf. Artificial Neural Networks*. 52–59.

Nikhil Naik, Achuta Kadambi, Christoph Rhemann, Shahram Izadi, Ramesh Raskar, and Sing Bing Kang. 2015. A light transport model for mitigating multipath interference in time-of-flight sensors. In *Proceedings of CVPR*. 73–81.

Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. 2003. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph.* 22, 3 (2003).

Matthew O'Toole, Felix Heide, Lei Xiao, Matthias B. Hullin, Wolfgang Heidrich, and Kiriakos N. Kutulakos. 2014. Temporal Frequency Probing for 5D Transient Analysis of Global Light Transport. *ACM Trans. Graph.* 33, 4, Article 87 (July 2014), 11 pages.

EM Patterson, CE Shelden, and BH Stockton. 1977. Kubelka-Munk optical properties of a barium sulfate white reflectance standard. *Applied Optics* 16, 3 (1977), 729–732.

Ofir Pele and Michael Werman. 2010. The quadratic-chi histogram distance family. In *Proceedings of ECCV*. Springer, 749–762.

Christoph Peters, Jonathan Klein, Matthias B. Hullin, and Reinhard Klein. 2015. Solving Trigonometric Moment Problems for Fast Transient Imaging. *ACM Trans. Graph.* 34, 6 (Nov. 2015).

Hui Qiao, Jingyu Lin, Yebin Liu, Matthias B Hullin, and Qionghai Dai. 2015. Resolving transient time profile in ToF imaging via log-sum sparse regularization. *Opt. Lett.* 40, 6 (2015).

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from RGBD images. *Computer Vision–ECCV 2012* (2012), 746–760.

Hao Su, Haoqiang Fan, and Leonidas Guibas. 2017. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In *Proceedings of CVPR*.

Andreas Velten, Di Wu, Adrian Jarabo, Belen Masia, Christopher Barsi, Chinmaya Joshi, Everett Lawson, Moungi G. Bawendi, Diego Gutierrez, and Ramesh Raskar. 2013. Femto-Photography: Capturing and Visualizing the Propagation of Light. *ACM Transactions on Graphics (SIGGRAPH 2013)* 32, 4 (2013).

Peng Wang, Xiaohui Shen, Zhe Lin, S. Cohen, B. Price, and A. Yuille. 2015. Towards unified depth and semantic prediction from a single image. In *Proceedings of CVPR*.

Di Wu, Andreas Velten, Matthew O'Toole, Belen Masia, Amit Agrawal, Qionghai Dai, and Ramesh Raskar. 2014. Decomposing Global Light Transport Using Time of Flight Imaging. *International Journal of Computer Vision* 107, 2 (April 2014), 123 – 138.

Jianxiong Xiao, Andrew Owens, and Antonio Torralba. 2013. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of ICCV*. 1625–1632.

Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. 2017. Multi-Scale Continuous CRFs as Sequential Deep Networks for Monocular Depth Estimation. In *Proceedings of CVPR*.

Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma. 2010. Image Super-Resolution Via Sparse Representation. *IEEE TIP* 19, 11 (2010).

Jure Žbontar and Yann LeCun. 2015. Computing the Stereo Matching Cost with a Convolutional Neural Network. In *Proceedings of CVPR*.