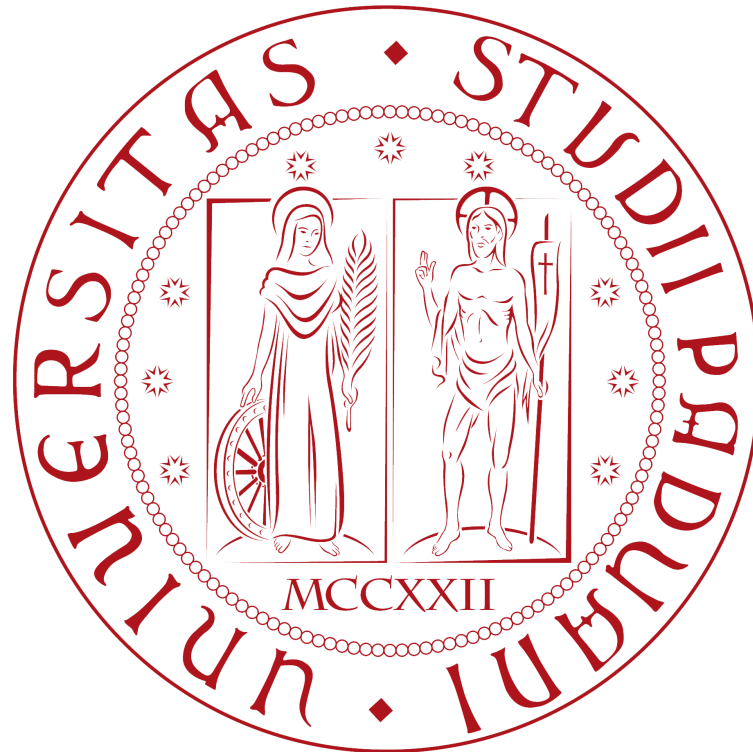


UNIVERSITY OF PADOVA

ACADEMIC YEAR 2019-2020 (797TH)

DEPARTMENT OF INFORMATION ENGINEERING (DEI)

DOCTORAL THESIS IN INFORMATION ENGINEERING



DATA DRIVEN APPROACHES FOR DEPTH DATA DENOISING

AUTHOR: GIANLUCA AGRESTI

SUPERVISOR: PIETRO ZANUTTIGH

CO-SUPERVISOR: HENRIK SCHAEFER

A Silvana, che mi supporta e mi è vicina sempre.

Abstract

The scene depth is an important information that can be used to retrieve the scene geometry, a missing element in standard color images. For this reason, the depth information is usually employed in many applications such as 3D reconstruction, autonomous driving and robotics.

The last decade has seen the spread of different commercial devices able to sense the scene depth. Among these, Time-of-Flight (ToF) cameras are becoming popular because they are relatively cheap and they can be miniaturized and implemented on portable devices. Stereo vision systems are the most widespread 3D sensors and they are simply composed by two standard color cameras. However, they are not free from flaws, in particular they fail when the scene has no texture. Active stereo and structured light systems have been developed to overcome this issue by using external light projectors.

This thesis collects the findings of my Ph.D. research, which are mainly devoted to the denoising of depth data. First, some of the most widespread commercial 3D sensors are introduced with their strengths and limitations. Then, some techniques for the quality enhancement of ToF depth acquisition are presented and compared with other state-of-the-art methods. A first proposed method is based on a hardware modification of the standard ToF projector. A second approach instead uses multi-frequency ToF recordings as input of a deep learning network to improve the depth estimation. A particular focus will be given to how the denoising performance degrades, when the network is trained on synthetic data and tested on real data. Thus, a method to reduce the gap in performance will be proposed. Since ToF and stereo vision systems have complementary characteristics, the possibility to fuse the information coming from these sensors is analysed and a method based on a locally consistent fusion, guided by a learning based reliability measure for the two sensors, is proposed. A part of this thesis is dedicated to the description of the data acquisition procedures and the related labeling required to collect the datasets we used for the training and evaluation of the proposed methods.

Sommario

La profondità della scena è un'importante informazione che può essere usata per recuperare la geometria della scena stessa, un elemento mancante nelle semplici immagini a colori. Per questo motivo, questi dati sono spesso usati in molte applicazioni come ricostruzione 3D, guida autonoma e robotica.

L'ultima decade ha visto il diffondersi di diversi dispositivi capaci di stimare la profondità di una scena. Tra questi, le telecamere Time-of-Flight (ToF) stanno diventando sempre più popolari poiché sono relativamente poco costose e possono essere miniaturizzate e implementate su dispositivi portatili. I sistemi a visione stereoscopica sono i sensori 3D più diffusi e sono composti da due semplici telecamere a colori. Questi sensori non sono però privi di difetti, in particolare non riescono a stimare in maniera corretta la profondità di scene prive di texture. I sistemi stereoscopici attivi e i sistemi a luce strutturata sono stati sviluppati per risolvere questo problema usando un proiettore esterno.

Questa tesi presenta i risultati che ho ottenuto durante il mio Dottorato di Ricerca presso l'Università degli Studi di Padova. Lo scopo principale del mio lavoro è stato quello di presentare metodi per il miglioramento dei dati 3D acquisiti con sensori commerciali. Nella prima parte della tesi i sensori 3D più diffusi verranno presentati introducendo i loro punti di forza e debolezza. In seguito verranno descritti dei metodi per il miglioramento della qualità dei dati di profondità acquisiti con telecamere ToF. Un primo metodo sfrutta una modifica hardware del proiettore ToF. Il secondo utilizza una rete neurale convoluzionale (CNN) che sfrutta dati acquisiti da una telecamera ToF per stimare un'accurata mappa di profondità della scena. Nel mio lavoro è stata data attenzione a come le prestazioni di questo metodo peggiorano quando la CNN è allenata su dati sintetici e testata su dati reali. Di conseguenza, un metodo per ridurre tale perdita di prestazioni verrà presentato. Poiché le mappe di profondità acquisite con sensori ToF e sistemi stereoscopici hanno proprietà complementari, la possibilità di fondere queste due sorgenti di informazioni è stata investigata. In particolare, è stato presentato un metodo di fusione che rinforza la consistenza locale dei dati e che sfrutta una stima dell'accuratezza dei due sensori, calcolata con una CNN, per guidare il processo di fusione. Una parte della tesi è dedicata alla descrizione delle procedure di acquisizione dei dati utilizzati per l'allenamento e la valutazione dei metodi presentati.

Contents

1	Introduction	1
2	3D Sensing Devices	3
2.1	Stereo Vision Systems	4
2.1.1	Stereo Matching Methods	6
2.1.2	Limitations of Stereo Vision Approaches	6
2.2	Active Triangulation Systems	8
2.3	Time-of-Flight Cameras	11
2.3.1	Continuous Wave ToF	12
2.3.2	Error Sources in ToF Recordings	14
3	Stereo and ToF Dataset Collection	21
3.1	Introduction to the Depth Datasets	22
3.2	Real Datasets	22
3.2.1	Proposed Multi-Frequency ToF Datasets	25
3.3	Synthetic Datasets	33
3.3.1	\mathbf{S}_1 : The Synthetic DS541 Dataset	33
3.3.2	SYNTH3	34
4	ToF Depth Data Refinement	37
4.1	Literature about ToF Data Refinement	37
4.2	Spatio-Temporal Modulated ToF	40
4.2.1	Introduction to STM-ToF	41
4.2.2	Error Propagation Analysis	44
4.2.3	Applying Structured Light on ToF Sensors	45

4.2.4	Fusion of ToF and SL Depth Maps	47
4.2.5	Results of the Fusion Method on the STM-ToF	49
4.3	Data Driven ToF Data Refinement	54
4.3.1	Introduction to Covolutional Neural Networks	55
4.3.2	R-CNN+B: MPI Estimation and Noise Filtering	59
4.3.3	TD-CNN: Depth Denoiser CNN	65
4.3.4	Training and Test Datasets	67
4.3.5	Results of the Proposed Data Driven Methods	68
5	Domain Adaptation	75
5.1	Domain Adaptation for Neural Networks	76
5.2	Introduction to Generative Adversarial Networks	77
5.3	Domain Adaptation for ToF Data Refinement	79
5.3.1	Proposed Method	80
5.3.2	Generator Network	80
5.3.3	Discriminator Network	81
5.3.4	Adversarial Learning Strategy	83
5.3.5	Synthetic and Real World Datasets	87
5.3.6	Experimental Results	88
5.4	Domain Adaptation on Semantic Segmentation	93
5.4.1	Architecture of the Proposed Approach	94
5.4.2	Datasets	98
5.4.3	Experimental Results	100
5.4.4	Ablation Study	104
6	Fusion of ToF and Stereo Data	107
6.1	Literature about Stereo-ToF Fusion	107
6.2	Stereo-ToF Fusion Guided by Learned Confidences	109
6.2.1	Confidence Estimation with Deep Learning	111
6.2.2	Training of the Convolutional Neural Network	113
6.2.3	Fusion of Stereo and ToF Disparity	115
6.2.4	Experimental Results	116
7	Conclusions	131

A Error Propagation Analysis	133
B Noise Variance due to Photon Shot Noise on ToF Recordings	135
C STM-ToF Correlation Function Evaluation and Error Propagation Analysis	139
C.1 Error Propagation Analysis on STM-ToF	140
C.2 Structured Light Depth Estimation with Implicit Phase Unwrapping	144
C.3 Error Estimation of the Structured Light Approach	145
Bibliography	147
Acknowledgments	159

Chapter 1

Introduction

Traditional color cameras are able to record the photometric appearance of the surrounding environment by projecting the color information on the image plane. This acquisition process does not capture the scene depth information, that is strictly related to the scene 3D structure. This missing information can have an important role in many computer vision applications such as autonomous driving, gesture recognition and robotics, where the scene geometry can be used to have a better understanding of the problem and its solution. For this reason, the last decade has seen a wide spread of depth sensing devices.

The first considered commercial depth sensors, also the simplest from a hardware point of view, are the stereo vision systems. These are composed by a couple of color cameras and employ the triangulation principle to estimate the depth of a scene. Regardless of their hardware simplicity, they use advanced techniques to locate where the target scene point is projected on the two color views. These systems are prone to errors in regions where no visual features are possible to be uniquely recognized, as in case of textureless flat walls or repeating patterns. Active stereo and structured light systems exploit the triangulation principle as well, but they use an external light projector to solve the issue related to the scene texture affecting the passive stereo vision systems.

Time-of-Flight (ToF) cameras are depth devices implementing a completely different working principle. They are equipped with a light projector, that lights the scene with an amplitude modulated light signal, and a special type of pixels, able

to correlate the received modulated light with a reference signal. In this way, they can estimate the time delay between the transmitted and the received light signal. This information can be used to estimate the scene depth, by assuming that the speed of light is constant in the air. These sensors can record depth maps at video frame rates, but due to the complexity of their pixels they have a limited spatial resolution, if compared to standard cameras.

Each of these types of commercial and portable depth sensors has its own strengths and weaknesses and they are somehow complementary on these. This thesis will introduce the flaws of these devices and, moreover, possible solutions will be proposed and tested. A particular attention will be given to ToF data denoising and the fusion of ToF and stereo vision systems data to improve the depth estimation accuracy.

The results collected in this thesis are the outcome of the work I have carried out during the three years of my Ph.D. program. My scholarship has been funded by Sony Europe and I have spent one year of my Ph.D. working in the Sony Eutec research center located in Stuttgart, Germany. There, I had the opportunity to be mentored by experts of ToF sensors and to use new ToF prototypes.

In the first chapter of this thesis, the aforementioned 3D sensors will be described in detail. Their working principles will be introduced and their main flaws will be analysed. Depth refinement techniques have to be validated and for this reason depth datasets have been collected. Chapter 3 will describe the procedure used to collect these depth data with the related ground truth depth. The developed ToF depth refinement techniques are introduced in Chapter 4. In particular, two approaches will be presented. The first [1] is based on a hardware modification of the commercial ToF projector and on a traditional signal processing approach. The second [2] can be applied on off-the-shelf ToF cameras and it is based on a deep learning strategy. Since the second approach is trained on synthetic data, its performance worsen when it is tested on real data. This is the *domain shift* issue and it will be analysed on Chapter 5. In this chapter, this issue will be considered in the case of deep learning approaches used to refine ToF data [3] and when the task is semantic segmentation [4]. Chapter 6 will focus on the complementarity of ToF and stereo vision systems. A method for the fusion of the two depth sources [5,6] will be presented and compared with other existing fusion methods.

Chapter 2

3D Sensing Devices

Nowadays, commercial depth sensing devices generally exploit the *reflective* properties of the scene. This family of sensors estimates the scene depth by evaluating the electromagnetic radiation in the target environment. Each specific depth sensing approach uses a different band of the spectrum and a specific way to interpret it. Indeed, it is possible to have optical devices working in the visible or in the infrared spectrum. Differently, other sensors as radars use radio waves to determine the position and the velocity of the target. Fig. 2.1, derived from [7], introduces a representation of this family of range devices.

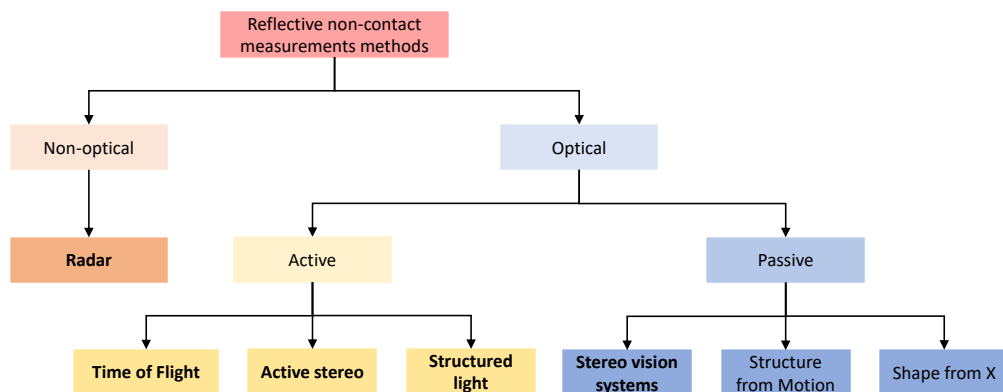


Fig. 2.1: Representation of the different families of reflective depth sensing devices.

This thesis mainly focuses on *optical* sensors and how to improve the level of accuracy in depth estimation. There exist two families of *optical* depth sensors: *passive* and *active* sensors. The passive sensors estimate the depth by exploiting the optical radiation already present in the scene. Stereo vision systems are the most widely used technology belonging to this family of devices. The active sensors instead employ an external projector to radiate the scene with a specific light signal, usually in the infra-red spectrum. This light signal can be modulate in space, as for active stereo and structured light sensors, or in time, as for Time-of-Flight (ToF) cameras.

The next of this chapter introduces the basic working mechanism together with their strengths and flaws of the aforementioned devices, since they will be the focus of this thesis.

2.1 Stereo Vision Systems

Stereo vision systems are passive depth sensors. These are composed by a couple of standard color cameras employing the *triangulation* principle to estimate the scene depth in the region framed by the two cameras. By looking at the same point from two slightly different view points it is possible to infer the depth, similarly to what we do with our eyes. Fig. 2.2 depicts a ZED stereo vision system [8], one of the most employed commercial stereo vision systems. It is possible notice the two color cameras on the sides of the devices able to acquire a scene depth map till 2K spatial resolution.



Fig. 2.2: ZED stereo system, an example of commercial stereo vision system [8].

Before estimating the depth, the standard cameras have to be geometrically

calibrated [9] and then rectified [10]. After these processes, the images recorded by the two cameras are projected on the same image plane and so the epipolar lines are made parallel and coincided to the same pixel row on the two pixel grids.

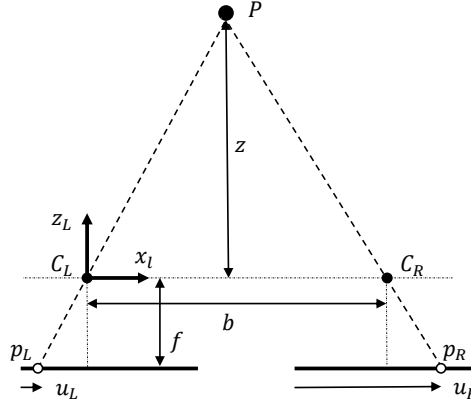


Fig. 2.3: Depth estimation by triangulation on two calibrated and rectified cameras composing a stereo vision system.

After the calibration and rectification procedures, a given scene point $P = (x, y, z)$ is observed on the two cameras by pixels belonging to the same row, but different column position. If we are able to locate the pixel positions, e.g., $p_L = (u_L, v_L)$ and $p_R = (u_R, v_R)$ respectively in the left and in the right camera, with $v_L = v_R$ because the vertical coordinate is correspondent after rectification [11, 12], then the depth along the Cartesian axis z perpendicular to the image planes can be estimated as

$$z = \frac{b f}{u_L - u_R} = \frac{b f}{disp} \quad (2.1)$$

where, b is the *baseline* of the stereo vision system, that is the distance between the optical center of the cameras C_L and C_R , and f is their focal length, identical after rectification. $disp$ is the *disparity*, that is equal to $u_L - u_R$. Eq. 2.1 can be extrapolated by exploiting the properties of the similar triangles in Fig. 2.3.

To estimate the depth with Eq. 2.1, it is required to estimate the pixel positions u_R and u_L . The search of these conjugated points is named *correspondence problem* and it can be solved with a *stereo matching* method. The next section introduces various techniques to implement it.

2.1.1 Stereo Matching Methods

Stereo matching methods have the task to find the conjugated points in a couple of rectified camera views. This is the most critical problem in stereo vision depth estimation, since it is strictly related to the disparity and the depth estimation. These methods can be classified as *local* or *global*. The first look just for local similarity in the neighbourhood of the target pixel, by enforcing the fronto-parallel hypothesis, that is by assuming that pixels near to each other share the same disparity value. This can be built by computing a cost function that compares a window centred on the target pixel on the left image with a window on the right whose position is given by a disparity hypothesis. The disparity can be selected by minimizing the cost function and the disparity value is selected in a winner-take-all fashion. Different cost functions can be used, the most common are Sum of Absolute Difference (SAD), Sum of Squared Difference (SSD), census or Normalized Cross-Correlation (NCC) [13]. Local methods can be implemented very efficiently but they have poor performance in particular in flat regions or with repeating patterns.

Differently, global approaches estimate the disparity by minimizing a cost that is function of the whole images. A possibility is to use a Markov Random Field (MRF) in combination with some smoothness criterion taking in account the eventual presence of depth discontinuities [14]. In general, global methods are more accurate than local ones, but they are computationally more expensive.

A trade off is given by semi-global approaches as Hirsh Muller's SGM [15]. It uses a point-wise cost function as local methods and a smoothness term that tries to enforce the consistency among pixels belonging to paths in the images. Since it can be implemented efficiently, it is one of the most diffuse stereo matching methods.

Recently, deep learning techniques for stereo depth estimation have been proposed [16–18]. They proved to be more accurate than traditional approaches. The strength of these methods relies on learning based features which exploit semantic information about the scene.

2.1.2 Limitations of Stereo Vision Approaches

When the stereo matching does not fail and it is able to locate the position of the conjugate points, stereo vision systems are accurate depth sensors. An error

propagation analysis on Eq. 2.1 can be applied to evaluate what are the elements influencing the overall depth estimation. More details about error propagation can be found in Appendix A. From this analysis, the noise standard deviation can be formulated as

$$\sigma_z = \frac{z^2}{b f} \sigma_{disp} \quad (2.2)$$

where it comes out that it is directly proportional to the squared depth. This means that the depth estimation is very accurate in the near range and the accuracy dramatically decreases in the far range. Eq. 2.2 shows that it is possible to improve the depth estimation performance by increasing the baseline b , so the distance between the two cameras. These are related to the system dimension and a trade off between the device size and the maximum range of applicability has to be found.

The above mentioned analysis is valid in case the stereo matching algorithm doesn't fail. In the next of this section the most critical flaws of stereo matching are listed.

Occlusions

Stereo matching algorithms have to find conjugate points on the two images. A first issue can arise if a scene point is visible only from a single camera. This is the case of occlusions, which can arise near to depth discontinuity. This issue can be detected by applying a left-right check, that is by computing the disparity map in both directions, from the left to the right image and vice versa, and checking their consistency.

Depth Edges

In the matching cost, problems can arise on the recovery of depth edges. The neighbourhood of the target point can be different in the two views, since they are composed by pixels at different depths, which appear differently on the cameras placed in different positions. This issue can be mitigated by decreasing the window size, used in the block matching methods, to increase the level of details in the depth map at the cost of a reduced resilience to noise. Another possible solution is to use matching methods which are not based on square blocks but using more

elaborate techniques, e.g., segmentation based matching methods, which are aware of the image structure.

Non-Lambertian Surfaces

For a correct matching, the same object needs to have the same photometric appearance on the two images. This assumption is not valid on non-Lambertian objects as in case of mirrors or elements composed by glass or metal producing a high amount of specular reflections.

Texture Dependency

Stereo matching algorithms try to find the projection of the same scene point on the two cameras. These elements can be located if they are characterized by well recognizable visual features. However, this is not the case for points belonging to textureless surfaces, e.g., a white wall in a room. In this situation, the matching algorithm can not find local features to estimate the correct couple of conjugate points, since the cost function becomes flat and multiple minima exist. A similar issue is given by repeating patterns, since also in this case the cost function can have multiple minima making the recovery of the correct couple of conjugate points ill-posed. Local matching approaches are particularly sensible to these scenarios. Differently, global matching functions or learning based methods try to mitigate these issues by enforcing consistency in a wide region of the scene. Active stereo systems and structured light systems try to solve in a more robust way the texture dependent issues. They use an external light projector to label each scene point with a code that can be used as a texture in the matching process. The next section introduces these devices.

2.2 Active Triangulation Systems

As mentioned in the previous section, the performance of stereo vision systems are very sensible to the texture contained in the scene, since this is used in the stereo matching process to locate conjugate points. Active triangulation systems try to solve this issue by employing a light projector to label each scene point with a

unique light code that can be used to locate it in the two color views. This kind of devices can be:

- *active stereo systems*, if the projected pattern is used by the cameras as an artificial texture that simplifies the stereo matching;
- *structured light systems*, if the triangulation is applied between the projector pixels and the standard cameras (see [19]). In general, structured light systems can be composed also by a single camera and a projector, but a careful calibration of the two is required.



Fig. 2.4: Intel RealSense DS435, an example of commercial active stereo system. [20]

Fig 2.4 depicts an Intel RealSense DS435 [20], an example of commercial active stereo system. It is composed by two standard cameras, placed on the sides of the device, and an infra-red light projector, placed in the middle. In general, in active triangulation systems the light projector illuminates the scene with a pattern that labels each scene point with unique well defined codeword. The way to create the codebook differentiate the typology of patterns. Different techniques to create the code exist and the most employed and robust are based on *spatial* or *temporal* modulation.

In case of spatial modulation, the codewords are given by the pattern projected on a certain neighbourhood of the target point. Some solutions use bi-level on-off illumination in which the codewords are based on the De Bruijn pseudo-random code [23]. Fig. 2.5 (a) shows an example of such patterns. This typology of patterns can be used for one shot depth acquisition, since the code is contained in each single image recorded by the cameras. However, the depth recovery near to depth discontinuities can be problematic if the window containing the spatial code is distorted.

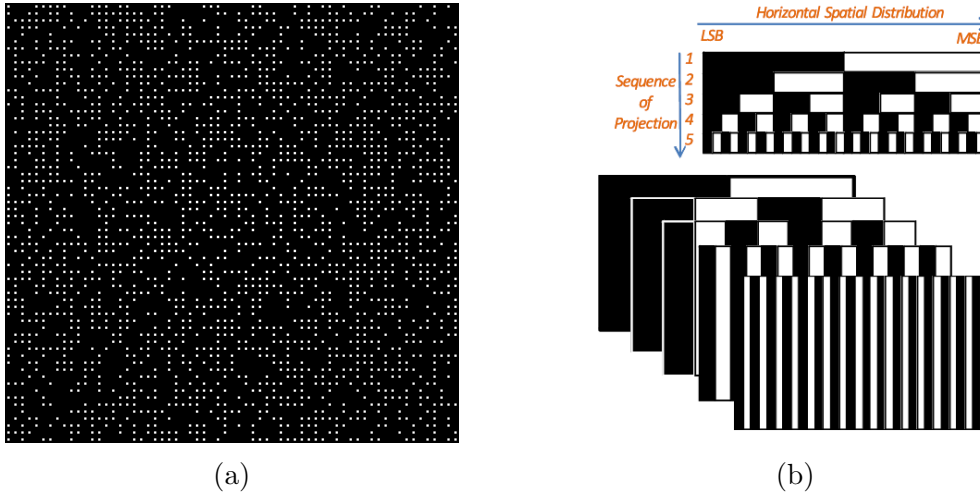


Fig. 2.5: Example of coded patterns. On the left a spatially modulated pattern based on a De Bruijn pattern [21]. On the right an example of a sequential binary-coded pattern [22].

In case of temporal modulation, the codewords are given by a sequence of illumination intensities projected on the same point. The use of a sequential binary-code [22] is a possible implementation of this concept. In this approach, illumination patterns, characterized by vertical white and black stripes, are projected on the scene. Fig. 2.5 (b) shows a sequence of this typology of patterns. The cameras associate the value 0 to the dark regions and the value 1 to the white ones. Concatenating the sequence of recorded 0-1 values, it is possible to label, with a well distinguishable binary number, each scene point belonging to a certain row on the image captured by the cameras. However, the discrete nature of these patterns reduces the depth resolution. Phase shifting methods, based on sinusoidal patterns, exploit higher depth resolution [22]. The phase shifted approach will be described more in detail in Chapter 3, since it will be used in the ground truth acquisition setup we built for our ToF datasets. Temporal modulated patterns are more robust than spatially modulated patterns on depth discontinuities. However, they assume that the scene is static during the sequence projection. If this assumption is violated, the codeword recovery and consequently the stereo matching fails.

In general, active triangulation systems are very accurate and solve the issues of passive systems related to the scene texture dependency. The same discussion

about the relationship between noise and depth, made in the previous section, still holds since it is related to triangulation nature of such devices. Consequently, the performance worsen when the depth going to be estimated increases.

2.3 Time-of-Flight Cameras

A completely different depth estimation approach is used by Time-of-Flight (ToF) depth sensors. They are equipped with an infra-red light projector which illuminates the scene with a predefined temporal modulated signal. The idea is to estimate the time required by the light to go from the projector to the scene and then come back on the camera. Since the speed of light in the air is approximately constant, it is possible to estimate the observed scene point depth from the light round trip time. Two different families of ToF devices exist, these can be classified as *direct* or *indirect* ToF cameras. The direct ToF cameras estimate the arrival time of each photon received by the sensor. Single-photon avalanche diodes (SPAD) ToF cameras belong to this family [24]. The sensors belonging to the second family indirectly estimate the light round trip time by modulating the projected light with a given signal. The recorded light is correlated with an internal reference signal, in order to estimate the phase displacement of the two signals. This type of ToF sensors are also known as continuous wave ToF (CW-ToF) sensors. The following of this chapter will focus on this typology of devices.

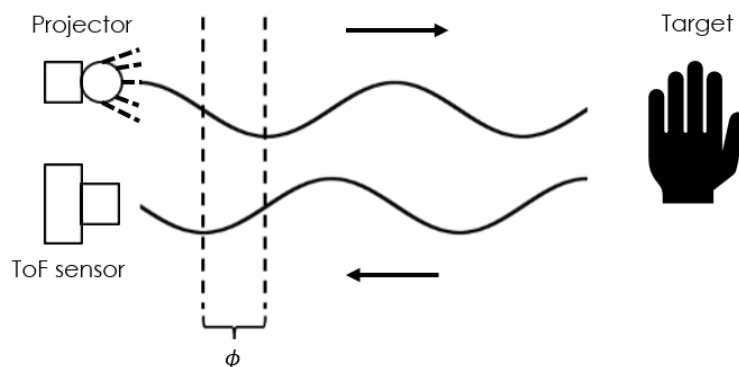


Fig. 2.6: Scheme representing the CW-ToF depth acquisition principle.

2.3.1 Continuous Wave ToF

This section presents the mathematical analysis of the depth estimation of a generic CW-ToF camera employing a sinusoidal wave as light modulation function and a rectangular wave as internal reference signal. This is the most widespread choice among commercial ToF devices. Recall that CW-ToF cameras using rectangular waves as light modulation function exist, but the mathematics used to estimate the depth is slightly different [10].

CW-ToF cameras use an infra-red projector to illuminate the scene with a periodic amplitude modulated light signal, e.g., a sinusoidal wave, and evaluate the depth from the phase displacement between the transmitted and received signal. This process is represented in Fig. 2.6. The projected light signal can be formulated as

$$s_t(t) = \frac{1}{2}a_t(1 + \sin(\omega_r t)) \quad (2.3)$$

where t is the time, ω_r is the signal angular frequency equal to $\omega_r = 2\pi f_{mod}$ and a_t is the maximum power emitted by the projector. The temporal modulation frequency f_{mod} is in nowadays sensors in the range $[10MHz; 200MHz]$. The received light signal can be modeled as:

$$s_r(t) = b_r + \frac{1}{2}a_r(1 + \sin(\omega_r t - \phi)) \quad (2.4)$$

where b_r is the light offset due to the ambient light, $a_r = \alpha a_t$ with α equal to the channel attenuation and ϕ is the phase displacement between the transmitted and received signal. The task of CW-ToF cameras is to compute ϕ since the scene depth d can be computed from ϕ through the well known equation

$$d = \frac{c_l \phi}{4\pi f_{mod}} \quad (2.5)$$

where c_l is the speed of light. Please note that the maximum depth that a ToF camera can capture is related to f_{mod} . Indeed, $\phi \in [0; 2\pi)$ and so $d \in [0; \frac{c_l}{2f_{mod}})$. For this reason, $\frac{c_l}{2f_{mod}}$ is the *unambiguous range* of the ToF acquisition.

The electronics inside ToF pixels are able to compute the correlation function between the received signal and a reference one, e.g., a rectangular wave at the same

modulation frequency $rect_{\omega_r}(t) = H(\sin(\omega_r t))$, where $H(\cdot)$ represents the Heaviside function. The correlation function sampled in $\omega_r \tau_i \in [0; 2\pi)$ can be modeled as

$$\begin{aligned} c(\omega_r \tau_i) &= \int_0^{\frac{1}{f_{mod}}} s_r(t) rect_{\omega_r}(t + \tau_i) dt = \\ &= \frac{1}{f_{mod}} \left[\frac{b_r}{2} + \frac{a_r}{4} + \frac{a_r}{2\pi} \cos(\omega_r \tau_i + \phi) \right]. \end{aligned} \quad (2.6)$$

$c(\omega_r \tau_i)$ represents a measure of the number of photons accumulated by a pixel during half the integration time. By defining $B = \frac{1}{f_{mod}} \left(\frac{b_r}{2} + \frac{a_r}{4} \right)$, collecting the additive constant term in the received light, and $A = \frac{a_r}{2\pi f_{mod}}$ the amplitude of the received sinusoidal light signal, that is proportional to the power of the direct component of the received light, Eq. 2.6 can be reformulated as

$$c(\omega_r \tau_i) = B + A \cos(\omega_r \tau_i + \phi). \quad (2.7)$$

Nowadays ToF cameras usually acquire 4 samples of the correlation function $c(\omega_r \tau_i)$ at $\omega_r \tau_i \in \{0; \frac{\pi}{2}; \pi; \frac{3\pi}{2}\}$ and for this reason we will consider this case in the next of this chapter. In this setting, we have:

$$\phi = \text{atan2} \left(c\left(\frac{3\pi}{2}\right) - c\left(\frac{\pi}{2}\right), c(0) - c(\pi) \right). \quad (2.8)$$

where $\text{atan2}(y, x)$ is the function returning the phase of the complex number $x + iy$.

Finally, it is possible to use Eq. 2.5 to estimate the depth d from ϕ . The depth estimation with CW-ToF cameras is computationally simple, when compared with stereo matching, and it is possible to record depth at video frame rates with an accuracy in the range of centimetres or even millimetres in favourable conditions. Moreover, the CW-ToF devices can be miniaturised and implemented in portable devices.

In the next chapters of this thesis, we will use the term ToF to refer to CW-ToF in order to simplify the notation.

2.3.2 Error Sources in ToF Recordings

The previous section introduced the basic concepts behind CW-ToF depth acquisitions in ideal conditions. Here, some of the main issues related to their depth acquisitions will be listed and analysed.

Photon Shot Noise

ToF cameras are active depth devices and the depth estimation accuracy is strictly related to the strength of the reflected modulated light. In case a high amount of light is reflected, the depth estimation is very accurate, otherwise the accuracy decreases. This is due to the random nature of the light. The amount of received photons can be modelled as a *Poisson random variable* whose mean (μ_l) and variance (σ_l^2) are equal to the number of received photons. σ_l^2 is the variance of the so called *photon shot noise*. The *signal to noise ratio* ($SNR = \frac{\mu_l}{\sigma_l}$) of the number of received photons increases when this number increases.

By applying an error propagation analysis (introduced in Appendix A) on the depth acquisition with Eq. 2.5 and by assuming the above mentioned *Poisson* nature of the light, it comes out that the variance of the noise due to photon shot noise on the depth evaluation is

$$\sigma_{ps}^2 = \left(\frac{c_l}{4\pi f_{mod}} \right)^2 \frac{B}{2A^2} \quad (2.9)$$

where A and B are respectively the amplitude of the received modulated light and the light offset as defined in Eq. 2.7. A and B are assumed to be measured in number of received photons. The derivation of this result can be found in Appendix B.

Thermal Noise

Apart from photon shot noise, also the thermal excitation of the electron produced by the photo-diode composing the ToF pixels influence the final depth estimation. The depth perturbation due to this phenomenon can be modelled as Gaussian random variable with zero mean and variance σ_t^2 . This variance is related to the type of camera used and increases when the temperature increases.

The zero-mean noise variances σ_t^2 and σ_{ps}^2 can be summed up together for a more general evaluation of the depth distortion.

Multi-path Interference

The ToF depth estimation is correct if the light received by the sensor is reflected only once inside the scene, the direct component of the light labelled as 1 in Fig. 2.7 (a). However, in real scenarios a part of the light emitted and received by the ToF system can also experience multiple reflections, this part of the received light is named global component of the light. The global component can be caused by multiple phenomena and some of them are depicted in Fig. 2.7 (a) as diffuse reflection (ray 2), specular reflection (ray 3), and sub-surface scattering (ray 4). Each of these reflections carries a sinusoidal signal with a different phase offset proportional to the length of the path followed by the light ray.

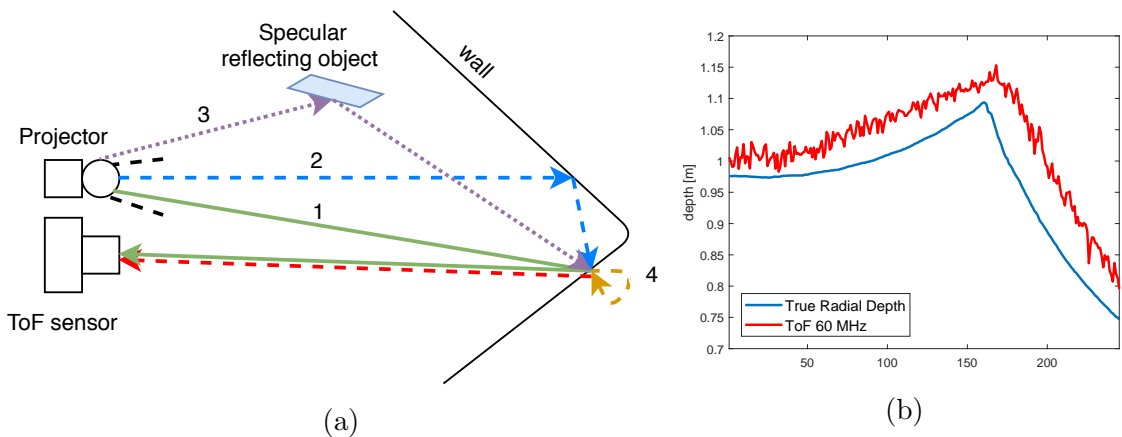


Fig. 2.7: Example of multi-path interference on a corner. On the left, representation of different types of reflections on a corner. On the right, comparison between the true radial depth of a corner and the ToF estimation on a cross-section of a corner. It is possible to appreciate the depth over estimation due to MPI.

In this scenario, the correlation function presented in Eq. 2.6 can be modelled as

$$\begin{aligned}
c(\omega_r \tau_i) &= \frac{1}{f_{mod}} \left[\frac{b_r}{2} + \frac{a_r}{4} + \frac{a_r}{2\pi} \cos(\omega_r \tau_i + \phi_d) + \frac{b_{r,g}}{2} + \frac{a_{r,g}}{\pi} \cos(\omega_r \tau_i + \phi_g) \right] \\
&= B_{FF} + A_{FF} \cos(\omega_r \tau_i + \phi_{FF})
\end{aligned} \tag{2.10}$$

where the first sinusoidal term is related to the direct component of the light and the second to the global one. $a_{r,g}$ and $b_{r,g}$ are respectively proportional to the amplitude and intensity of the global light waveform due to MPI. The superimposition of the direct and global components originates the *multi-path interference* (MPI) phenomenon. In the next of this thesis, the signal resulting from the superimposition of the direct and global components will be named *full field* component. By sampling the ToF correlation function, only the *full field* phase ϕ_{FF} can be recovered. It is a corrupted version of corrected phase ϕ_d . The resulting ToF depth acquisition is generally overestimated due to the mixing of direct and global component of the light. Fig. 2.7 (b) depicts the overestimation of the depth due to MPI in a corner scene. Please note the gap between the true radial depth (blue line) and the ToF estimation (red line) due to the diffuse reflections on the side of the corner.

Another important aspect of MPI phenomenon is its *frequency diversity*. Since the phase displacement of the received sine waves ϕ is equal to $\frac{2\pi f_{mod} d}{c}$, where d is the distance travelled by the considered wave, the interference between the direct and the global components of the light (modelled by the first row of Eq. 2.10) is dependent by the modulation frequency f_{mod} . Thus, ϕ_{FF} and by consequence also the final depth estimation change by changing f_{mod} . This aspect is used by some MPI correction methods which try to estimate the corruption due to this phenomenon by sensing the scene using different modulation frequencies, as also discussed in Chapter 4.

Mixed Pixels

Each pixel of the ToF camera observes a patch of the scene and not an ideal point. When the observed patch contains a depth discontinuity, the ToF estimation is a linear combination of the different depths contained in it. This is function of the reflectivity and the depth of the different observed points.

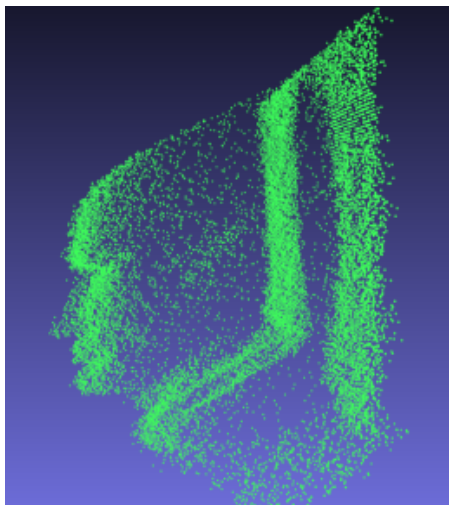


Fig. 2.8: Example of mixed pixels. The face of the foreground fades with the wall on the background on the borders.

Fig. 2.8 shows an example of mixed pixels on the borders of a face in front of a wall. Please note how the points near to the depth discontinuities are in between the two depths, but no real point is exactly at those positions.

The mixed pixels effect is enhanced by the small spatial resolution of nowadays ToF cameras, since each camera pixel is associated to a wide region of the scene. Currently, the most employed commercial ToF sensors have about VGA resolution. However, in the near future the spatial resolution of this sensors will increase, and for example the newest Microsoft kinect Azure [25] has megapixel resolution. The reduced spatial resolution, compared to standard color cameras, is due to the complexity of ToF pixels. In the future, it will be possible to reduce the effect of mixed pixels by increasing the spatial resolution of the ToF cameras.

Harmonic Distortion

The theory for depth estimation with ToF cameras introduced in Section 2.3.1 is valid if the transmitted sinusoidal light signal and the rectangular reference signal are ideal. However, this is not the case on real ToF cameras. Usually, the transmitted signal is something in between a rectangular wave and a sinusoidal one, so neither purely sinusoidal nor rectangular. This introduces a systematic distortion in the depth estimation, using Eq. 2.5, due to additional spurious sinusoidal

components in the emitted wave.

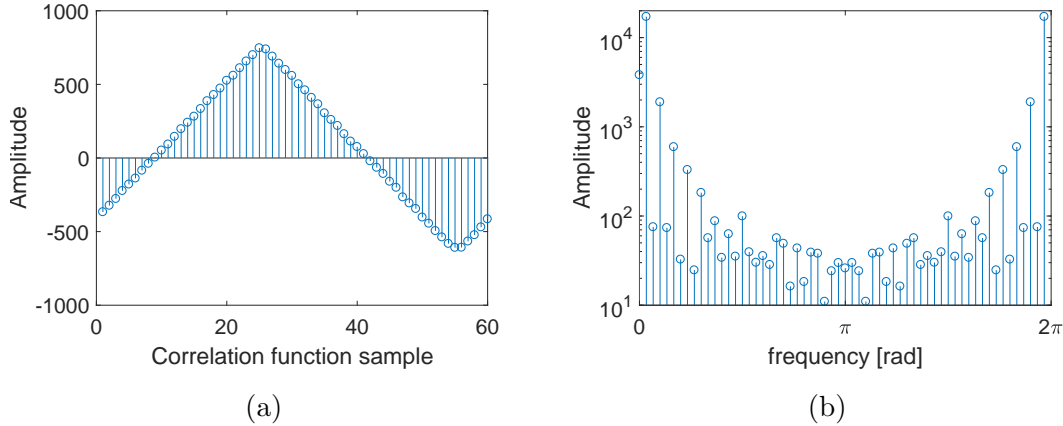


Fig. 2.9: Correlation function on a real ToF camera. On the left, its representation in the temporal domain. On the right, its representation in the frequency domain. It is more similar to a triangular function than a sinusoidal one.

In this scenario, the resulting correlation function appears to be something in between a sinusoidal and a triangular wave (correlation between two rectangular waves). Fig 2.9 shows an example of ToF correlation function obtained using a *Sony* ToF camera [26] with modulation frequency set to 10 MHz. Here, 60 samples of the correlation function are used, but in standard acquisitions usually only 4 are captured for frame rate constraints. Sampling the correlation function on only 4 points causes aliasing and it is impossible to disambiguate between the fundamental sinusoidal component (the one useful for the depth estimation) and the spurious ones.

Fig. 2.10 shows how this non ideality of the employed signals affects the ToF depth (a) and amplitude (b) retrieval when the standard ToF sinusoidal model of Eq. 2.7 is used. This is the so called *harmonic distortion*. The distortion is periodic and in case of depth estimation the error can reach a value of 30 cm. In Fig. 2.10 (b) the effect on the amplitude estimation is depicted showing the ratio between the actual recorded amplitude and the amplitude in case of absence of aliasing.

The harmonic distortion is an issue with a strong impact on the ToF depth acquisitions and different methods for its correction have been proposed in literature. The proposal of Lindner et al. [27] is based on a *look-up table* strategy and it is one

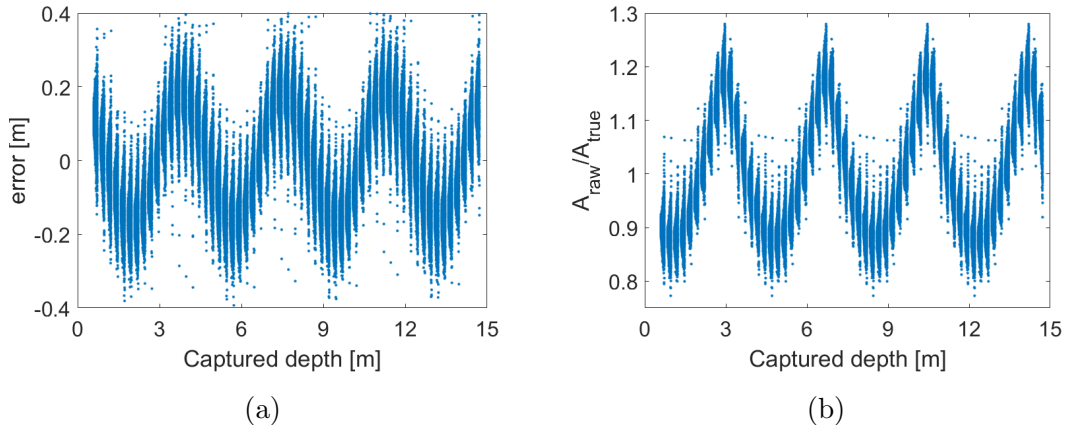


Fig. 2.10: Effect of the harmonic distortion on the depth (a) and amplitude (b) estimation in a ToF acquisition by assuming an ideal system using 4 samples of the correlation function (Eq. 2.7).

of the most employed.

The correction of the amplitude distortion is not common since it is not directly used in the depth estimation process. However, as it will be shown in Chapter 4 the correction of the harmonic distortion on ToF amplitude has an important role in the correction of MPI. For this reason, we adapted the approach presented in [27] for the additional calibration of ToF amplitude.

Pixels Non-Uniformity

It is worth mentioning that the photo response of each pixel of the camera is not uniform, due to small differences in the doping of the silica originated in the production of the sensors. This causes a slightly different number of collected electrons in the different pixels given the same number of received photons. The different response of each pixels can be mitigated by implementing a pixel level calibration step in the harmonic undistortion method introduced in [27].

Chapter 3

Stereo and ToF Dataset Collection

This chapter introduces the depth datasets, collecting stereo vision system and ToF camera recordings, used during my Ph.D. work. These datasets have been employed to evaluate the performance of raw depth acquisitions and to test the real effectiveness of the newly introduced methods for ToF data denoising and ToF-stereo data fusion. In particular, some datasets as the LTTM5 [28] and REAL3 [6] datasets are taken from the literature, instead I have collected other datasets ad hoc for my work. Most of previous publicly available depth datasets, provided with depth ground truth, contained depth recordings from stereo vision systems only, without ToF sensors acquisitions as in [19, 29]. Some recent works have introduced ToF datasets with the related depth ground truth as in [30–32], but they contain just simulated data. This is due to the fact that the recording of depth ground truth from the ToF camera viewpoint involves complex and time consuming procedures. The usage of simulators can simplify the ToF data collection task, giving an initial evaluation of the denoising methods. However, not all the acquisition phenomena can be faithfully simulated and various differences between simulated and real data can be encountered as it was shown in our paper [2] and as it will be discussed in Chapter 4.

3.1 Introduction to the Depth Datasets

This chapter introduces the depth datasets which have been used to train and evaluate the depth refinement methods described in the next of this thesis. Table 3.1 collects some of the characteristics of the datasets, in order to make it simpler to compare and distinguish them. In the next of this chapter: first, the ToF-stereo datasets LTTM5 [28] and REAL3 [6] will be introduced; then, the novel datasets, collected ad hoc for my Ph.D. work, will be described together with the procedures used for their collection.

Dataset	Type	Devices	GT	# scenes	Used for
LTTM5	Real	2 Basler + MESA SR4000	Yes	5	Testing
REAL3	Real	ZED + Kinect v2	Yes	8	Testing
SYNTH3 _{train}	Synth	ZED + Kinect v2	Yes	40	Training
SYNTH3 _{test}	Synth	ZED + Kinect v2	Yes	15	Testing
$S_{1,train}$	Synth	DS541	Yes	40	Training
$S_{1,test}$	Synth	DS541	Yes	14	Testing
S_2	Real	DS541	No	97	Training
S_3	Real	DS541	Yes	8	Validation
S_4	Real	DS541	Yes	8	Testing
S_5	Real	DS541	Yes	8	Testing

Table 3.1: Collection of some key characteristics of the depth datasets introduced in this chapter.

3.2 Real Datasets

Few real datasets containing calibrated depth data collected jointly with a stereo vision system and a ToF camera exist and the LTTM5 and the REAL3 are among these.

LTTM5 dataset was introduced in [28] and it collects data acquired by two Basler scA1000 RGB cameras and a MESA SR4000 ToF camera. The acquisition set-up is depicted in Fig. 3.1. The ToF camera is placed in the middle of the two color cameras. These have a baseline of 17 cm and they acquire two 1032×778 pxl RGB images. The ToF camera can acquire 176×144 pxl depth and amplitude images in the range $[0; 5]$ m using 30 MHz as modulation frequency.

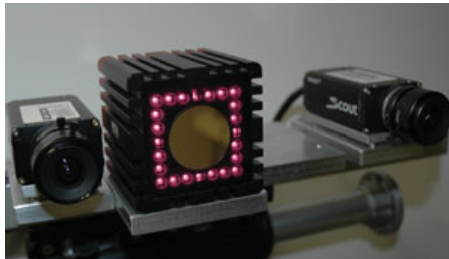


Fig. 3.1: Stereo-ToF acquisition system used to collect the LTTM5 dataset.

The ground truth has been estimated by using an external light projector together with the stereo system. 600 different patterns have been emitted on the target scene, following the approach presented in [33]. The patterns have been used to label each scene point with a unique code, obtained by combining all the projected patterns observed from the color cameras. The disparity maps have been computed with a block matching algorithm, looking for the same code word in the 2 images. Finally, a subpixel refinement and a left-right check were also applied. The accuracy of the obtained depth ground truth is of about 2 mm. The 5 static scenes contained in the LTTM5 dataset are depicted in Fig. 3.2. These have been captured in a lab environment.

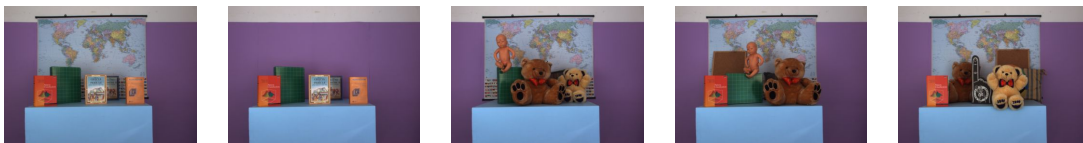


Fig. 3.2: Color views of the scenes contained in the LTTM5 dataset.

REAL3 dataset was introduced in [6], a paper I wrote in collaboration with other members of the LTTM laboratory at University of Padova. Specifically, this dataset was collected by Giulio Marin. It is a real world dataset acquired with a Microsoft Kinect v2 ToF camera and a ZED stereo vision system from Stereolabs [8]. The ZED is based on a passive stereo technology and it is equipped with two 4 MP cameras that provide images up to 2208×1242 pxl at 15 fps. The sensor is able to provide images up to 100 fps at a lower resolution. The baseline is of 12 cm and the diagonal field of view is 110° . Kinect v2 is one of the best and most diffuse consumer ToF depth cameras. Compared to other ToF cameras it provides a cleaner

and denser depth map. The Kinect v2 is able to acquire a 512×424 pxl depth and amplitude map at 30 fps with a depth estimation error typically smaller than 1% of the measured distances and a diagonal field of view of 92° .



Fig. 3.3: Representation of the real Stereo-ToF acquisition system used for the REAL3 dataset. The figure shows the relative position of the ZED camera and of the Kinect v2.

The algorithm developed to compute the ground truth map uses the stereo cameras to match corresponding pixels and estimate the disparity between them. A line laser, with a regular red beamer visible to humans, is used to label the conjugate points in the two views. The goal is to “paint” the scene with the line laser and for each acquisition match corresponding lit points in the two images. Ideally, only one point for each row of the image for each acquisition is lit. Due to noise in the images, the estimated disparity is updated for a given pixel every time there is a new measurement and by accumulating all the values. The median value is kept as ground truth. A servomotor is employed to control the laser movement making the system fully automatic.

The dataset contains 8 scenes, all including static scenarios in an indoor environment. The scenes have different complexity, ranging from flat surfaces to more complex shapes like the leaves of a plant. These contain objects with and without texture to check the behaviour of the algorithms with disparate conditions. Materials with challenging reflection properties compose the scenes, including reflective and glossy surfaces as well as rough material that usually cause problems to active cameras. Fig. 3.3 shows the relative position of the two sensors. The color view of the 8 scenes contained in REAL3 are depicted in Fig. 3.4.

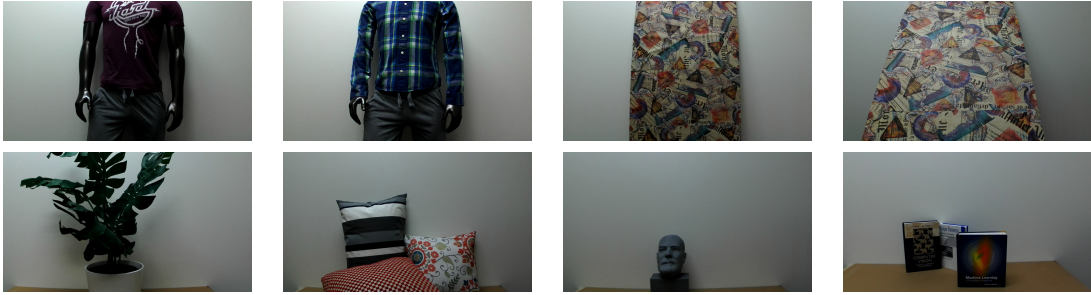


Fig. 3.4: Color views of the scenes contained in the REAL3 dataset.

3.2.1 Proposed Multi-Frequency ToF Datasets

Both LTTM5 and REAL3 datasets contain ToF data, but no raw multi-frequency acquisitions are contained. Indeed the ToF depth and amplitude images are acquired at 30 MHz for LTTM5. In the REAL3 dataset, just the output of a proprietary algorithm that combines the ToF data acquired by the kinect v2 at 16, 80 and 120 MHz is available. These two datasets can not be used to test ToF depth refinement methods exploiting multi-frequency ToF data, as the SRA [34] MPI correction method. For this reason, we have set-up an acquisition system composed by a Sony DS541 ToF camera and two Basler acA2500-14gm RGB cameras together with a light projector, going to illuminate the scene with a known sequence of patterns. Fig. 3.5 depicts the employed trinocular system.

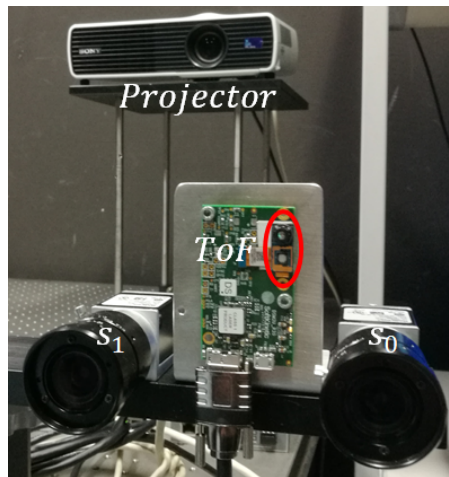


Fig. 3.5: Trinocular system used for the acquisitions of the multi-frequency ToF datasets. The standard cameras are placed on the sides with a baseline of 10 cm.

The three cameras are mounted on a tripod and they have the optical axis approximately parallel to each other. The ToF camera is placed in the middle of the two standard cameras, which form a stereo vision system with a baseline equal to 10 cm. They are labelled as s_0 and s_1 in Fig. 3.5. s_0 is used as reference camera of the stereo vision system. The external light projector is placed in a position suitable to illuminate the part of the scene observed by the cameras. This will compose an active stereo system with s_0 and s_1 , that will be used to estimate the depth ground truth.

Table 3.2 collects some hardware parameters of the employed cameras. The two typologies of cameras share similar horizontal and vertical Field-of-View (FoV), but the one related to the standard cameras (Basler) is slightly smaller.

	Basler	ToF camera
Resolution	2592×1944	320×239
Horizontal FOV	55°	60°
Vertical FOV	40°	45°
Focal length	6 mm	2.8 mm
Pixel size	$2.2 \mu\text{m}$	$10 \mu\text{m}$

Table 3.2: Hardware parameters of the Basler gray-scale camera and DS541 ToF camera.

System Calibration

Before starting to record data with the composed trinocular system, the system has been calibrated. The calibration involved the geometric calibration of the three cameras and the harmonic undistortion of the ToF camera. Regarding the geometric calibration of the system, the intrinsic and extrinsic parameters of the cameras have been estimated. We exploited the technique proposed by Zhang in [35] using a dot pattern as calibration pattern. Fig. 3.6 shows the acquisition of the dot pattern from the trinocular system, where the amplitude image is used for the calibration of the ToF camera. We selected to use the dot pattern since we evaluated that it is more robust for calibration when a low resolution ToF camera is involved in the process. The low resolution affects less the localization of a dot center with respect to a feature in a standard checkerboard pattern, that is usually employed

for geometric calibration.



Fig. 3.6: Acquisition of the dot geometrical calibration pattern from s_0 (left), s_1 (center) and the amplitude image of the ToF camera (right).

In a second step, the harmonic distortion related to the ToF camera recording has been corrected. As discussed in Section 2.3.2, this issue affects both the depth and the amplitude recordings. For the depth undistortion we used a method based on the approach proposed by Lindner et al. [27]. A flat wall is placed orthogonal to the optical axis of the ToF camera. Then, the camera records the wall by changing the initial phase shift of the reference signal. In this way, a virtual shift of the wall is emulated and by knowing the original position of the wall and comparing it with the raw ToF outcome is possible to set-up a *look-up-table* to correct this distortion. A similar approach is implemented for the amplitude undistortion. Since the virtual displacement of the wall corresponds to an oversampling of the ToF correlation, it was possible to estimate the correct amplitude by investigating with a Fourier analysis the amplitude of the fundamental harmonic. This calibration procedure is repeated for each ToF modulation frequencies used in the dataset.

Depth Ground Truth Acquisition

In this section, it is described how we used the active stereo system, composed by the two Basler cameras and the light projector, to estimate a depth ground truth of the scenes. In a second phase, this accurate depth map is projected on the ToF sensor.

We used the external light projector to illuminate the scene with a known sequence of patterns. The projected patterns have to uniquely label each scene point when looked on horizontal lines on the pixel grids of the stereo cameras. We used

a phase shifting approach to generate the patterns, as introduced in Section 2.2. A sequence of vertical sinusoidal patterns are emitted by the light projector on the scenes. Examples of the employed patterns can be found in Fig. 3.7. The intensity of the light emitted by the projector for the pixel in position $(x; y)$ is described by the equation:

$$L_i(x; y) = \frac{1}{2} \left(1 + \cos\left(\phi_i + \frac{2\pi x}{h_{res}} f\right) \right) \quad (3.1)$$

where h_{res} is the horizontal resolution of the projector, f is the frequency (number of sinusoidal periods per image) of the projected pattern and ϕ_i is the initial phase. In our acquisitions we used 5 pattern frequencies, $f \in \{1; 2; 4; 8; 16\}$ [$\#periods/image$], and $N = 16$ phase displacements of the sinusoidal patterns with initial phases $\phi_i = \frac{2\pi i}{N}$, where $i = 0; 1; \dots; N - 1$. So, we projected 80 patterns for each depth ground truth acquisition. Please note that also the phase shifting approach for active stereo is sensible to diffuse reflections as mentioned by Gupta et al. in [36], this is in some way related to *multi-path interference* in ToF devices. This issue can be mitigated by using high frequency sinusoidal patterns [36]. From our investigation, it came out that using the maximum frequency $f = 16$ was enough to have a ground truth accurate enough for our ToF dataset.

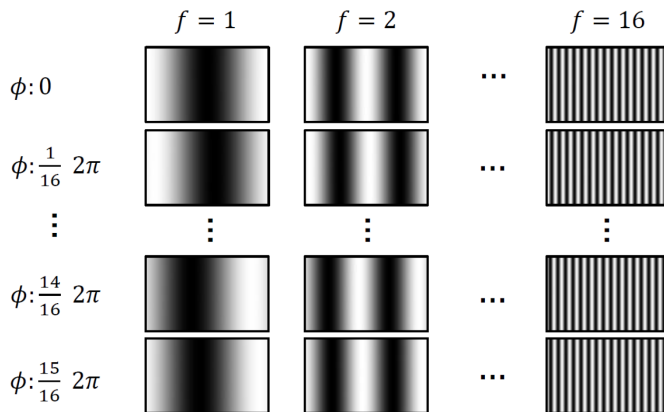


Fig. 3.7: Example of projected sinusoidal patterns for different spatial frequencies f and initial phase ϕ .

Each time a pattern is projected, the two standard cameras, s_0 and s_1 , record a grayscale image of the scene. This brings to have a stereo acquisition of each

pattern. For each pixel $(i; j)$ of the two stereo views we estimate the observed pattern offset $\theta_{(i;j);f} = \frac{2\pi x}{h_{res}} f$. The pattern phase offset for each frequency f can be estimated by modelling a sine wave, e.g., with Fourier analysis, among the acquisitions with different initial phases. Once the complete pattern phase offsets have been computed for each stereo view and for the different pattern frequencies, we are going to phase unwrap the pattern phase offsets at the highest frequency, by using the lower frequencies, bringing the phase from the range $[0; 2\pi)$ to $[0; 16 \cdot 2\pi)$. The couple of unwrapped phase images are rectified by using the intrinsic and extrinsic parameters estimated with the geometrical calibration of the trinocular system. This allows us to obtain two coded images, one for each stereo view, which have a unique codeword (related to a scene point) in each horizontal line in the sensors.

We ran a simple matching block stereo algorithm with a Sum of Absolute Difference (SAD) metric on the two coded images in order to compute an accurate disparity map of the scene from the s_0 viewpoint. The computed disparity map is used to estimate the scene depth map using the calibration parameters of the cameras.

The computed depth map is projected on the ToF sensor by using intrinsic and extrinsic camera parameters. Since the ToF camera has a spatial resolution that is much smaller than the standard cameras (see Table 3.2), it results that usually a ToF pixel corresponds to multiple pixels, with the related depth information, on s_0 . The smallest depth value is selected in order to avoid problems related to occlusion, when projecting the points from the stereo reference camera to the ToF sensor. This operation brought the active stereo depth on the ToF perspective and we used this as depth ground truth for the ToF sensor. Fig. 3.8 contains an example with the ToF depth acquisition and the acquired depth ground truth. The ToF error map is computed as ToF depth map minus the ground truth.

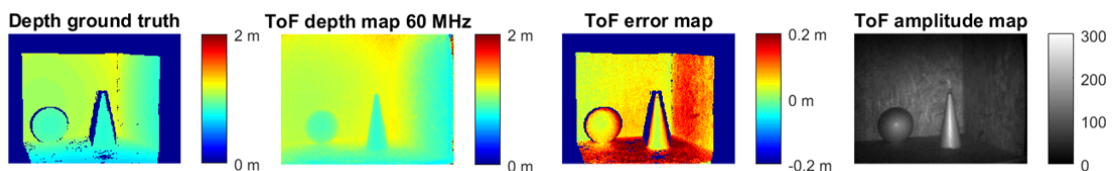


Fig. 3.8: ToF data and related depth ground truth on a sample scene. Please note the depth overestimation (ToF error map) due to MPI.

We tested the accuracy of the ground truth estimation on a corner scene as the one depicted in Fig. 3.9. We compared the estimated ground truth model with a synthetic model created in Blender. The estimated and the synthetic meshes have been first aligned and then compared with the *CC software* [37]. It comes out that the estimated corner geometry has a MAE (mean absolute error) of 0.9 mm when compared with its synthetic model, about 100 times more accurate than the ToF acquisition corrupted by MPI.

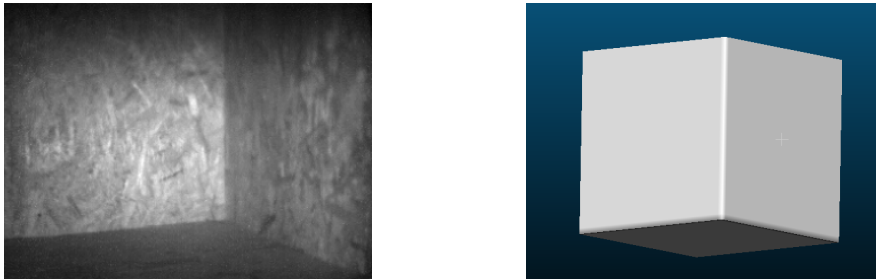


Fig. 3.9: On the left, the ToF amplitude image of the corner used for the evaluation of the ground truth accuracy. The mesh of the ideal 3D model is on the right.

Description of the Datasets

The system introduced in the previous section has been used to collect four datasets. Following the nomenclature used in [3], these are respectively named S_2 , S_3 , S_4 and S_5 . The data contained in them will be described in the next of this section.

These datasets contain ToF data acquired using six different modulation frequencies, from 10 MHz to 60 MHz with steps of 10 MHz, but for our work we used just the data acquired at 20, 50 and 60 MHz. The depth data have been phase unwrapped by using these multi-frequency information, in order to have the maximum unambiguous range equal to 15 m. Note that all the datasets contain structures originating MPI. In the following of this section we are going to explain the specific characteristics of each of the four datasets.

S₂ The unlabeled real dataset S_2 is composed by scenes captured in a office environment in uncontrolled light conditions (ambient light was present). The acquisitions frame static scenes containing tables, chairs, lockers and many other different objects that can be found in a office. The dataset contains 97 recorded

scenes, and for each of them the calibrated depth and amplitude images have been stored. The depth values are in the range from 0.5 to about 10 m. For this dataset no ground truth has been acquired and it has been used for unsupervised training in our works. Fig. 3.10 shows some examples of depth and amplitude images contained in S_2 .

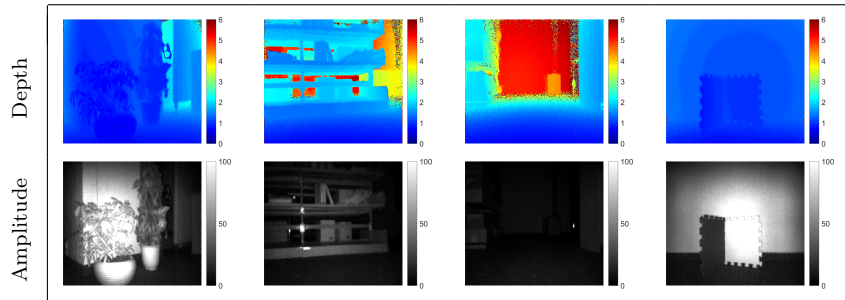


Fig. 3.10: Representation of some of the ToF recordings contained in the S_2 dataset. Here we show the depth in meters and the amplitude images captured at 60 MHz.

S₃ The subjects of the recordings in the real dataset S_3 are static scenes containing puppets, small boxes, wooden corners and polystyrene cones and spheres. The recorded depth images are in the range between 0.5 and 2 m. The depth ground truth of the ToF acquisitions has been generated with the active stereo system registered with the ToF camera. This dataset contains 8 scenes and it has been used as validation dataset for the proposed deep learning methods. Fig. 3.11 shows some examples of depth and amplitude images contained in S_3 .

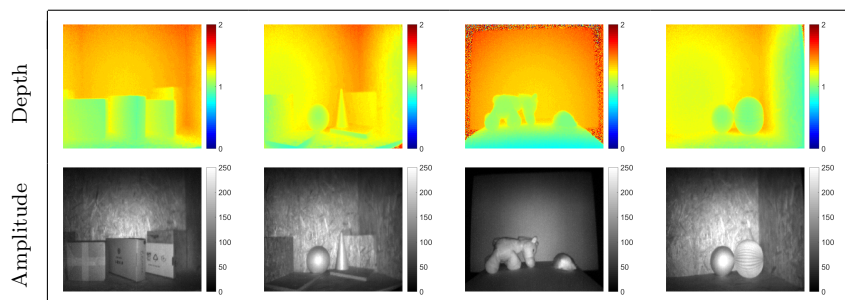


Fig. 3.11: Representation of some of the ToF recordings contained in the S_3 dataset. Here we show the depth in meters and the amplitude images captured at 60 MHz.

S₄ This dataset contains 8 real world scenes whose subjects are wooden corners and object of different materials, as plastic and ceramic, placed in a wooden box,

where a lot of MPI is present. The ToF recordings are provided with depth ground truth. This dataset is used for testing the proposed denoising methods. Fig. 3.12 shows some examples of depth and amplitude images contained in the dataset S_4 .

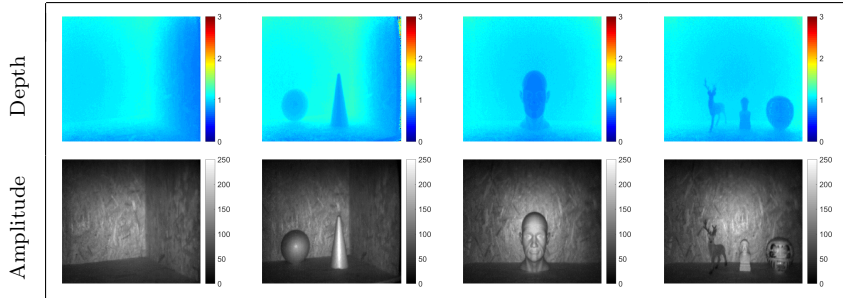


Fig. 3.12: Representation of some of the ToF recordings contained in the S_4 dataset. Here we show the depth in meters and the amplitude images captured at 60 MHz.

S_5 The subjects of the recordings from the real dataset S_5 are 8 static scenes containing boxes of various shapes and dimensions. We decided to create this *box* dataset since ToF sensors can be used in logistics and manufacturing for inspection, handling and dimensioning of box-shaped objects and we would like to evaluate which are the performance of our methods in this scenario. The dataset also contains the ground truth depth maps related to the ToF acquisitions. This dataset is used for testing the proposed denoising methods. Fig. 3.13 shows some of the depth and amplitude images contained in S_5 .

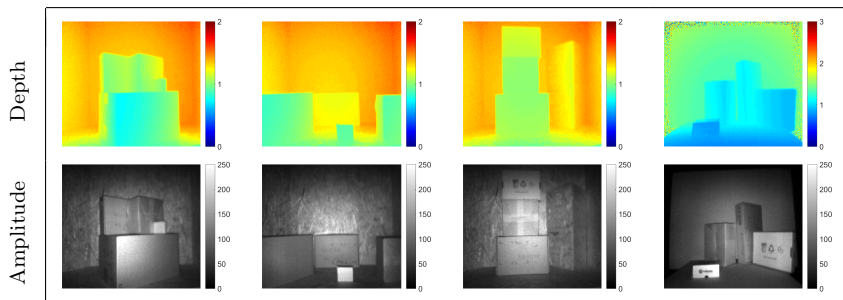


Fig. 3.13: Representation of some of the ToF recordings contained in the S_5 dataset. Here we show the depth in meters and the amplitude images captured at 60 MHz.

3.3 Synthetic Datasets

All the datasets introduced in the previous sections are composed by a small number of scenes, which are not sufficient for a reliable training of complex neural networks. The collection of bigger and more various datasets is impracticable in a lab environment, but a possible compromise is to create a synthetic dataset, where a simulator can emulate the recordings of standard color and ToF cameras on pre-built 3D models of scenes. Here, we introduce the synthetic datasets S_1 and SYNTH3 we developed to train the proposed fusion and denoising methods based on deep learning techniques.

3.3.1 S_1 : The Synthetic DS541 Dataset

This section introduces the synthetic dataset collecting the acquisitions from a simulated ToF camera. The dataset is named S_1 and was originally introduced in our paper [2] trying to mimic the acquisitions of the Sony DS541 ToF camera used for the recordings of the datasets S_2 , S_3 , S_4 and S_5 . The scenes contained in the dataset are generated using the 3D creation suite *Blender* [38]. These have been downloaded from *Blend Swap* [39], a website collecting the artwork of 3D graphic designers, and they have been appropriately modified and rendered from virtual viewpoints in order to generate the ToF dataset.

The data captured by the ToF camera have been computed by using the *Sony ToF Explorer* simulator developed by Sony Eutec. The *Sony ToF Explorer* simulator is an extended version of the simulator from Heidelberg University [40] that is able to accurately simulate the data acquired by a real ToF camera including different sources of error as shot noise, thermal noise, read-out noise, lens effect, mixed pixels and the interference due to the global illumination (multi-path interference). The ToF simulator takes in input the scene information generated by *Blender* exploiting the rendering engine *LuxRender* [41]. We acquired with the simulator the 320×240 pxl depth and amplitude maps (the resolution and the other simulator parameters have been set in order to emulate the DS541 camera used in the real world setup).

Moreover, the dataset contains also the scene depth ground truth relative to the

point of view of the ToF camera.

The dataset is split in a training and a test set. The training set, $S_{1,train}$, contains 20 unique scenes each rendered from two different viewpoints, leading to a total of 40 scenes split into a training and a validation set. Even if the number of scenes is low if compared with datasets used for the training of deep networks for other tasks, it is still one of the largest datasets for ToF data denoising currently available. Furthermore, the scenes are very different one from the other representing different conditions. The test set, $S_{1,test}$, instead contains 14 unique scenes.

The various scenes from $S_{1,train}$ and $S_{1,test}$ contain walls, furniture and objects of various shapes and color in different environments, e.g., living rooms, kitchen rooms or offices but also outdoor locations with non-regular structures. The depth range is also very different across the various scenes ranging from about 50 cm to 10 m thus providing a large range of measurements.

3.3.2 SYNTH3

Starting from the same 3D models of the scenes used to simulate the S_1 dataset, we created a stereo-ToF dataset we named SYNTH3. This dataset tries to mimic the recordings contained in the real dataset REAL3. It contains ToF depth and amplitude images using the hardware parameters of a Microsoft kinect v2 [10, 42] and the color images acquired by a simulated ZED stereo vision system [8]. The complete virtual system is depicted in Fig. 3.14, while Table 3.3 summarizes the parameters of the cameras.

	Stereo setup	ToF camera
Resolution	1920×1080	512×424
Horizontal FOV	69°	70°
Focal length	3.2 mm	3.66 mm
Pixel size	$2.2 \mu\text{m}$	$10 \mu\text{m}$

Table 3.3: Parameters of the stereo and ToF sensors.

The color images have been generated using the 3D renderer engine *LuxRender* inside *Blender*. The stereo setup is made of two Full-HD (1920×1080) color cameras with a baseline of 12 cm and the optical axes and image planes parallel to each other.

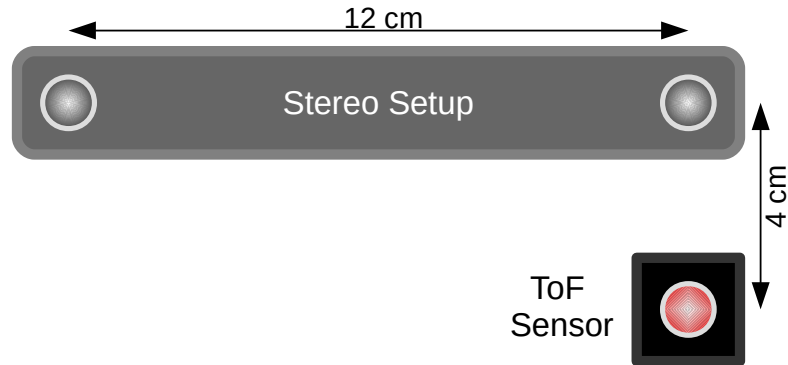


Fig. 3.14: Representation of the synthetic Stereo-ToF acquisition system. The ToF sensor is placed below the color camera.

Since the cameras are ideal and their optical axes are already aligned, there is no need to rectify the two color views, as instead is done for the real world data.

As described in the previous section, the *Sony ToF Explorer* has been used on the synthetic 3D models to estimate the depth and amplitude images acquired by the synthetic Kinect v2 ToF camera.

The image plane and optical axis of the ToF camera are parallel to those of the stereo camera and the ToF viewpoint is placed under the right camera of the stereo system at a distance of 4 cm.

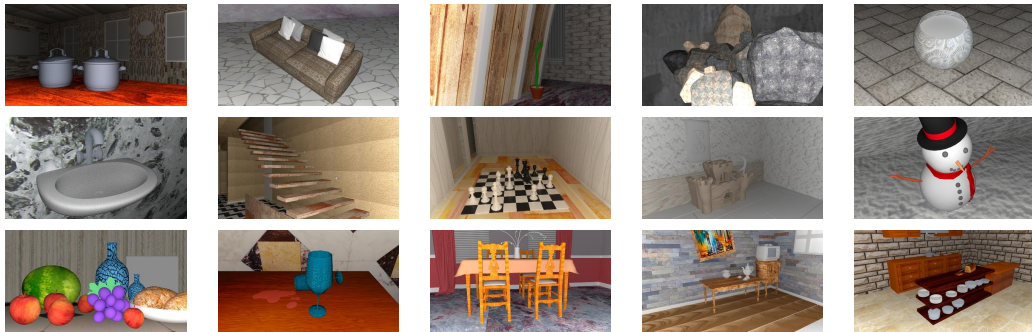


Fig. 3.15: Test set used for the evaluation of the denoising methods. The figure shows the color image from the reference camera of ZED system for each scene in the SYNTH3_{test} dataset.

Also the SYNTH3 dataset is divided in a training set, SYNTH_{train} and a test set, SYNTH_{test} . The same split used for the S_1 dataset has been used. However,

SYNTH_{test} contains a new scene, that contains low reflective objects, with respect to S_1 . Fig. 3.15 shows the color view of the test scenes. In particular, the newly introduced scene is the one in the bottom right. It contains a very low reflective table in the middle of the scene that stresses the depth acquisition of the ToF camera and it is useful to evaluate the capabilities of a stereo-ToF fusion method.

The following of this thesis will describe the various methods for depth data denoising introduced during my three year Ph.D. program and all the datasets presented in this chapter have been used to train and test them.

Chapter 4

ToF Depth Data Refinement

This chapter presents some techniques to improve the overall depth accuracy of stand alone ToF cameras. As mentioned in Chapter 2, ToF cameras are gaining popularity for the simplicity of their processing operations, the possibility to generate a dense depth map, the absence of artifacts due to occlusions and the independence from scene texture. Apart from these good aspects, they have also some flaws for which they need to be further analyzed and improved. In particular, the main task of the methods presented in this chapter, initially introduced in our papers [1,2], is to reduce the noise due to thermal and photon shot noise, which are zero-mean error, and correct the depth overestimation due to the multi-path interference (MPI) phenomenon. Before starting with the description and evaluation of the proposed methods, the next section will review the existing literature about the topic.

4.1 Literature about ToF Data Refinement

Regarding the reduction of thermal and photon shot noise, different approaches can be found. Usually, these are inspired by image denoising techniques and adapted to the particular nature of ToF data. Among them, in [43] a denoising method based on a wavelet analysis of the data and guided by the noise statistic is proposed. Bilateral filtering or total variation techniques could be used as well as suggested by Lenzen et al. in [44]. Recently, a denoising method using *non-local means* has been proposed

by Georgiev et al. in [45] to refine ToF depth in case of very limited reflected light, usually the case for portable ToF cameras using a low power projector. All these methods are very performing and able to effectively reduce the thermal noise and photon shot noise.

Differently, many methods for MPI correction have been proposed [46], but it remains an open problem. MPI correction methods can be classified in consideration to what kind of ToF data they exploit, if single frequency or multi-frequency ToF data, if the standard ToF hardware is customized and if the methods are analytical or data driven.

The methods which use single frequency ToF data exploit some reflection models in order to estimate the geometry of the scene and correct MPI as done by Fuchs in [47], where reflections with a maximum of two bounces are considered. This method is further extended in [48] where multiple albedo and reflection bounces are taken in account. Jimenez et Al. proposed a radiometric model to simulate ToF acquisitions and the reflection phenomenon and then to correct MPI through a non linear optimization problem [49]. These methods are slow and computational expensive. Moreover, they are not able to manage the MPI phenomenon related to interfering rays coming from outside the field of view of the ToF camera.

The methods based on customization of the ToF device, usually modify the ToF projector and the light signal emitted by it. Kadambi et al. [50] proposed to temporally modulate the emitted sine wave with a random on-off code. The reflected light signal is demodulated using the same random on-off code in order to estimate the scene impulse response and so estimate the return time of the first ray. Other methods spatially modulate the ToF light signal, e.g., in [51–53] a modified ToF light source is used to project a sequence of patterns to separate the direct light, reflecting only once inside the scene, from the interfering rays, the so called global light, in case of diffuse reflections.

The analytical methods using multi-frequency ToF data usually exploit the *frequency diversity* of the MPI phenomenon (see Section 2.3.2) to solve it. The light is described as the summation of few distinct sinusoidal waves which are interfering one another in case of MPI. They try to recover the interfering rays and the one with the shortest path is assumed to be the one carrying the correct depth information (direct light). Among the methods belonging to this family, it is possible to

find the methods proposed by Bhandari et al. [54] and the SRA method proposed by Freedman et al. [34]. In [54] a closed form solution for MPI removal is proposed and moreover the authors introduced a theoretical lower bound for the number of modulation frequencies required to solve the interference on a fixed number of rays. The issue with this method is that it is required to use data collected using a high number of modulation frequencies in order to have an accurate MPI correction. However, this is infeasible on in-the-wild recordings for frame rate constraints. Differently, the method proposed by Freedman in [34] uses ToF data acquired using just three modulation frequencies. The idea is to set-up a linear optimization problem modelling the MPI phenomenon as the interference of few and distinct interfering rays, whose amplitude and phase are estimated exploiting the *frequency diversity* of MPI. The authors implemented a *look-up-table* approach for a real-time implementation of the method. The limitation of these two approaches is that the hypothesis of few interfering rays is related to just the specular reflections and does not take in account diffuse reflections. For this reason, they have a reduced accuracy when tested on real scenes.

In order to avoid the use of explicit reflection models, data driven approaches correcting MPI have been presented recently. Son et al. in [55] use a deep neural network, trained on labelled single frequency 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 [30] proposed an auto-encoder Convolutional Neural Network (CNN) to refine ToF data acquired at 20 MHz. This is trained in two phases: in the first phase, real depth data without ground truth are used for the unsupervised training of the auto-encoder in order to reconstruct the input at the output of the CNN; then, the encoder part is kept fixed and the decoder part is trained with a synthetic dataset in order to learn how to correct MPI.

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 [31, 32]. In these methods, the CNNs have been trained on synthetic data, but the performance on real data has been investigated only from a qualitative point of view in [32] and on a single corner scene in [31].

The next of this chapter introduces the approaches for ToF data denoising which I worked on during my Ph.D.:

- the first is based on a customization of the ToF projector. A spatial modulation of the standard ToF light signal is implemented by means of sinusoidal patterns, using the technique introduced by Whyte et al. [51]. A structured light (SL) approach is implemented on the ToF device to estimate a SL depth map of the scene. This is fused with the ToF depth estimation to obtain a more accurate depth.
- The second approach uses multi-frequency ToF data as input of a CNN to refine the depth data. Two methods based on this approach are presented. The first uses the CNN to estimate the MPI depth corruption, that is directly subtracted on the noisy depth map. Finally an *adaptive bilateral filter* is used to filter out the zero-mean error. The second proposed method uses a CNN to directly refine the ToF data.

4.2 Spatio-Temporal Modulated ToF

The first method for ToF denoising and MPI correction introduced in this thesis is based on the separation of the direct and global component of the light through the projection of multiple sinusoidal patterns as proposed by Whyte and Dorrington [51,56]. The method presented in [51], here named Spatio-Temporal modulated ToF (STM-ToF), allows to correct a wide range of MPI phenomena as inter-reflection and sub-surface scattering, but the obtained depth estimations are noisier if compared with standard ToF systems. The proposed extension of this method starts from this rationale but goes further by implementing a SL depth estimation approach on a ToF system based on this idea. It is specifically designed for short range scenes. Fig. 4.1 depicts the complete scheme of the proposed method trying to summarize all the main components of the depth estimation process. This starts from the data acquisition on the target scene and on a reference scene, composed by a wall at a known distance, and continues with the depth estimation with the ToF and SL principle. Finally, it ends with the proposed *Maximum Likelihood* fusion of the two depth fields. In order to evaluate the performance of the proposed method, we

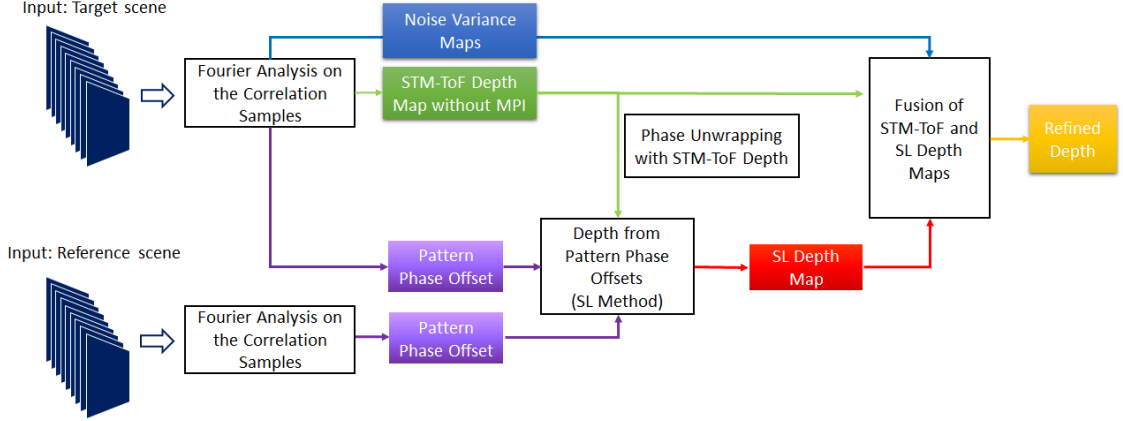


Fig. 4.1: Proposed method flow chart.

tested it on a synthetic dataset simulated with the *Sony ToF Explorer* simulator realized by Sony Eutec (See Section 3.3.2). The evaluation scenes are the subset of scenes contained in $S_{1,train}$ and $S_{1,test}$ which have a maximum depth of 4 m, since the proposed method is designed for short range scenes.

Before going on the implementation details of our method, the STM-ToF depth acquisition process is analysed together with its MPI removal approach. Then the proposed method is discussed and its evaluation will follow.

4.2.1 Introduction to STM-ToF

In order to obtain a depth estimation free from MPI distortion, it is required to separate the direct component of the light from the global one. The approach we exploited is inspired by the method described by Whyte in [51], but extends it taking into account the fact that most real world ToF cameras work with square wave reference signals. The system presented in [51] is composed by a standard ToF sensor and a modified ToF projector that emits a periodic light signal (Fig. 4.2): the standard temporally modulated ToF signal of Eq. 2.3 is also spatially modulated by a predefined intensity pattern. In the developed method we are going to consider the sinusoidal intensity pattern

$$L_{x,y}(\omega_r \tau_i) = \frac{1}{2} \left(1 + \cos(\omega_r \tau_i - \theta_{x,y}) \right) \quad (4.1)$$

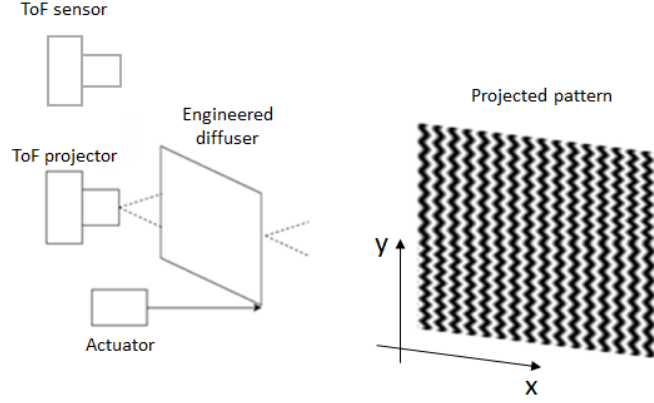


Fig. 4.2: ToF acquisition system for direct and global light separation.

	$i = 0$	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$	$i = 8$
Sampling point of the ToF correlation function $\omega_r \tau_i$	0	$\frac{2\pi}{9}$	$\frac{4\pi}{9}$	$\frac{6\pi}{9}$	$\frac{8\pi}{9}$	$\frac{10\pi}{9}$	$\frac{12\pi}{9}$	$\frac{14\pi}{9}$	$\frac{16\pi}{9}$
Phase shift of the projected pattern $l\omega_r \tau_i$, $l=3$	0	$\frac{2\pi}{3}$	$\frac{4\pi}{3}$	0	$\frac{2\pi}{3}$	$\frac{4\pi}{3}$	0	$\frac{2\pi}{3}$	$\frac{4\pi}{3}$
Employed pattern									

Fig. 4.3: Synchronization between phase shift of the projected pattern and phase sample of the ToF correlation function.

where (x, y) denotes a pixel position on the projected image, $\theta_{x,y} = \frac{2\pi x}{p} + \sin\left(\frac{2\pi y}{q}\right)$ is the pattern phase offset at the projector pixel (x, y) , p and q are respectively the periodicity of the pattern in the horizontal and in the vertical direction, l is a positive integer number and $\omega_r \tau_i \in [0; 2\pi)$ is a sampling point of the ToF correlation function as defined in Eq. 2.6. The projector and the camera are assumed to have parallel image planes. Notice that for each computed sample of the ToF correlation function a specific pattern is used to modulate the standard ToF signal of Eq. 2.3.

Denoting the angular modulation frequency of the ToF camera as $\omega_r = 2\pi f_{mod}$, the projected pattern $L(\omega_r \tau_i)$ is phase shifted with angular frequency $l\omega_r$. Fig. 4.3 shows the pattern projection sequence for the case in which $l = 3$ and the ToF camera evaluates 9 samples of the correlation function.

Here on, we assume that the ToF signal is modulated by the phase shifted

patterns depicted in Fig. 4.3 considering the proposed synchronization between the pattern phase offsets and the ToF correlation sampling points. If the spatial frequency of the projected patterns is high enough to separate the direct and global component of the light [57] (this holds in case of absence of specular reflections), it results that only the direct component of the light is modulated by the patterns. In this case, the ToF correlation function Eq. 2.10 computed by the ToF camera on a generic pixel can be modelled as:

$$c(\omega_r \tau_i) = B + A \cos(\omega_r \tau_i + \phi_d) + A_g \cos(\omega_r \tau_i + \phi_g) + \frac{\pi A}{2} \cos(l \omega_r \tau_i - \theta) + \frac{A}{2} \left[\cos((l-1)\omega_r \tau_i - \phi_d - \theta) + \cos((l+1)\omega_r \tau_i + \phi_d - \theta_{x,y}) \right]. \quad (4.2)$$

As in Eq. 2.10, we define b_r as the ambient light offset, a_r as the amplitude of the direct component of the received ToF signal, $b_{r,g}$ as the offset of the global component of the light and $a_{r,g}$ as the amplitude of the received global component. By using this nomenclature, it comes out that $B = \frac{1}{f_{mod}} \left(\frac{b_r}{2} + \frac{a_r}{8} + \frac{b_{r,g}}{2} \right)$ is an additive constant that represents the received light offset, $A = \frac{a_r}{4\pi f_{mod}}$ is proportional to the power of the direct component of the received light, $A_g = \frac{a_{r,g}}{\pi f_{mod}}$ is proportional to the power of the global component of the received light. ϕ_d is the phase offset related to the direct component of the light (not affected by MPI), ϕ_g is the phase offset related to the MPI phenomenon and $\theta_{x,y}$ is the phase offset of the projected pattern on the specific scene point observed by the considered ToF pixel. Notice that both ϕ_d (through the ToF model of Section 2.3) and $\theta_{x,y}$ (through the SL approach of Section 4.2.3) can be used to estimate the depth at the considered location. In the following of this section, it is considered $l = 3$ since it avoids aliasing with just 9 samples of the correlation function. No other value of l brings to a smaller number of acquired samples. By using this setting and opportunely arranging the acquisition process, the projector has to update the emitted sinusoidal patterns at 30 fps in order to produce depth images at 10 fps. A complete derivation of Eq. 4.2 can be found in Appendix C.

A first difference with the analysis carried out in [51,56] is that in these works the reference signal used for correlation by the ToF camera is a sine wave without offset, instead in our model we use a rectangular wave since this is the waveform used by

most real world ToF sensors. This choice in the model brings to an harmonic at frequency $l = 3$ that was not considered in [51,56], and this harmonic is informative about the pattern phase offset θ . In the next section and more in detail in Appendix C, it is shown that by estimating θ from this harmonic allows a more accurate estimation than computing it from the $(l - 1) - th$ and $(l + 1) - th$ harmonics. In order to estimate a depth map of the scene free from MPI we are going to apply Fourier analysis on the retrieved ToF correlation signal of Eq. 4.2 as also suggested in [51,56]. By labelling with φ_k the phase of the $k - th$ harmonic retrieved from the Fourier analysis we have that:

$$\phi_d = (\varphi_4 - \varphi_2)/2, \quad \theta = -\varphi_3 \quad (4.3)$$

By estimating ϕ_d as mentioned above we can retrieve a depth map of the scene that is not affected by MPI but the result appears to be noisier than standard ToF acquisitions as discussed in the next section. We are going to name this approach for MPI correction as STM-ToF, that is the acquisition based on the method proposed by Whyte [51]. In Section 4.2.3, θ will be used for SL depth estimation.

4.2.2 Error Propagation Analysis

In order to evaluate the level of noise of the depth estimation with STM-ToF acquisition, we used an error propagation analysis to predict the effects of the noise acting on ToF correlation samples on the phase estimation. In particular, we consider the effects of the *photon shot* noise. The noise variance in standard ToF depth acquisitions can be computed with the classical model described in Section 2.3.2:

$$\sigma_{d_{std}}^2 = \left(\frac{c}{4\pi f_{mod}} \right)^2 \frac{B_{std}}{2A_{std}^2}. \quad (4.4)$$

where $A_{std} = \frac{a_r}{2\pi f_{mod}}$ and $B_{std} = \frac{1}{f_{mod}} \left(\frac{b_r}{2} + \frac{a_r}{4} \right)$. In a similar way we can estimate the level of noise in the proposed system.

If we assume to use nine ToF correlation samples $c(\omega_r \tau_i)$ with $\omega_r \tau_i = \frac{2\pi}{9} i$ for $i = 0, \dots, 8$ affected by photon shot noise, it is possible to demonstrate (the complete derivation of the model through error propagation is in Appendix C) that the mean value of the noise variance in the STM-ToF depth estimation is

$$\bar{\sigma}_{d_{noMPI}}^2 = \left(\frac{c}{4\pi f_{mod}} \right)^2 \frac{4B}{9A^2} \quad (4.5)$$

where A and B are defined as in the previous section. Here we are considering only the mean value of the noise variance for the estimated depth map, since the complete formulation contains also sinusoidal terms which depend on the scene depth and the pattern phase offset.

By comparing Eq. 4.4 and 4.5 and opportunely considering the scaling effects due to the modulating projected pattern, if $b_r \gg a_r$ (that is when the ambient light component is much bigger than the received ToF signal amplitude, usually the case) we have that $\bar{\sigma}_{d_{noMPI}}^2 / \sigma_{d_{std}}^2 = 3.56$, i.e., the noise variance obtained by using the approach in [51] is around four times noisier if compared with a standard ToF camera that uses the same peak illumination power.

4.2.3 Applying Structured Light on ToF Sensors

In this section, we propose to use the pattern phase offset θ observed by the ToF sensor in order to estimate a second depth map of the scene with a *structured light* (SL) approach. The phase image θ can be estimated with the approach of Section 4.2.1, i.e., from Eq. 4.3. Notice that our model considers a rectangular wave as reference signal (that is typically the case in commercial ToF cameras) and we could exploit the harmonic at frequency $l = 3$ of Eq. 4.2, allowing to obtain a higher accuracy than using the second and the fourth harmonics as in [51]. More in detail, if we compare the level of noise in estimating θ from the second and fourth harmonics (i.e., as done in [51]) with the noise in the estimation from the third harmonic (as we propose), we have that:

$$\bar{\sigma}_{\varphi_2, \varphi_4}^2 = \frac{4B}{9A^2}, \quad \bar{\sigma}_{\varphi_3}^2 = \frac{8B}{9\pi^2 A^2}. \quad (4.6)$$

Thus θ estimated from the third harmonic has a noise variance about four times smaller if compared with the estimation from the second and fourth harmonics.

The estimated pattern phase offset can be used to compute the second depth map of the scene with a SL approach. If the pattern phase image θ_{ref} is captured on a reference scene for which the distance d_{ref} from the camera is known, e.g.,

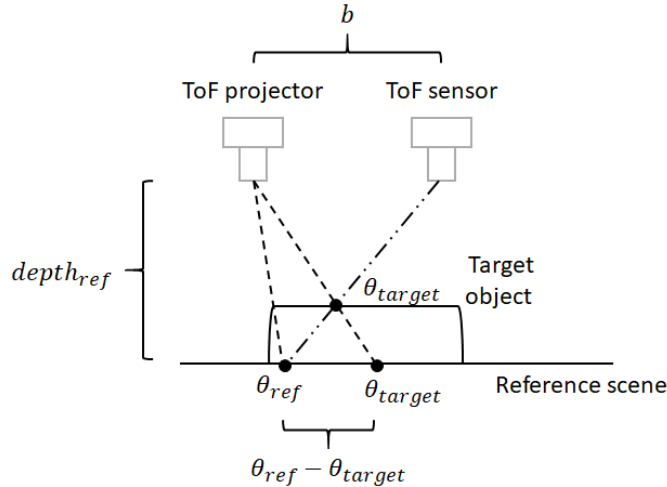


Fig. 4.4: Geometry of the SL acquisition on target and reference scenes.

a straight wall orthogonal to the optical axis of the camera, then it is possible to estimate the depth of any target scene by comparing pixel by pixel the estimated phase image θ_{target} with the reference one (see Fig. 4.4).

A similar approach has been exploited by Xu et Al. in [58] for standard color cameras in a structured light system. In that case a phase unwrapping of the phase images has to be applied before being able to estimate the depth. This can be obtained by projecting multiple lower frequency patterns on the scene. Assuming that θ_{ref} and θ_{target} have been phase unwrapped in θ_{ref}^{PU} and θ_{target}^{PU} , the depth of the target scene can be estimated as:

$$d_{SL} = d_{ref} \left(1 + \frac{Q}{b} (\theta_{ref}^{PU} - \theta_{target}^{PU}) \right)^{-1} \quad (4.7)$$

where d_{ref} is the distance between the reference scene and the ToF camera, Q is a parameter related to the acquisition system setup that can be estimated by calibration and b is the baseline between the camera and the projector, 3 cm in the proposed setup. In standard SL systems a bigger baseline (e.g., 10 cm) is required to reliably estimate depth in the far range. Here, we can afford a smaller baseline since we can exploit the ToF depth (more reliable in the far range) in the fusion process described in Section 4.2.4 to obtain a more reliable depth map. Moreover, a smaller baseline reduces the problem of occlusions in standard SL estimation.

We avoid the use of additional patterns for phase unwrap θ by employing the

ToF depth map computed with the method of Section 4.2.1. The idea is to use the phase image θ_{ToF} , the one that would have produced the ToF depth map in case of a SL acquisition, to apply an implicit phase unwrapping. We can compute the depth with the SL approach assisted by the ToF estimation as:

$$d_{SL} = d_{ref} \left(1 + \frac{d_{ref} - d_{ToF}}{d_{ToF}} + \frac{Q}{b} (\theta_{ToF} - \theta_{target})_{[-\pi;\pi]} \right)^{-1} \quad (4.8)$$

where:

$$\theta_{ToF} = \theta_{ref} - \frac{b}{Q} \cdot \frac{d_{ref} - d_{ToF}}{d_{ToF}} \quad (4.9)$$

In this approach we are using θ_{ToF} as a new reference phase offset to be used to estimate the SL depth map related to θ_{target} . The complete derivation of the SL implicit phase unwrapping is reported in Appendix C.2.

In this case the variance of the noise corrupting d_{SL} can be computed from error propagation analysis (see the Appendix C.3 for more details):

$$\sigma_{d_{SL}}^2 = \left(Q \frac{d_{target}^2}{d_{ref} b} \right)^2 \sigma_{\theta}^2. \quad (4.10)$$

From Eq. 4.10 it is possible to notice that the depth estimation accuracy improves if we increase the baseline between the sensor and the projector and it degrades with the increase of the depth that we are going to estimate. This is a common behavior for SL systems. The reference scene distance d_{ref} has no effect in the accuracy since Q is directly proportional to d_{ref} .

4.2.4 Fusion of ToF and SL Depth Maps

The approaches of Sections 4.2.1 and 4.2.3 allow to compute two different depth maps, one based on the Time-of-Flight estimation with MPI correction (the STM-ToF acquisition) and one based on a SL approach. In the final step, the two depth maps have to be fused into a single accurate depth image of the scene. The exploited fusion algorithm is based on the Maximum Likelihood (ML) principle [59]. The idea is to compute two functions representing the likelihoods of the possible depth values given the data computed by the two approaches and then look for the depth value Z that maximizes at each location the joint likelihood that is assumed to be composed

by the independent contributions of the two depth sources [59, 60]:

$$d_{fus}(i, j) = \operatorname{argmax}_Z P(I_{ToF}(i, j)|Z)P(I_{SL}(i, j)|Z) \quad (4.11)$$

where $P(I_{ToF}(i, j)|Z)$ and $P(I_{SL}(i, j)|Z)$ are respectively the likelihoods for the STM-ToF and SL acquisitions for the pixel (i, j) while $I_{ToF}(i, j)$ and $I_{SL}(i, j)$ are the computed data (in our case the depth maps and their error variance maps). The variance maps are computed using the error propagation analysis made in Sections 4.2.2 and 4.2.3 starting from the data extracted from the Fourier analysis of the ToF correlation function. They allow to estimate the depth reliability in the two computed depth maps and they are fundamental in order to guide the depth fusion method towards obtaining an accurate depth estimation. Different likelihood structures can be used, in this work we used a *Mixture of Gaussians* model that is more robust against the *flying pixel* issue [61]. For each pixel and for each estimated depth map (from SL or STM-ToF approach), the likelihood is computed as a weighted sum of Gaussian distributions estimated on a patch of size $(2w_h + 1) \times (2w_h + 1)$ centred on the considered sample. For each pixel of the patch we model the acquisition as a Gaussian random variable centred at the estimated depth value with variance equal to the estimated error variance. The likelihood is given by a weighted sum of the Gaussian distributions of the samples in the patch with weights depending on the Euclidean distance from the central pixel. The employed model in the case of the ToF measure is given by the following equation:

$$P(I_{ToF}(i, j)|Z(i, j)) \propto \sum_{o, u=-w_h}^{w_h} \frac{e^{-\frac{\|(o, u)\|_2}{2\sigma_s^2}}}{\sigma_{ToF}(i+o, j+u)} e^{-\frac{(d_{ToF}(i+o, j+u)-Z(i, j))^2}{2\sigma_{ToF}^2(i+o, j+u)}} \quad (4.12)$$

where $\sigma_{ToF}(i, j)$ is the standard deviation of the depth estimation noise for pixel (i, j) as computed in Section 4.2.2, σ_s manages the decay of the distribution weights with the spatial distance in the considered neighbourhood of (i, j) . In our experiments we fixed $\sigma_s = 1.167$ and $w_h = 3$, i.e., we considered data in a 7×7 neighbourhood of each pixel. The likelihood $P(I_{SL}(i, j)|Z(i, j))$ for the SL depth is evaluated in the same way just by replacing ToF data with SL data.

In order to speed up the fusion of the two depth maps, we restricted the candidates for $d_{fus}(i, j)$ in a range of 3 times the standard deviation from the computed depth values for both the ToF and SL estimations.

4.2.5 Results of the Fusion Method on the STM-ToF

In this section, the performance of the proposed fusion method is analysed and compared with standard ToF acquisitions, with the spatio-temporal modulation implemented on the ToF system (STM-ToF) introduced in [51] and described in Section 4.2.1 and finally with the multi-frequency method of Freedman et al. (SRA) [34]. For the comparison with [34] we performed the experiments using 3 modulation frequencies, i.e., 4.4, 13.3 and 20 MHz in order to have the maximum frequency equal to the one we used for a fair comparison and the others selected with scaling factors similar to those used in [34]. We have used a synthetic dataset for which the ground truth geometry of the scenes can be accurately extracted to test the different approaches. In this way a reference depth ground truth for the ToF acquisitions is available and can be used for the numerical evaluation. This synthetic dataset is simulated with the *Sony ToF Explorer* simulator realized by Sony Eutec (see Section 3.3.2). The evaluation scenes are a subset of 21 scenes contained in $S_{1,train}$ and $S_{1,test}$ which have a maximum depth of 4 m, since the proposed method is designed for short range scenes. The camera parameters used in the simulations are the same of the Sony DS541 camera. The 21 ToF acquisitions have as subject scenes with complex textures and objects with different shape and size, in order to test the methods on various illumination and MPI conditions.

The performance of the proposed method are first discussed from a qualitative and then from a quantitative point of view. Fig. 4.5 shows the depth maps and the corresponding error maps for the different components of our approach on four synthetic scenes. In particular, the first row contains the ground truth depth maps. The second and the third rows show respectively the depth maps and the error maps (equal to the acquired depth minus the true depth) for a standard ToF camera using four samples of the correlation function. The fourth and the fifth rows show the results for the STM-ToF approach based on [51] and implemented as discussed in Section 4.2.1. In the sixth and seventh rows instead we collected the depth and

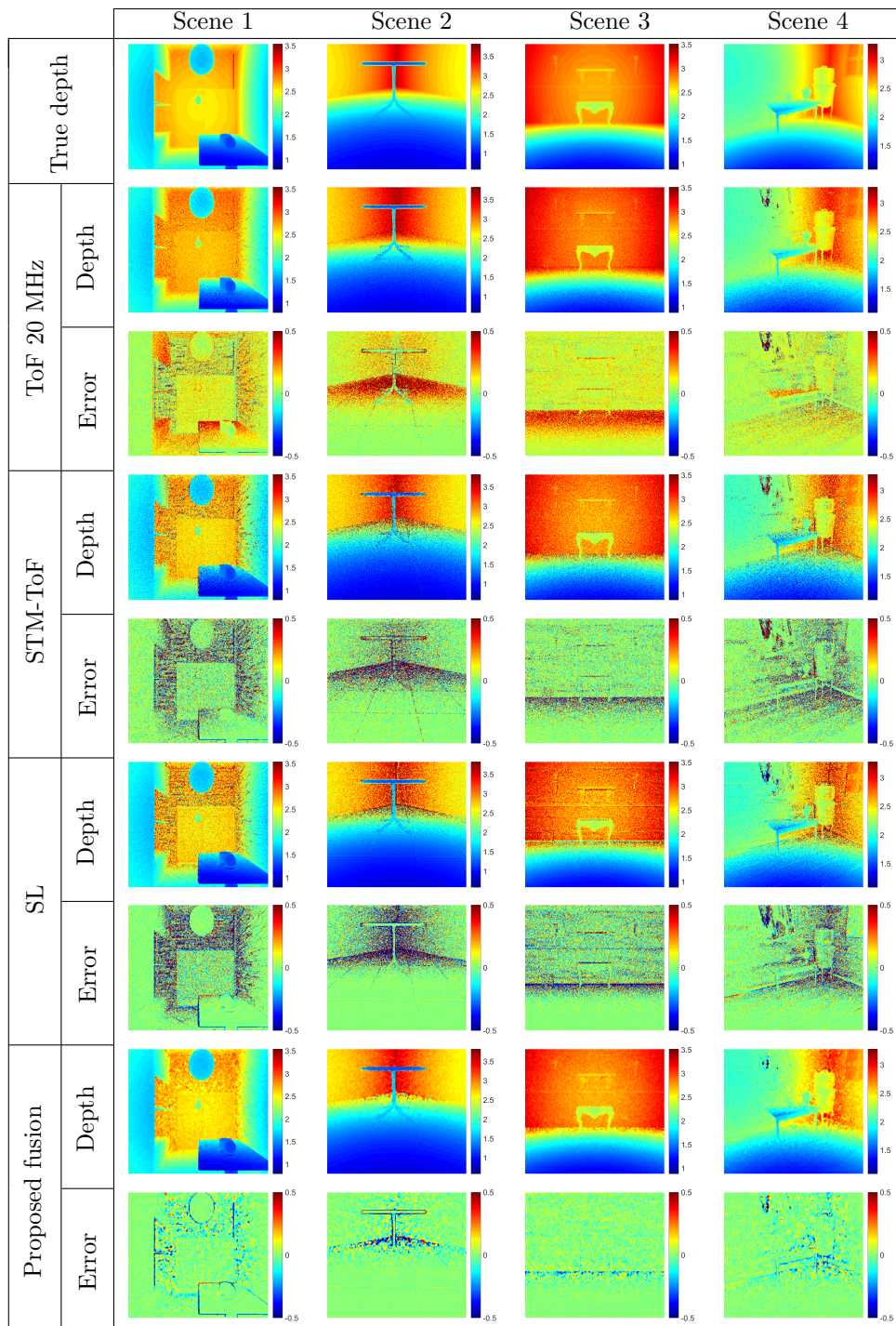


Fig. 4.5: Qualitative comparison for STM-ToF, SL and their fusion on some sample scenes. All the values are measured in meters. In the error maps, dark red is equivalent to 0.5 m, dark blue to -0.5 m and green to no error.

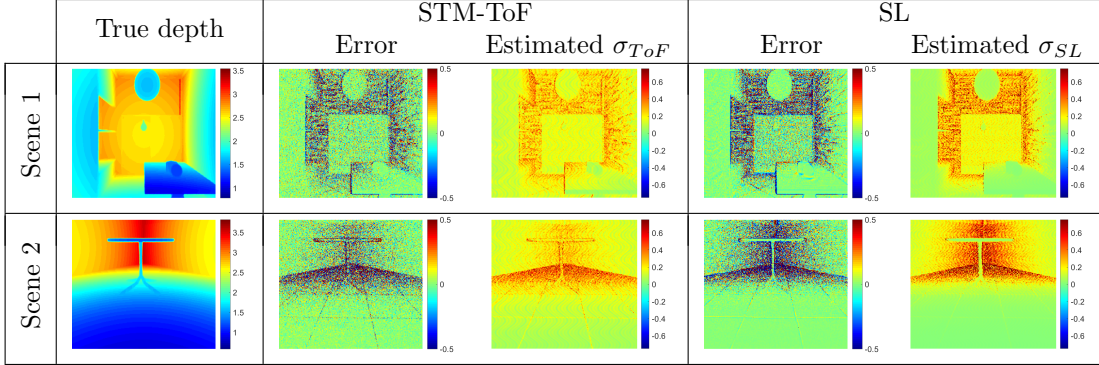


Fig. 4.6: Qualitative comparison for STM-ToF and SL regarding their true error and the estimate of their standard deviation, σ_{ToF} and σ_{SL} , on some sample scenes. All the values are measured in meters. In the error maps, dark red is equivalent to 0.5 m, dark blue to -0.5 m and green to no error.

the error maps obtained with the SL approach on ToF acquisitions as described in Section 4.2.3. The output of the proposed fusion approach given by the combination of the MPI correction method based on [51] with the SL depth maps by exploiting the estimated standard deviation of their error, σ_{ToF} and σ_{SL} used in Equation 4.11 and 4.12 and computed as discussed in Section 4.2.2 and 4.2.3, is represented in the eighth and ninth rows of Fig. 4.5. Notice that the two depth fields going to be fused are captured together with a single ToF acquisition as described in Section 4.2.1. Fig. 4.6 shows a comparison between the error for the STM-ToF and SL and the estimation of their standard deviations, σ_{ToF} and σ_{SL} , used in the fusion process. The σ_{ToF} and σ_{SL} are a good metric to evaluate the error corruption distribution and so they can reliably guide the fusion of the two depth fields.

As it is possible to observe from Fig. 4.5, the standard ToF acquisitions are characterized by a dramatic overestimation of the depth near to the corners caused by the MPI phenomenon. Differently, by using the STM-ToF approach the depth overestimation due to MPI is reduced (no more uniform red regions in the error maps) as it can be seen in column 2 and 3 from the corners composed by the floor and walls. On the other hand, the data appears to be much more noisy, in particular in regions where only a small amount of light is reflected back (e.g., distant corners and the borders of the tiles on the floor in column 2). This problem of the STM-ToF approach was already pointed out in Section 4.2.2, indeed the depth generated with

	MAE (<i>all</i>) [mm]	MAE (<i>valid*</i>) [mm]
ToF 20MHz	73.9	56.8
STM-ToF [51]	93.4	65.2
SL	80.8	49.7
SRA [34]	-	50.8
Proposed	21.8	14.2

Table 4.1: Mean Absolute Error (MAE) for the compared approaches on the synthetic dataset averaged on the 21 scenes (measured in millimeters).

*: The minimization used by SRA does not give an outcome for all points, for a fair comparison we also show the results on the subset of points computed by SRA.

this approach has an error variance that is about four times higher than a standard ToF acquisition with the same settings. Concerning the depth maps estimated with the SL approach, also in this case the overestimation due to MPI is absent, but there are artifacts not present in standard ToF acquisitions. The overestimation close to corners is almost completely removed and the amount of noise on flat surfaces is less than in the ToF approach. On the other hand, there are artifacts in heavily textured regions (e.g., on the back in column 1) and sometimes the color patterns can propagate to the depth estimation (the following section discusses this issue). By observing the depth and error maps obtained with the proposed fusion approach, it is possible to see that both the MPI corruption and the zero-mean error have been reduced obtaining a much higher level of precision and accuracy when compared with the other approaches. In particular, notice how there is much less zero-mean noise, the MPI corruption is limited to the points extremely close to the corners and artifacts of both methods like the ones on the border of the tiles have been removed, without losing the small details in the scenes.

The qualitative discussion is confirmed by the quantitative comparison. We used the *Mean Absolute Error* (MAE) as metric for the comparison. Table 4.1 collects the results averaged on the 21 scenes that compose the dataset while Fig. 4.7 contains a pictorial representation of the error histogram.

The MAE values and the histogram show that standard ToF acquisition has a bias due to the overestimation caused by MPI. This bias is much reduced by the STM-ToF, SL, SRA and proposed methods. The STM-ToF [51] strongly reduces

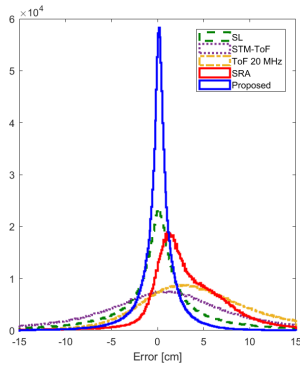


Fig. 4.7: Histogram of the error distribution for the considered methods.

ToF 20MHz Depth Error	STM-ToF Depth Error	SL Depth Error	Fusion Depth Error	Ground Truth

Fig. 4.8: Critical cases in which the method reduces the overall level of error but adds small distortions. All the values are measured in meters. In the error map dark red is equivalent to 0.5 m, dark blue to -0.5 m and green to no error.

MPI but it has a high MAE due to the increased noise level. Concerning SRA, it reduces the positive bias in the error due to MPI but not so effectively as the proposed method. The main reasons for this not optimal behavior of SRA are that it is susceptible to noise and that the sparseness assumption for the global component is not completely fulfilled in a diffuse reflection scenario. Finally, it is possible to notice that the proposed method outperforms all the other approaches achieving a lower MAE and removing MPI. Furthermore, the histogram in Fig. 4.7 shows that the initial biased error of the standard ToF estimation is moved close to 0 by the proposed method and that the overall variance is much smaller for our approach compared to all the others.

In Fig. 4.8 instead we depicted a couple of critical cases in which the proposed method is able to reduce the overall level of error, but it adds some small undesired distortions. In the first case (row 1), the SL estimation is corrupted in the regions

that present a strong local variation of the color (see the vertical stripe in the *color view*), a well-known problem of *Structured Light* systems. In the fusion process the effect of this issue are reduced but not completely removed. The second line of Fig. 4.8 shows that the SL estimation adds a distortion near to the center of the corner due to the refraction of the patterns. This is a second well-known issue related to the systems which employ SL approach [36]. This could be solved by increasing the spatial frequency of the projected patterns but the small resolution of current ToF camera makes this solution challenging to apply. The aforementioned distortions are reduced but not completely corrected by the proposed fusion approach.

We have not presented an evaluation of this method implemented on a real device yet. However, we are considering to build a prototype camera using a modified ToF device in combination with a DMD projector as also done by O’Toole et al. in [62].

4.3 Data Driven ToF Data Refinement

The second approach introduced in this chapter is a data driven ToF data refinement approach. It does not require any hardware modifications on the standard commercial ToF cameras, and in principle it can be implemented on any commercial multi-frequency ToF (MF-ToF) cameras able to record depth and amplitude using the modulation frequency set to 20, 50 and 60 MHz. The task is to obtain accurate ToF depth data by removing MPI corruption and reducing zero-mean error related to shot noise and thermal noise. Here, two different implementations are presented and their performance will be evaluated.

In the first method, here named R-CNN+B, the MPI is estimated by exploiting a Convolutional Neural Network (CNN) whose input are data extracted from a MF-ToF camera, while the zero-mean error is reduced by an *adaptive bilateral filter*, guided by the noise statistic estimated on the input data.

In the second method, here named TD-CNN, a CNN has to correct both the MPI and the zero-mean error using as input the same data used by the previous method and to output the ToF depth (TD).

In the next of this section, the basic principles of CNNs will presented and then their use in R-CNN+B and TD-CNN will be introduced. The performance of the proposed ToF depth refinement methods will be evaluated and compared with other

state-of-the-art methods on both a synthetic and real datasets.

4.3.1 Introduction to Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specialization of Artificial Neural Networks (ANNs), which are computing systems which “learn” to perform tasks by observing examples of data and inferring some knowledge from these. In particular, the tasks considered in this thesis are ToF data refinement and semantic segmentation. ANNs try to emulate the learning and recognition process in human brain. The basic building element of an ANN is the artificial neuron, that is its basic computing unit. As depicted in Fig. 4.9, the neuron takes as input different scalar values and it computes their weighted sum, possibly adding a bias b , and sequentially applies a non linear activation function f on it.

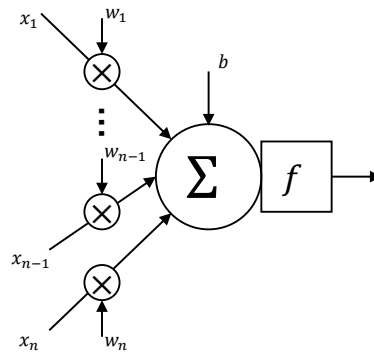


Fig. 4.9: Working scheme of the artificial neuron.

One of the most popular non linear activation functions is the *Rectified Linear Unit* (ReLU) that applies the following operation to its input

$$\text{ReLU}(x) = \max(0; x). \quad (4.13)$$

In the next of this thesis, ReLU activation will be employed in ANNs if not differently mentioned. However, other activation functions as the *sigmoid* and the *leaky ReLU* exist and a comprehensive analysis and discussion about them can be found in [63].

In ANNs, the neurons can be organized in layers and different layers are typically applied in cascade. In this way, each layer produces some output that will be used

as input features for the next layer. This enables to create complex functions and so to extract high level information from the input data. Fig. 4.10 shows an example of *fully connected* neural network in which there are three layers with six, five and five neurons. The peculiarity of *fully connected* neural networks is that each node of the $i - th$ layer is connected to all the neurons of the $(i + 1) - th$ layer. This kind of structure is very powerful since each element of the output is function of the whole input. However, the number of parameters is huge since each layer has a number of parameters that can be computed as

$$n_{params, layer\ i} = n_{neurons, layer\ i-1} \cdot (1 + n_{neurons, layer\ i}) \quad (4.14)$$

and it linearly increases if the number of neurons in the layer before, the size of the input, increases.

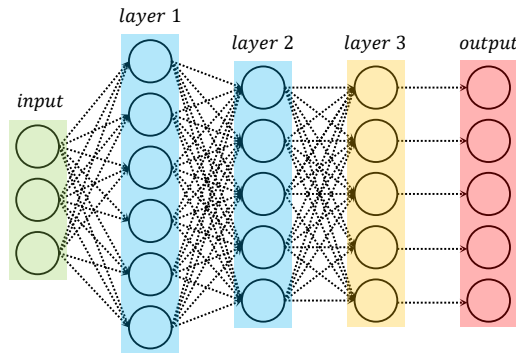


Fig. 4.10: Example of *fully connected* ANN, with layers organized in cascade. This network has three layers respectively with six, five and five neurons. The third layer produces the output of the network.

Convolution Neural Networks (CNNs) are neural networks which require a smaller number of parameters. To do so, it is assumed that an output of the neural network is only influenced by a neighborhood of data in the input and so the spatial correlation of the data is exploited to estimate the output. In case of ordered data as audio signals (1D), gray scale images (2D) or color images (3D, tensor), the neurons in a layer are not connected to the whole input but they are organized as a kernel of a filter that is then convoluted with the input to produce the output. In case of tensors, the convolutional kernel contains $H \times W \times D$ weights, where H , W and D

are respectively the height, width and depth of the kernel representing the neuron. The depth of the kernel is usually equal to the depth of the input feature (it can be different for 3D convolutional kernels, but this case will not be considered in this thesis). In each layer, it is possible to have multiple kernels and each of them contributes to an entry in the third dimension of the produced feature. For example, if the input feature of a layer has dimension $h \times w \times D$ and there are K kernels with dimension $H \times W \times D$, then the output feature will have dimension $h' \times w' \times K$. The application of padding on the input or eventual stride operations influence the actual value of h' and w' .

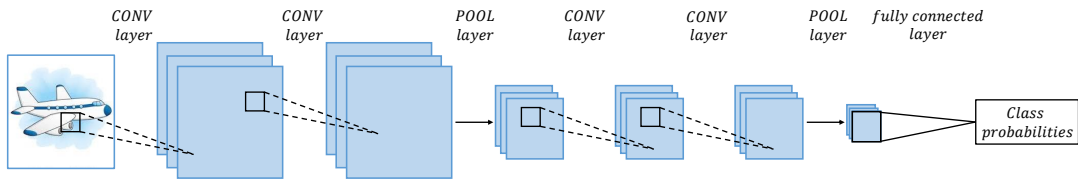


Fig. 4.11: Example of CNN classifier.

Fig. 4.11 shows an example of a CNN classifier, in which the input is a color image and a series of convolutional layers are applied in cascade. In the image, the black box highlights where the convolutional kernel is applied. The kernel is used to filter the whole input feature map. In CNNs, *pooling* layers are usually applied. These layers sub-sample the internal features in order to reduce their size and consequently increase the receptive field of the network. Pooling operators could be implemented as max or mean pooling, which respectively extract a patch from the input feature and for it just one value, the output of the aforementioned operations on the patch, is stored. Usually, CNN classifiers have as last stage a *fully connected* layer in which the probability that the input image belongs to a specific class is associated to the output of a neuron.

Neural Network Training

After defining the architecture of a neural network, it is required to train it to make it able to accomplish a task, that can be a classification or a regression task. To train a network means to optimize all the network weights and biases, $\theta \in \mathcal{R}^d$, to minimize a loss function L that indicates how good is the network at the target

task. Here, the supervised training case is considered. This is the case when there exists a training set of data with the related ground truth, the desired network output for the considered input. However, unsupervised training is possible as it is discussed in the next chapter.

Given a set of input, the loss L has to measure the error between the output of the network and the true value that is desired. In case of a regression task as depth data denoising, the l_1 norm is a suitable loss function. In this case, the loss can be formulated as

$$L_1 = E[|net(input) - gt|_1]. \quad (4.15)$$

Unfortunately, it is not possible to compute the expectation of the error and then to optimize the network weights and biases to minimize the loss L_1 . A possible alternative solution is to take a batch of examples from the dataset, the training set, and to compute the gradient of the error, $\nabla_{\theta_i} L$, between the network output and the ground truth on this data with respect to each network parameter θ_i . After computing the gradient, it is possible to update each network parameter in order to reduce the error on this batch as

$$\theta_i = \theta_i - \lambda \cdot \nabla_{\theta_i} L \quad (4.16)$$

where λ is the learning rate and it manages the magnitude of the update at every training step. This is the so called *Stochastic Gradient Descent* (SGD). Different, more elaborated techniques to update the network parameters are proposed in literature, and among these the ADAM algorithm [64] is one of the best performing. ADAM will be the technique used for the optimization of the CNNs in the next of this chapter. The optimization process is repeated for all the batches in the training set. The term *epoch* is used to refer to one complete usage of the dataset. Usually, a neural network is trained for several epochs before reaching a stabilization of the error.

To define the architecture and the required number of layers of the ANN, a second dataset, different from the training set, is used to evaluate if the trained network is able to generalize to unseen data. This dataset is named *validation set* and the validation error can be used to interpret if the network has sufficient capacity or if it is too complex and so it over-fits on the training set.

After this short introduction to ANNs and CNNs, the next of this chapter will focus on how CNNs can be used to refine ToF data.

4.3.2 R-CNN+B: MPI Estimation and Noise Filtering

In the method R-CNN+B, but also in TD-CNN, the idea is to use a CNN, whose input is composed by features extracted from a MF-ToF camera, in order to exploit the *frequency diversity* of the MPI phenomenon to evaluate if the MPI is acting and in case what is the amount of depth distortion due to it. Indeed, by recalling Eq. 2.10, where the ToF correlation function in case of MPI is formulated, the values of the corrupted ToF phase ϕ_{FF} and amplitude A_{FF} change when the modulation frequency changes unless no MPI is acting. Unfortunately, no close-form solution exists to estimate the correct phase (ϕ_d) given the ToF acquisition at different frequencies. For this reason, we proposed to use a CNN as a predictor to try to recover ϕ_d .

Here, we used data from a ToF camera that captures the scene using the modulation frequencies set to 20, 50 and 60 MHz. We extracted some features that are meaningful for MPI analysis directly exploiting the *frequency diversity* on the acquired depth and amplitude images. Moreover, we designed the proposed CNNs to exploit also the information about the geometry of the scene to estimate the MPI, as done in some approaches using single frequency data as [30, 47, 48].

Regarding R-CNN+B, the general architecture of the depth refinement method is shown in Fig. 4.12.

The data acquired by the MF-ToF system are first pre-processed in order to extract a representation that contains relevant information about the MPI presence and strength. As detailed in the next section, where also the motivation for the selection of each input source is presented, the deep network has five different input channels containing the ToF depth extracted from the phase at 60 MHz, the difference between the depth maps at different frequencies and the ratio of the amplitudes also at different frequencies.

The employed CNN architecture, R-CNN, is made of two main blocks, a coarse network that takes in input the five representations and estimates the MPI at low resolution and a fine network that takes in input the five representations and the

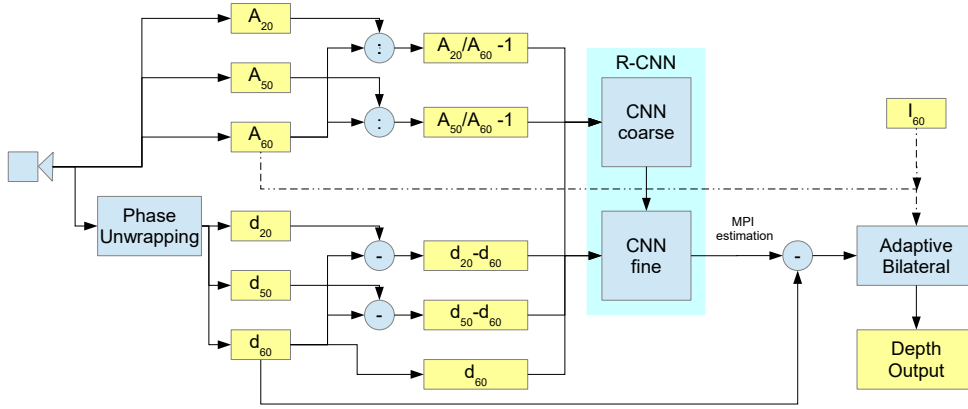


Fig. 4.12: Architecture of the R-CNN+B depth refinement method.

output of the coarse network in order to estimate the MPI interference at full resolution. The estimated multi-path error is then directly subtracted from the ToF depth map (at 60 MHz), thus obtaining a depth map free from MPI distortion (but still affected by other zero-mean error sources).

The resulting depth map is first filtered with a 3×3 median filter in order to remove depth outliers, then the final output of the proposed method is obtained by further filtering it with an adaptive version of the bilateral filter [65] because of its capability of reducing noise while preserving edges. Bilateral filters have been already used on ToF data [66, 67], specially to denoise and upsample the depth map using information from a standard video camera. In our implementation the bilateral filter is guided by the noise information estimated from the received signal amplitude and intensity from which the error variance related to shot noise can be estimated. As suggested in [68], we fixed the spatial smoothing parameter σ_d to a constant value, while the range parameter σ_r is taken proportional to the level of noise. We made the bilateral filter adaptive by using a per pixel noise model for σ_r . In particular we took $\sigma_r = c_r \cdot \sigma_n$, where σ_n is an estimate of the depth noise standard deviation due to shot noise by means of Eq. 2.9. We optimized the values of σ_d and c_r on a subset of the synthetic training dataset. Then, we used the selected values ($\sigma_d = 3$ and $c_r = 3.5$) in the evaluation phase.

ToF Data Representation

As mentioned before, we used a CNN to estimate the MPI corruption on the ToF depth map at 60 MHz that is phase unwrapped by using the 20 MHz and 50 MHz ToF data. Notice that these frequency values have been selected since they resemble the ones used in real world ToF cameras. We also investigated the possibility of performing the phase unwrapping using the proposed CNN (introduced in the next section), but the disambiguation using the MF data proved to be reliable and the deep network optimization is more stable if already phase unwrapped data is fed to it. A critical aspect is the selection of input data that should be informative about the MPI phenomenon. We decided to use as input the following elements:

- The first input $C_1 = d_{60}$ is the ToF depth map at 60 *MHz*. It is required not only because it is the corrupted input that needs to be denoised but also because the geometry of the scene influences the MPI error and the ToF depth represents the best estimate of the geometry available before the MPI removal process. We selected the depth captured at 60 MHz since the higher the modulation frequency, the more accurate the depth estimation.
- The difference between the depth maps estimated at the different modulation frequencies, used since the MPI corruption changes with the frequency (generally the higher the modulation frequency, the smaller is MPI [69]). We used the differences between the depths at 20*Mhz* and 60*Mhz*, and between the ones at 50*Mhz* and 60*Mhz*, i.e., $C_2 = d_{20} - d_{60}$ and $C_3 = d_{50} - d_{60}$.
- The ratio of the amplitudes of the received light signal at different modulation frequencies. In presence of MPI the light waves experiences destructive interferences and in ToF data acquired in presence of MPI the higher the modulation frequency, the lower the resulting amplitude. For this reason, comparing the amplitudes at different frequencies gives us a hint about the MPI presence and strength. We used the ratios between the amplitudes at 20*Mhz* and 60*Mhz*, and between the ones at 50*Mhz* and 60*Mhz*, i.e., $C_4 = (A_{20}/A_{60}) - 1$ and $C_5 = (A_{50}/A_{60}) - 1$. We decided to use the ratio between the amplitudes since in this way it is possible to cancel out the gain of the sensor, that can be different for different sensors, making the method more robust to hardware

changes. The “ -1 ” term has been introduced to center the data around 0 in case of MPI absence.

The proposed CNN, R-CNN, aims at estimating the MPI corruption on the 60 MHz depth map: the targets for the training procedure have been computed by taking a filtered version of the difference between d_{60} and the ground truth depth d_{GT} (the filtering is used to remove the zero mean error, notice that MPI is a low frequency noise). We decided to use this set of inputs for the proposed *Coarse-Fine CNN* since depth and amplitude are data which are generally accessible from commercial ToF cameras. We have tried to use subsets of the input data, but this reduced the performance in MPI estimation. Notice that other techniques based on multi-frequency approaches as [34, 54] use a per pixel model based on the sparsity of the backscattering vector, that is the vector containing the arrival time of each modulated light ray, while in our proposal we are implementing a data driven model that will suit the diffuse reflection case and thanks to the CNN receptive fields we are capturing the geometrical structure of the scene in addition to the *frequency diversity*. We decided to pre-filter the CNN inputs with a 5×5 median filter to obtain a more stable input and reduce their zero-mean variation.

Proposed Deep Learning Architecture

The architecture of the proposed *Coarse-Fine CNN* is shown in Fig. 4.13: the network is made of two main parts, a coarse sub-network and a fine one.

Since the MPI phenomenon depends on reflections happening in different locations, a proper estimation of its presence needs a relatively wide receptive field of the CNN in order to understand the geometrical structure of the scene. Following this rationale, the coarse network performs an analysis of the input data by applying downsampling with pooling layers increasing the receptive field as a consequence. The coarse network takes in input the five data channels described in the previous section and it is made of a stack of five convolutional layers each followed by a ReLU with the exception of the last one. The first two convolutional layers are also followed by a max-pooling stage reducing the resolution of a factor of 2. All the layers perform 3×3 pixels convolutions and have 32 filters, except the last one that has a single filter, producing as output a low resolution estimate of the

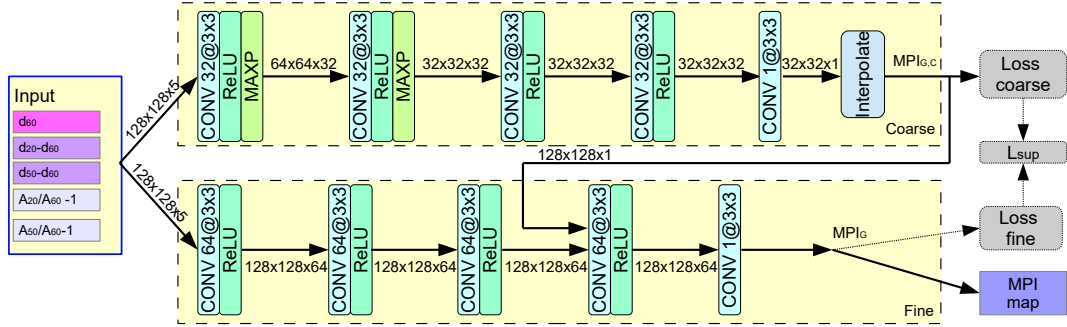


Fig. 4.13: Architecture of the Coarse-Fine CNN used for MPI estimation in the R-CNN+B method.

MPI. The estimated MPI error is finally upsampled of a factor of 4 using a bilinear interpolation in order to bring it back to the original input resolution. This network allows us to obtain a reliable estimate of the regions affected by MPI but, mostly due to the pooling operations, the localization of the interference is not precise and directly subtracting the output of this network to the acquired data would lead to artifacts specially in proximity of the edges. For this reason, we used a second network working at full resolution to obtain a more precise localization of the error. This second network also has five convolutional layers with 3×3 convolutions and ReLU activation functions (except the last as before). It has instead 64 filters for each layer and no pooling blocks. The input of the first layer is the same of the previous network but the fourth layer takes as input not only the output of the third layer but also the upsampled output of the coarse network. This allows us to combine the low resolution estimation with a wide receptive field of the previous network with the more detailed but local estimation done by the fine network and to obtain an MPI estimation that captures both the scene global structure and the fine details.

The network has been trained using the synthetic dataset $S_{1,train}$ of Section 3.3. Even if it is one of the largest ToF dataset with multi-frequency data and ground truth information, its size is still quite small if compared to datasets typically used for CNNs training. In order to deal with this issue and avoid over-fitting we applied data augmentation techniques on the training data as random sampling of patches,

rotation and flipping operations. We extracted 10 random patches of size 128×128 pxl from each of the 40 scenes, then we applied to each of them a rotation of ± 5 degrees and horizontal and vertical flipping.

This leads to a total of about $40 \times 10 \times 5 = 2000$ patches (invalid patches with non complete covering on rotated images have been excluded), that represents a good amount of data for the training of the proposed deep network. The number of patches could be increased by using smaller patches, but this would weaken the ability of the network to understand the geometrical structures of the scenes and to retrieve the MPI corruption.

Due to the small amount of data we have used *K-fold cross-validation* with $K=5$ on the training set to validate the hyper-parameters of the CNN and of the training procedure as the architecture of the network, the number and depth of the layers, the learning rate and the regularization constant. We have divided the 40 scenes of the training set into 5 folds of 8 scenes each, then we have selected one fold as validation set and used the remaining for the training and repeated this procedure for each of the folds. We have selected the CNN hyper-parameters in order to avoid overfitting and obtain the minimum mean validation MAE among the 5 folds. Once the hyper-parameters have been selected, the CNN has been trained on the whole training set.

For the training we minimized a combined loss made by the sum of two loss functions, one computed on the interpolated output of the coarse network and the other computed on the output of the fine network. This approach allowed to obtain better performance than the separate training of the two sub-networks. Each of the two loss functions is the l_1 norm of the difference between the MPI error estimated by the corresponding network and the MPI error computed by comparing the ToF depth at 60 MHz with true depth as described in Section 4.3.2. The l_1 norm is more robust to outliers in the training process if compared with the l_2 norm and had more stable results in the validation of the network hyper-parameters. Furthermore the use of l_1 norm proved to be more efficient for image denoising [70]. During the training, we exploited the *ADAM* optimizer [64] and a batch size of 16. We started the training with an initial set of weight values derived with Xavier's procedure [71], a learning rate of 10^{-4} and a l_2 regularization with a weighting factor of 10^{-4} for the norm of the CNN weights. Fig. 4.14 shows the mean training and validation

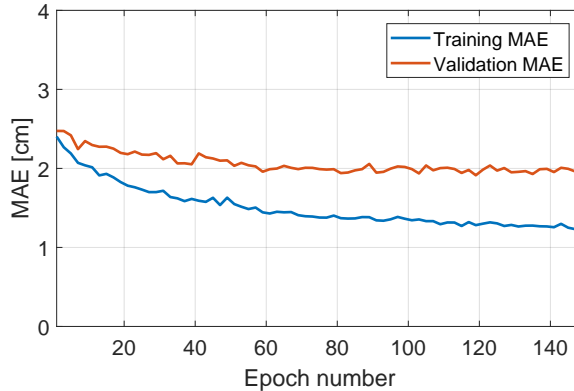


Fig. 4.14: Mean training (blue) and validation (red) error at each epoch of the R-CNN.

error across all the epochs of the *K-fold cross-validation*: we trained the network for 150 epochs, that in our case proved to be enough for the validation error to stabilize. As mentioned above, we used *K-fold cross-validation* in order to have a more informative metric to evaluate the real performance of the network and to exclude over-fitting. The network has been implemented using the *TensorFlow* framework and the training took about 30 minutes on a desktop PC with an Intel i7-4790 CPU and an *NVIDIA Titan X (Pascal)* GPU. The evaluation of a single frame with the proposed network takes instead just *9.5ms*.

4.3.3 TD-CNN: Depth Denoiser CNN

This section introduces the TD-CNN method able to reduce the noise and the MPI distortion. Differently from R-CNN+B, the denoiser CNN used in this method tries to combine the two refinement steps together, without requiring the application of the *adaptive bilateral filter*. Many aspects of the used CNN are similar to the one presented in the previous sections.

First of all, the CNN input features are extracted from the same multi-frequency ToF data captured at 20, 50 and 60 MHz. These are $C_1 = d_{60}$, $C_2 = d_{20} - d_{60}$, $C_3 = d_{50} - d_{60}$, $C_4 = (A_{20}/A_{60}) - 1$ and $C_5 = (A_{50}/A_{60}) - 1$. But in this case, these features are not pre-filtered and they have been used to feed the CNN as they are. Indeed, here the task is not to estimate the low-frequency MPI depth corruption,

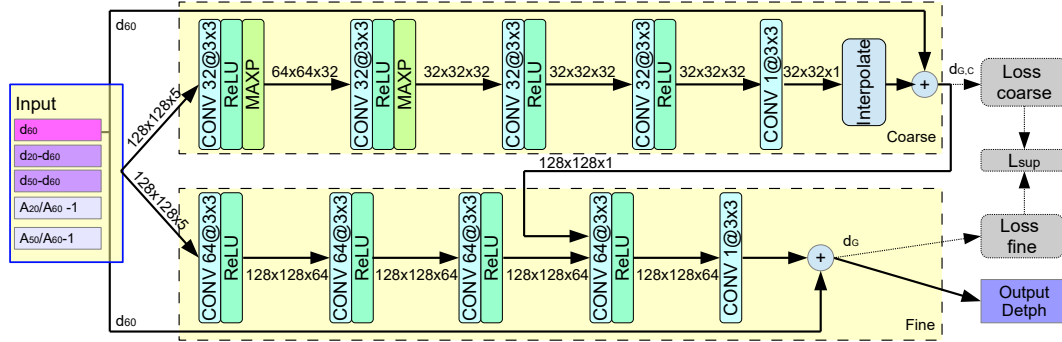


Fig. 4.15: Architecture of the Coarse-Fine CNN used in the TD-CNN method.

for which small local variation are not important, but to directly recover to refined depth map. The same considerations about the meaningfulness of these features w.r.t. the MPI phenomenon hold.

Regarding the CNN architecture used in TD-CNN, it has the same Coarse-Fine structure used in the previous method. It uses the aforementioned not-filtered input in order to output the refined scene depth map. Fig. 4.15 shows the employed CNN architecture.

However, in TD-CNN the input ToF depth map acquired at 60 MHz (d_{60}) the depth we want to denoise, is directly summed to the output of the CNN branches. In this way, the CNN layers have to internally estimate the error map, as in the previous method, which is summed to the noisy depth map in order to correct it. Apart from this, the two Coarse-Fine CNN layer structures are identical.

Also in this case, the CNN is trained on the $S_{1,train}$ dataset and the mentioned data augmentation techniques have been used during the training. For the training, we minimized a combined loss composed by the sum of two loss functions, one computed on the interpolated output of the coarse network and the other computed on the output of the fine network. Each of the two loss functions is the l_1 norm of the difference between the depth map estimated by the corresponding network and the true scene depth. We started the training with an initial set of weight values derived with Xavier's procedure [71], a learning rate of 10^{-4} and a l_2 regularization with a weighting factor of 10^{-4} for the norm of the CNN weights. These network hyper-parameters have been validated by means of *5-fold cross-validation* and the

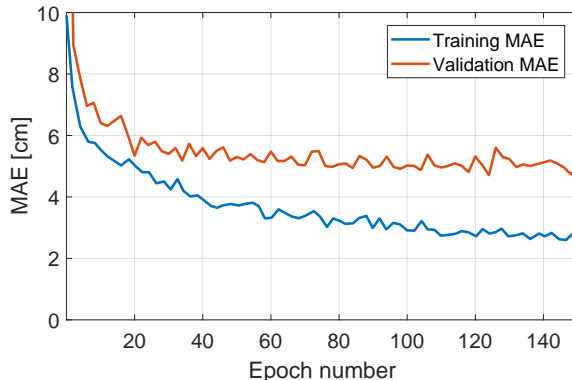


Fig. 4.16: Mean training (blue) and validation (red) error at each epoch of the TD-CNN.

mean validation and training error, related to this validation, are shown in Fig. 4.16. We trained the network for 150 epochs. The network has been implemented using the *TensorFlow* framework and the training took about 30 minutes on a desktop PC with an Intel i7-4790 CPU and an *NVIDIA Titan X (Pascal)* GPU.

Both the training and validation error are higher than the ones reported for the R-CNN. This happens because in this case also the zero-mean error related to thermal and photon shot noise are taken in account and no pre-denoising of the input and of the output is applied.

4.3.4 Training and Test Datasets

This section describes the training and test datasets used for the methods evaluation. It is complex and time consuming to collect a real world dataset big enough for CNN training with ToF data and the related depth ground truth. For this reason, we decided to exploit a dataset composed by synthetic scenes, for which the true depth is known. In particular, we used the S_1 dataset introduced in Section 3.3. The ToF acquisitions have been performed with the *Sony ToF Explorer* simulator realized by Sony Eutec able to faithfully reproduce ToF acquisition issues like the shot and thermal noise, the read-out noise, artifacts due to lens effects, mixed pixels and specially the multi-path interference. The S_1 dataset is split in the training set $S_{1,train}$, composed by 40 scenes, and the test set $S_{1,test}$, composed by 14 different scenes. Each scene has been rendered from a virtual viewpoint with the ToF simu-

lator in order to acquire the ToF raw data (amplitude, intensity and depth image) at the modulation frequencies of 20, 50 and 60 MHz. The scene depth ground truth is available for each scene ToF simulation.

The ToF denoising methods have been also tested on the ToF real dataset S_4 . As mentioned in Section 3.2.1, this dataset is composed by 8 scenes and is provided with the ToF acquisitions at 20, 50 and 60 MHz and the related depth ground truth. The S_4 dataset has been used for testing purposes only, since they are too small to be used for training.

More details about the employed datasets have been presented in Chapter 3.

4.3.5 Results of the Proposed Data Driven Methods

In order to evaluate the R-CNN+B and the TD-CNN methods, we used the two different test sets presented in Section 4.3.4. The evaluation results related to the synthetic dataset $S_{1,test}$ are presented in the next section. Then, the results on the employed real world dataset S_4 will be discussed. Due to the size of this dataset, it has been used for evaluation purposes only.

Results on Synthetic Data

Fig. 4.17 shows the results of the application of the proposed methods on a subset of the scenes extracted from $S_{1,test}$ and used for testing. It shows the input depth map from the ToF camera at 60 MHz (with phase unwrapping), the depth map after the application of the adaptive bilateral filter (ABF) and the final results of the R-CNN+B and the TD-CNN methods with their related errors maps and the depth ground truth information. By looking at the fourth and fifth rows it is possible to notice how the adaptive bilateral filter is able to reduce the zero-mean error by preserving the fine details in the scenes, e.g., the small moon in the *castle* is preserved by the filtering process, but the depth overestimation due to MPI is still present. From the sixth and seventh rows, it is possible to see how both the multi-path error and the zero-mean noise have been widely reduced by the complete version of R-CNN+B. For example in the first three scenes there is a very strong multi-path distortion on the walls in the back that has been almost completely removed by the proposed approach for MPI correction. The multi-path estimation

is very accurate on all the main surfaces of the scenes, even if the task proved to be more challenging on some small details like the top of the pots in columns 1 or the stairs in column 2. However, notice that thanks to the usage of the Coarse-Fine network the small details of the various scenes are preserved and there is no blurring of the edges. This can be seen for example another time from the details of the castle (e.g., the moon shape) in column 3. The box scene (column 4) is another example of the MPI removal capabilities. Notice how the multi-path on the edges between the floor and the walls is correctly removed. Also the error on the slope in the middle of the box (that is more challenging due to bounces from locations farther away) is greatly reduced even if not completely removed. The eighth and ninth rows show the output of TD-CNN and the related error map. The MPI correction performance is very similar to R-CNN+B. The main differences are in the managing of the noise level. Since TD-CNN is trained having in input not filtered data and it is trained to directly estimate the depth ground truth, it seems to slightly better recover the very noisy regions as the floor in the scene contained in column 2. This evaluation is confirmed also by numerical results, the Mean Absolute Error (MAE) is reduced from 156 mm on the input data to 74.9 mm using R-CNN+B and to 62.1 mm using TD-CNN.

Fig. 4.18 shows the impact of the various components of the Coarse-Fine CNN architecture, used in R-CNN+B, on the *castle* and *stairs* scenes. The first column shows ToF error at 60 MHz. The second column shows the estimation taken from the interpolated output of the coarse network: notice how the general distribution of the MPI is correctly estimated but the edges and details (e.g., the moon over the castle) are lost in this estimation due to the pooling operations that reduce the resolution. The last column shows instead the output of the Coarse-Fine architecture and it is possible to notice how the general MPI distribution is maintained but there is a much higher precision on boundaries and small details. The second row shows the same data for the *stairs* scene, also in this case notice how the general structure is the same but the estimation follows more accurately the shape of the stairs in the Coarse-Fine output.

We compared the proposed methods with some competing approaches from the literature. In particular, we considered the MF-ToF MPI correction scheme proposed by Freedman [34] and the method based on deep learning presented by

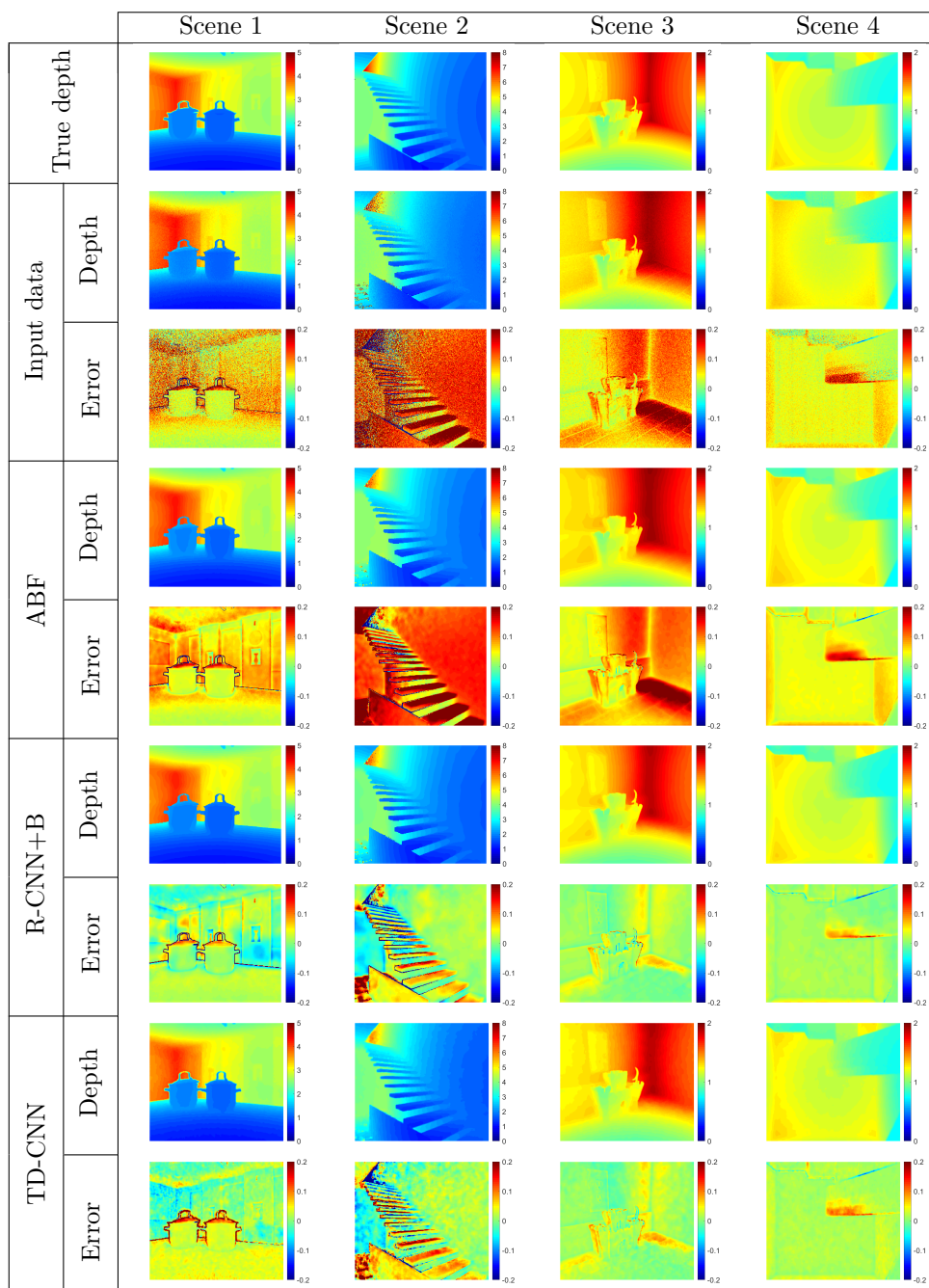


Fig. 4.17: Input depth map at 60 MHz, output of the adaptive bilateral filter (BF) and output of the proposed approach (with MPI correction) on same sample synthetic scenes with the corresponding error maps. All the values are measured in meters.

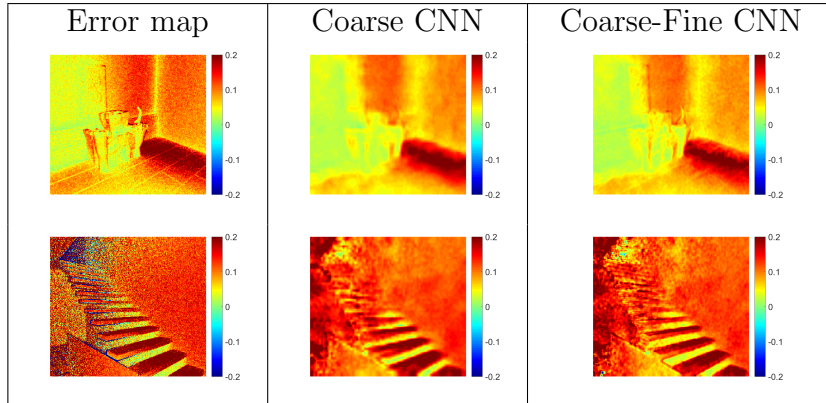


Fig. 4.18: Estimation of the MPI performed by the proposed approach using only the coarse network or the complete Coarse-Fine architecture.

Marco in [30] that takes in input the depth map at 20 MHz to remove MPI. The method proposed by Freedman was adapted to use the same triple of frequencies used by the proposed approaches. The first column of Table 4.2 shows the MAE obtained by comparing the output of the four methods with the ground truth data on the synthetic dataset. R-CNN+B is able to reduce the error from 156 to 74.9 mm, reducing it to less than half of the original error. The TD-CNN is able to do even better reducing the error to 62.1 mm. It also outperforms with a wide margin the Freedman’s and Marco’s methods. The Freedman method [34] is able to remove only about 10% of the error in the source data obtaining an accuracy of 140 mm. The method of [34] works under the hypothesis that the light backscattering vector is sparse and this is not true in scenes where diffuse reflections are predominant as the considered ones. For this reason, its effectiveness is limited. The method of [30] works under the assumption that the reflections are diffuse and it achieves better results removing about 20% of the original error, but it is still far from the performance of the proposed approaches. This is due to the fact that the CNN proposed in [30] uses single frequency ToF data, instead we showed that a multi-frequency approach can achieve much higher performance using a less complex CNN.

	Synthetic data		Real World data	
	MAE [mm]	Relative Err. [%]	MAE [mm]	Relative Err. [%]
ToF input (60 Mhz)	167.3	-	54.3	-
ToF input (20 Mhz)	327.8	-	72.8	-
Freedman et Al. [34]	149.8	89.5%	51.1	94.1%
Marco et Al. [30]	260.9*	79.6%	51.3*	70.5%
R-CNN+B	74.9	44.8%	31.9	58.7%
TD-CNN	62.1	37.1%	31.3	57.6%

Table 4.2: Mean MAE for competing schemes from the literature and for the proposed approach on synthetic and real world data. The table shows the MAE in millimeters and the relative error between the output of the various methods and the error on input data. Our approach and [34] are multi-frequency methods and are compared with the highest employed frequency (60 MHz) for the relative error, instead [30] (*) is compared with the only frequency it uses (20 MHz).

Results on Real World Data

After evaluating the proposed approach on synthetic data, we performed also some experiments on real world data. For this evaluation we used the real test set introduced in Section 4.3.4, S_4 , that is composed by 8 scenes. It has a more limited variety of settings with respect to the synthetic data but still the scenes contain objects of different sizes, types of material and surfaces with different orientations where the MPI can arise. The Coarse-Fine CNNs were trained on the synthetic dataset that is composed by scenes with ideal properties, e.g., the reflections are perfectly diffuse, and due to some limitations of the simulator, the synthetic data, even if quite accurate, does not exactly model all the issues of real data.

Fig. 4.19 shows the results of the application of the proposed approaches to the set of real world scenes. As before, it shows the input depth map from the ToF camera at 60 MHz and the depth map resulting after the application of the considered methods with their corresponding error maps and ground truth information. By looking at the images, R-CNN+B and TD-CNN share similar performance in reducing MPI. However, a noticeable amount of MPI error remains in the scenes. It is possible to notice how the MPI is almost completely removed on the vertical walls, in particular in proximity of edges between facing surfaces. The reduction is

strong also on the small objects like the sphere, the cone or the deer even if some multi-path in proximity of boundaries remains on these objects. On the other hand the MPI error is under-estimated on surfaces with a strong inclination, in particular the floor in the various scenes, where the approach is able to reduce only part of the multi-path. By comparing Fig. 4.17 and Fig. 4.19 it is possible to notice how the strong MPI on these surfaces (e.g., the floors) is not present in the synthetic scenes. This is probably due to the fact that reflections happening on the considered real materials are not ideally diffuse when the light rays are strongly inclined and the ToF simulator does not model this phenomenon. More in general, this is a problem of *domain shift*, that is happening when a network is trained on a domain, the synthetic dataset, and tested on a different domain, the real dataset. Our approach, as any other machine learning scheme, learns from the training data and is not able to correct issues not present in the training examples. R-CNN+B and TD-CNN have a different outcome related to the zero-mean error, with the first method that is able to better smooth the flat surfaces.

We compared R-CNN+B and TD-CNN with [34] and [30] also on the real world data. The results are in the third and fourth column of Table 4.2. On real data, R-CNN+B is able to reduce the error from 54.3 to 31.9 mm, i.e., to 58.7% of the original error, and TD-CNN is able to reduce it to 31.3 mm. These are very good performance outperforming both the compared approaches from literature even if with a smaller gap than the one achieved on synthetic data. In particular, the proposed methods were able to improve the accuracy of the depth estimation on all the considered scenes: in the worst case scene the error was reduced to about the 70% of the initial error. Recall that the training is done on synthetic information only, as pointed out in the visual evaluation the issues on the floor reduce the performance. This is the quantification of the effect of the *domain shift*. The error removal capability of [34] is limited also in this case, it removes about 5.9% of the error. The method of [30] removes about 30% of the error and gets a bit closer to ours in this experiment, but there is still a gap of more than 10%.

The next chapter will focus on the *domain shift* issue arising when a network is trained on a domain and tested on a different one. In particular, the first part will present an *unsupervised domain adaptation* method to reduce this issue for TD-CNN on real data, even if no real ground truth is exploited.

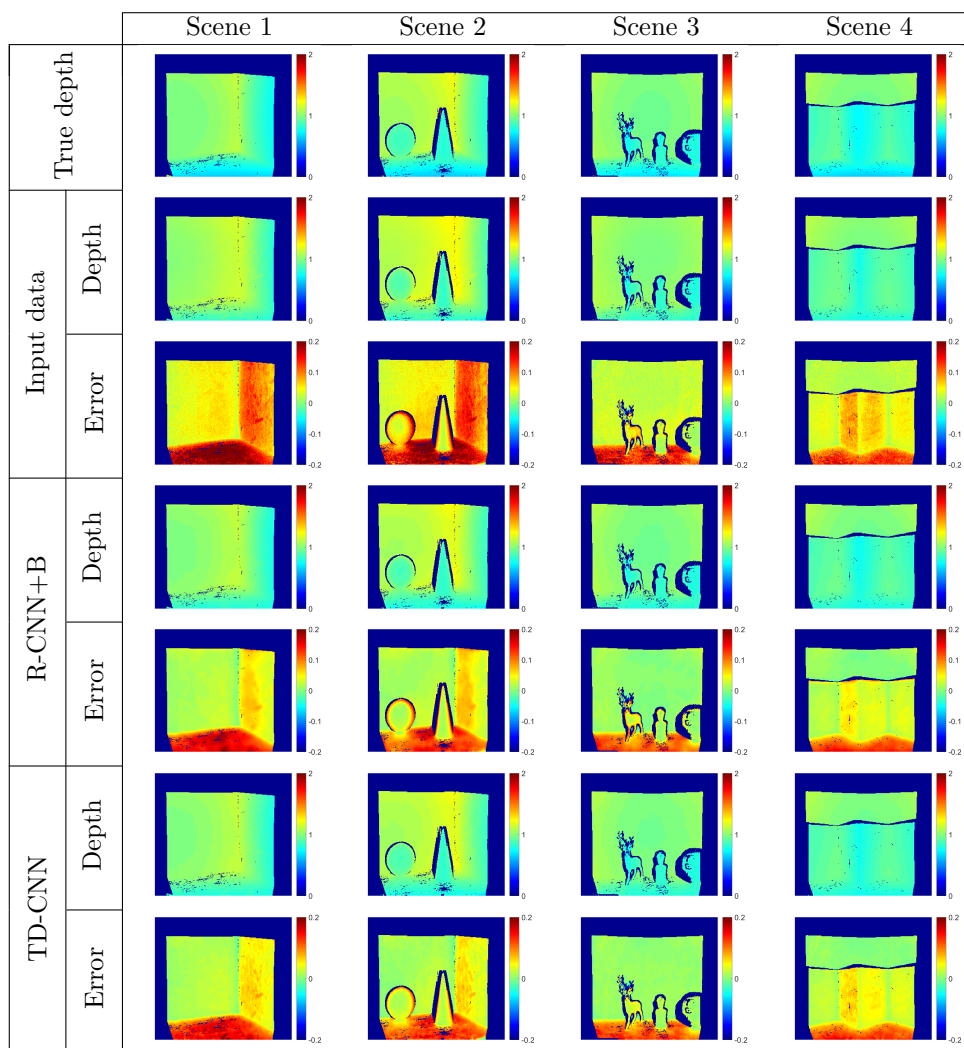


Fig. 4.19: Input depth map at 60 MHz and output of the proposed approach on same sample real world scenes with the corresponding error maps.

Chapter 5

Domain Adaptation

This chapter deals with the *domain shift* issue. This is a problem arising in machine learning when a predictor is trained on a source domain and tested on a target domain that is different from the source one. In a scenario like this, the performance of the trained predictor decays in the target domain due to the intrinsic differences between the two domain statistics. This problem is present regardless what is the task of the predictor. A common case in which *domain shift* can be experienced is when a simulator is used to generate the training set because it is complex to collect couples of real input data and the related ground truth. In this situation a synthetic dataset is the source domain and a real dataset is the target domain. In the previous chapter, we already encountered this setting in case of ToF depth data refinement.

The effects of *domain shift* can be fought by domain adaptation methods which try to improve the performance of the predictor on the target domain. The next section will present an overview about domain adaptation methods. Then, a domain adaptation method for ToF depth data refinement will be introduced and its performance evaluated. Another domain adaptation method for semantic segmentation will be analysed in the last part of this chapter.

5.1 Domain Adaptation for Neural Networks

Domain adaptation is a growing research area and this chapter will focus on the case when it is applied in unsupervised way. A widely explored field for unsupervised domain adaptation is the classification task. Domain adaptation is usually applied by trying to reduce the differences between the statistic of the neural network internal features on the source data and on the target data. Different approaches have been proposed to implement this idea. Some focus on reducing the first order [72, 73] and the second order [74, 75] statistic discrepancy of the network internal features. Also variants of the *batch normalization* layers have been used to align the features in the two domains [76–78]. Another interesting approach involves the use of a *domain classifier* in order to understand if the intermediate network features are coming from the source or the target domain [79, 80]. This domain classifier is then used to implement an adversarial loss, similar to the one used in generative adversarial networks (GANs) [81], to align the two data statistics. A different approach involves the use of GANs to create new samples from the target domain, with related label, and to train the task classifier exploiting these new “fake” images to apply the unsupervised domain adaptation [82–84].

The ideas exploited in case of unsupervised domain adaptation for classifiers have been reformulated and opportunely adapted to be applied in different learning fields as semantic segmentation of color images and regression tasks. In particular, unsupervised domain adaptation for semantic segmentation is acquiring attention from the research community. Many semantic segmentation networks have been proposed (see [85] for a recent review of the field). These show impressive performance but they all share the fundamental issue that a large amount of labeled data is needed for their training. They are typically trained on huge datasets with pixel-wise annotations, e.g., the Cityscapes [86], CamVid [87] or Mapillary [88], whose acquisition is highly expensive and time consuming. Recent research focuses on how to deal with this issue by adapting the training done on a different set of data with slightly different statistics to the problem of interest. A common setting for this task is domain adaptation from synthetic data to real world scenes. The development of advanced computer graphics techniques enabled to collect huge synthetic datasets for semantic segmentation purposes. Examples of synthetic semantic segmentation

datasets for the autonomous driving scenario are the GTA5 [89] and SYNTHIA [90] datasets, which have been employed in part of our work presented in Section 5.4.1. One of the first works to deal with cross-domain semantic segmentation is [91], where the adaptation is performed by aligning the network internal features from the different domains during the proposed adversarial training procedure. A similar idea is exploited in [92], where the feature alignment is obtained using a generative model built on GANs. A curriculum-style learning approach is proposed in [93], where firstly the easier task of estimating global label distributions is learned and then the segmentation network is trained forcing that the target label distribution is aligned to the previously computed properties. Other approaches try to solve the *domain shift* by translating the input synthetic data, whose pixel level semantic map is known, in real data. An example is CyCADA [94] that uses a cycle consistency to ensure that the semantic map of the scene is not corrupted in the translation process. CrDoCo [95] is a recent improvement of the aforementioned domain translation method. A different strategy is to align the output space of the network [96,97]. Other approaches apply unsupervised adaptation for semantic segmentation exploiting distillation loss [98] and entropy minimization [99].

Unsupervised domain adaptation for regression tasks is a less investigated field. However, recent works have focused the attention to apply domain adaptation for monocular depth estimation as in [100], where multi-task learning is exploited to align the internal features of the network in the source and target domain. In [101], feature alignment is implemented by means of an adversarial loss to adapt a network trained for multi-task regression (normal, edge and depth from color image) from synthetic to real data. An application of domain adaptation for stereo vision depth estimation is presented in [102], where a self training guided by traditional stereo vision algorithms is employed.

5.2 Introduction to Generative Adversarial Networks

Before introducing the domain adaptation techniques developed during my Ph.D. work, it is worth to introduce the fundamental working principles of Generative

Adversarial Networks (GANs). These systems exploit an *adversarial loss* to implement an unsupervised training, and this typology of loss will be exploited by the domain adaptation techniques discussed in the next of this chapter.

GANs were initially introduced by Goodfellow et al. [81]. They are generative models in which a neural network, called *generator* and here referred to as G , is trained to generate images starting from random noise. The produced images have to follow a predefined statistic, as the example shown in Fig. 5.1. The issue with this task is that a mapping between random noise and target distribution does not exist and so the idea from [81] was to implement a loss using a discriminator network D , this is the so called *adversarial loss*.

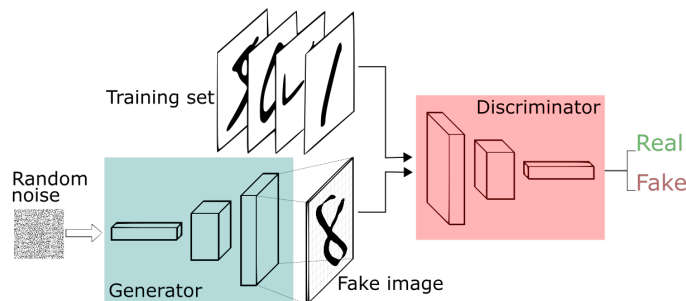


Fig. 5.1: GAN training scheme. [103]

The idea is to train iteratively the generator and the discriminator in an adversarial way. The discriminator is trained to try to understand if the input data is coming from the target statistic (p_{data}) or from the generator statistic ($p_G(z)$, where $z \sim p_{noise}$ is the random noise given as input to G). Differently, G is trained to fool D by creating data belonging to the target statistic p_{data} . This means that G and D are playing a two-player minimax game in which they want to overcome the other one. From a mathematical point of view, the loss functions for the training of D and G are complementary and the optimization problem can be represented as

$$\min_G \max_D E_{x \sim p_{data}} [\log(D(x))] + E_{z \sim p_{noise}} [\log(1 - D(G(z)))]. \quad (5.1)$$

In the equation above, the parameters of D are optimized to let it recognize the data coming from the target distribution as belonging to the class 1 (Real) and the data produced by G as belonging to the class 0 (Fake). Differently, G is trained

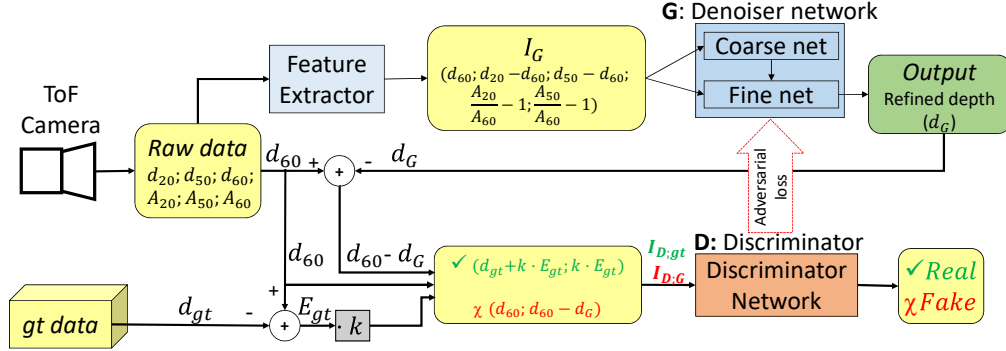


Fig. 5.2: Architecture of the proposed approach.

to create data that are recognized by D as belonging to the class 1 (the target distribution). As showed in [81], this optimization process reaches the stability in a saddle point, when the data produced by the generator are not distinguishable by the data sampled by the target distribution and so reaching the final goal of the generator G .

The *adversarial losses* are very versatile, and they can be used when it is not possible to access to the ground truth data and so the supervised training is not possible. For this reason, the *adversarial loss* can be exploited in the domain adaptation tasks which are presented in the next of this chapter.

5.3 Domain Adaptation for ToF Data Refinement

The previous chapter has shown how CNNs can be used to refine ToF depth data in a really efficient way. However, due to the limited availability of real ToF data with related depth ground truth, synthetic data have been used for the network training. This choice showed the intrinsic limitation of training a network on a domain, a synthetic dataset, and testing it on a different domain, a real dataset. Here, a method for unsupervised domain adaptation for the ToF data refinement task is presented. Our goal is to improve the performance of the denoiser network on real data even if no real ground truth is accessible during the training.

5.3.1 Proposed Method

For our work, we started from TD-CNN, that is the ToF depth data denoiser CNN introduced in Section 4.3.3. In order to improve the performance of TD-CNN on real data, without using real depth ground truth, we fitted it into a novel adversarial learning framework, used to perform unsupervised domain adaptation from synthetic to real data.

The general architecture of the proposed adversarial learning strategy is shown in Fig. 5.2. The generator (G) of the adversarial learning framework is implemented by means of the *Coarse-Fine* CNN TD-CNN. It takes different features extracted from the sensor raw data as input and produces an estimate of the noise-free depth map of the scene. The discriminator network (D), introduced in Section 5.3.3, is used to capture the statistic of ground depth data and to guide the training of the generator on real ToF data for which the ground truth is not available. The adversarial learning procedure, implemented with the discriminator, will be presented in Section 5.3.4.

The next sections introduce the networks used in the proposed framework and how they are trained to implement an unsupervised domain adaptation.

5.3.2 Generator Network

The generator G is implemented with the TD-CNN described in Section 4.3.3. However, other different denoising networks can be used, but we decided to keep this CNN structure since it has very good performance. As explained in Section 4.3.3, TD-CNN takes in input features extracted from the data acquired by the ToF cameras using the modulation frequencies 20, 50 and 60 MHz. The idea is to exploit the *frequency diversity* of the MPI phenomenon in order to correct it. The acquired information is pre-processed in order to extract a representation I_G that contains relevant information about the MPI presence and strength: five different feature channels have been extracted from the ToF data (see Section 4.3.3), thus obtaining the following input representation:

$$I_G = \left(d_{60}; d_{20} - d_{60}; d_{50} - d_{60}; \frac{A_{20}}{A_{60}} - 1; \frac{A_{50}}{A_{60}} - 1 \right) \quad (5.2)$$

where d_x and A_x are the ToF depth and amplitude maps, captured at x MHz.

The input data I_G is fed to the generator network in the proposed adversarial learning framework. In the next of this chapter, the output of the of the coarse branch will be referred to with $d_{G,C} = G_c(I_g)$ and the output of the final output of the generator as $d_G = G(I_G)$.

5.3.3 Discriminator Network

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 5.3.4, 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 5.3.1, the discriminator takes as input the noisy depth map d_n and the error map E . E 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 $E_G = d_n - d_G$. 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 as 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. 5.3: 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:

$$L_D = -E\left(\log(D(I_{D;gt})) + \log(1 - D(I_{D;G}))\right) \quad (5.3)$$

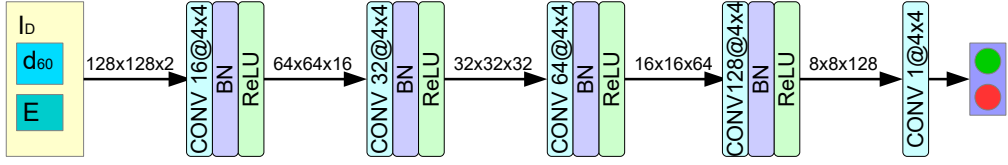
Please note that we are using for the training of the whole system a synthetic dataset provided with the ground truth depth of the scenes (d_{gt}^s) and an unlabeled real dataset. In the rest of this discussion, we will use the “s” and “r” apexes to distinguish between synthetic and real data.

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 5.3.4). 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.

On the other hand, 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 $I_{D;gt}^s$. 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

$$I_{D;gt}^s = (d_{gt}^s + E'_{gt}; E'_{gt}) \quad (5.4)$$


 Fig. 5.3: Architecture of the discriminator network D .

with E'_{gt} given by

$$E'_{gt} = k \cdot (d_{60}^s - d_{gt}^s) = k \cdot E_{gt}^s, \quad (5.5)$$

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.

5.3.4 Adversarial Learning Strategy

The denoiser 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 5.3.5) 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:

$$L_G = L_{sup} + w \cdot L_{adv}, \quad (5.6)$$

where

$$L_{sup} = E[|d_G^s - d_{gt}^s|] + E[|d_{G,C}^s - d_{gt}^s|] \quad (5.7)$$

$$L_{adv} = E[-\log(D(I_{D;G}^r))]. \quad (5.8)$$

The first term is optimized in a supervised way on synthetic data only (dataset S_1 , Section 5.3.5). It is modeled as the sum of the l_1 distances between the outputs of G (i.e., the output $d_G^s = G(I_G^s)$ of the fine network and the output $d_{G,C}^s = G_C(I_G^s)$ 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 5.3.5) without using ground truth information. By minimizing the loss of Eq. 5.8 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 $I_{D;G}^r = (d_n^r; d_n^r - d_G^r)$ (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. 5.3 and Eq. 5.8 follows the LS-GAN structure proposed in [104], 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 (see Section 3.2.1). At first, the synthetic data are used to train the discriminator as mentioned in Section 5.3.3. By following the idea introduced in [83, 105], we exploited a buffer to collect examples of fake data $I_{D;G}^s$, 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. 5.8 and on the synthetic data by minimizing the loss in Eq. 5.7.

The complete learning procedure is summarized in Algorithm 5.1 and Fig. 5.4.

Algorithm 5.1 Domain Adaption Procedure

```

1: procedure TRAINING STEP
2:    $(I_G^s; d_{gt}^s) \leftarrow S_1$  ▷ Get synthetic data
3:    $I_G^r \leftarrow S_2$  ▷ Get real world data
4:    $d_{60}^s \leftarrow I_G^s$  and  $d_{60}^r \leftarrow I_G^r$ 
5:    $E_{gt}^s = d_{60}^s - d_{gt}^s$ 
6:    $k = \text{rand.unif}([1 - \epsilon; 1 + \epsilon])$  ▷ For noise augmentation
7:    $I_{D;gt}^s = (d_{gt}^s + k \cdot E_{gt}^s; k \cdot E_{gt}^s)$ 
8:    $I_{D;G}^s = (d_{60}^s; d_{60}^s - G(I_G^s))$ 
9:    $I_{D;G}^r = (d_{60}^r; d_{60}^r - G(I_G^r))$ 
10:  if  $\text{rand.uniform}([0; 1]) > 0.5$  then
11:     $I_{D;G}^{s, curr} = I_{D;G}^s$ 
12:  else
13:     $I_{D;G}^{s, curr} = \text{queue.get\_sample}()$ 
14:     $\text{queue.push}(I_{D;G}^s)$ 
15:  end if
    ▷ Optimize the discriminator ( $D$ )
16:  minimize  $L_D$  (Eq. 5.3) on  $I_{D;gt}^s$  and  $I_{D;G}^{s, curr}$ 
    ▷ Optimize the generator ( $G$ )
17:  minimize  $L_{sup}$  (Eq. 5.7) on  $(I_G^s; d_{gt}^s)$ 
18:  minimize  $L_{adv}$  (Eq. 5.8) on  $I_{D;G}^r$ 
    
```

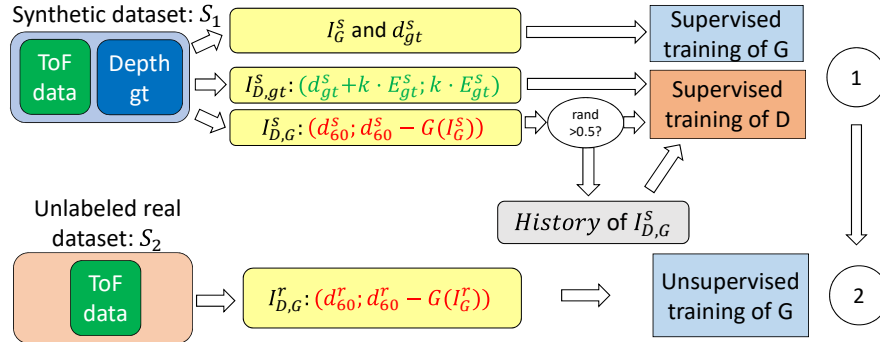


Fig. 5.4: Schematic representation of a training step.

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 3.2.1) after the k-fold cross validation on the synthetic dataset.

We optimized the two neural networks using the TensorFlow framework [106] 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. 5.5 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 5.3.5) 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 per-

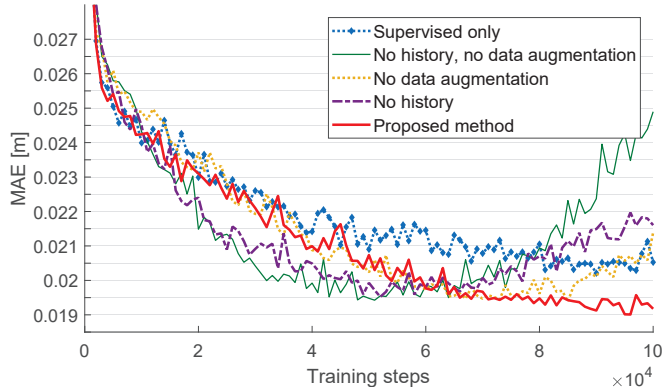


Fig. 5.5: Validation error during the training procedure for different versions of the proposed approach.

formance 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.

5.3.5 Synthetic and Real World Datasets

We exploited five different datasets, introduced in Chapter 3, for the training and evaluation of this work. See Table 5.1 for a summary about their composition and role in the training process.

For the supervised training with synthetic data we used the dataset, $S_{1,train}$, introduced in Section 3.3. This dataset is composed by 40 synthetic scenes with multi-frequency data and ground truth depth. We performed data augmentation by extracting 10 random patches of size 128×128 [pxl] from each scene and by applying rotation and flipping of the patches, as originally done for TD-CNN (Section 4.3.3).

In order to perform the adversarial training procedure, we used the unlabeled real world dataset (S_2), acquired with a Sony DS541 ToF camera.

We used a smaller set of real world scenes for validation purposes. This smaller dataset, S_3 , has only 8 scenes acquired with the DS541 ToF camera, but contains

Dataset	Type	gt	# scenes	Used for
$S_{1,train}$	Synth	Yes	40	Supervised training
S_2	Real	No	97	Adversarial training
S_3	Real	Yes	8	Validation
S_4	Real	Yes	8	Testing
S_5	Real	Yes	8	Testing

Table 5.1: Datasets exploited for the training of the proposed domain adaptation framework.

also ground truth information, acquired with an active stereo matching system as described in Section 3.2.1.

In order to evaluate the performance of the proposed approach, we used the real world datasets S_4 and S_5 (*box dataset*). These datasets contain 8 real world scenes each with ground truth data.

Section 3.2.1 describes the datasets in more details and presents some visual examples of the data.

5.3.6 Experimental Results

The proposed method has been evaluated using the real datasets S_4 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.

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. 5.6 shows the output of the proposed unsupervised domain adaptation 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. The difference of the latter method with respect to TD-CNN, introduced in Section 4.3.3, is that the training hyper-parameters have been set looking at the real validation set. 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. 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 refining 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.

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) [34] and the deep learning based approaches of Marco et al. (DeepToF) [30].

Additionally, we also considered a combination of TD-CNN, used in our domain adaptation model, with the domain adaptation scheme of [79, 101] (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 two domains and reducing the domain shift.

From the quantitative evaluation on the S_4 dataset in Table 5.2, the analytical

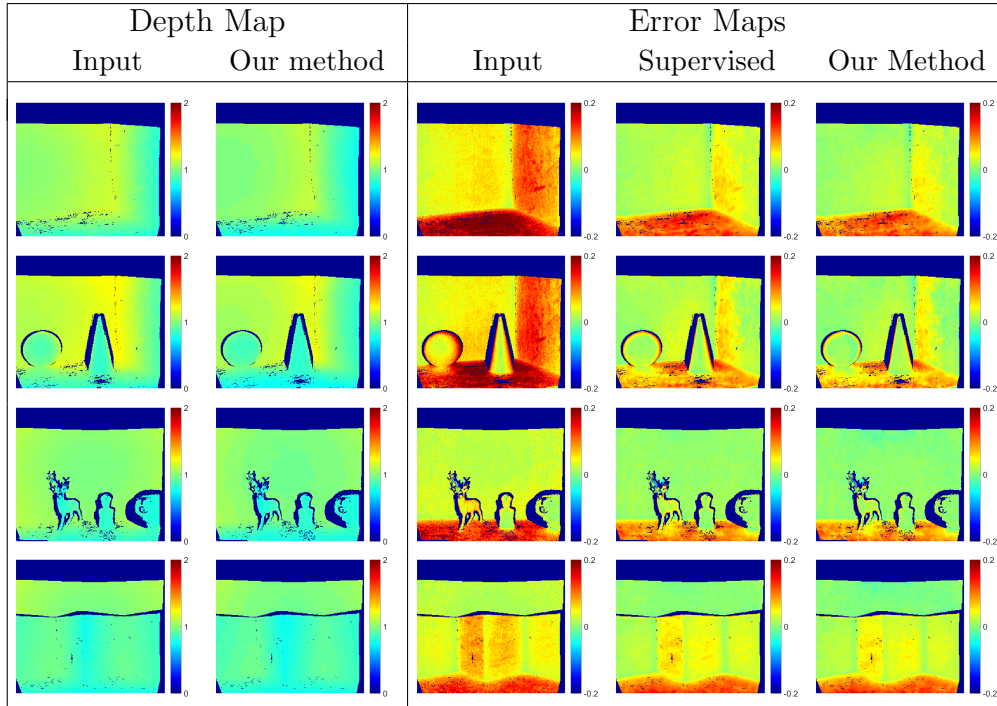


Fig. 5.6: Output of the synthetic supervised and of the proposed domain adaptation approach on some sample scenes from the S_4 dataset. The values are measured in meters.

method of [34] 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 performance: DeepToF [30] 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 TD-CNN, 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 performance (even if lower than the proposed method) and removes about 52% of the error. Note that TD-CNN, sharing the same 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 [34] has roughly the same performance obtained on the other dataset, removing only 7% of the error.

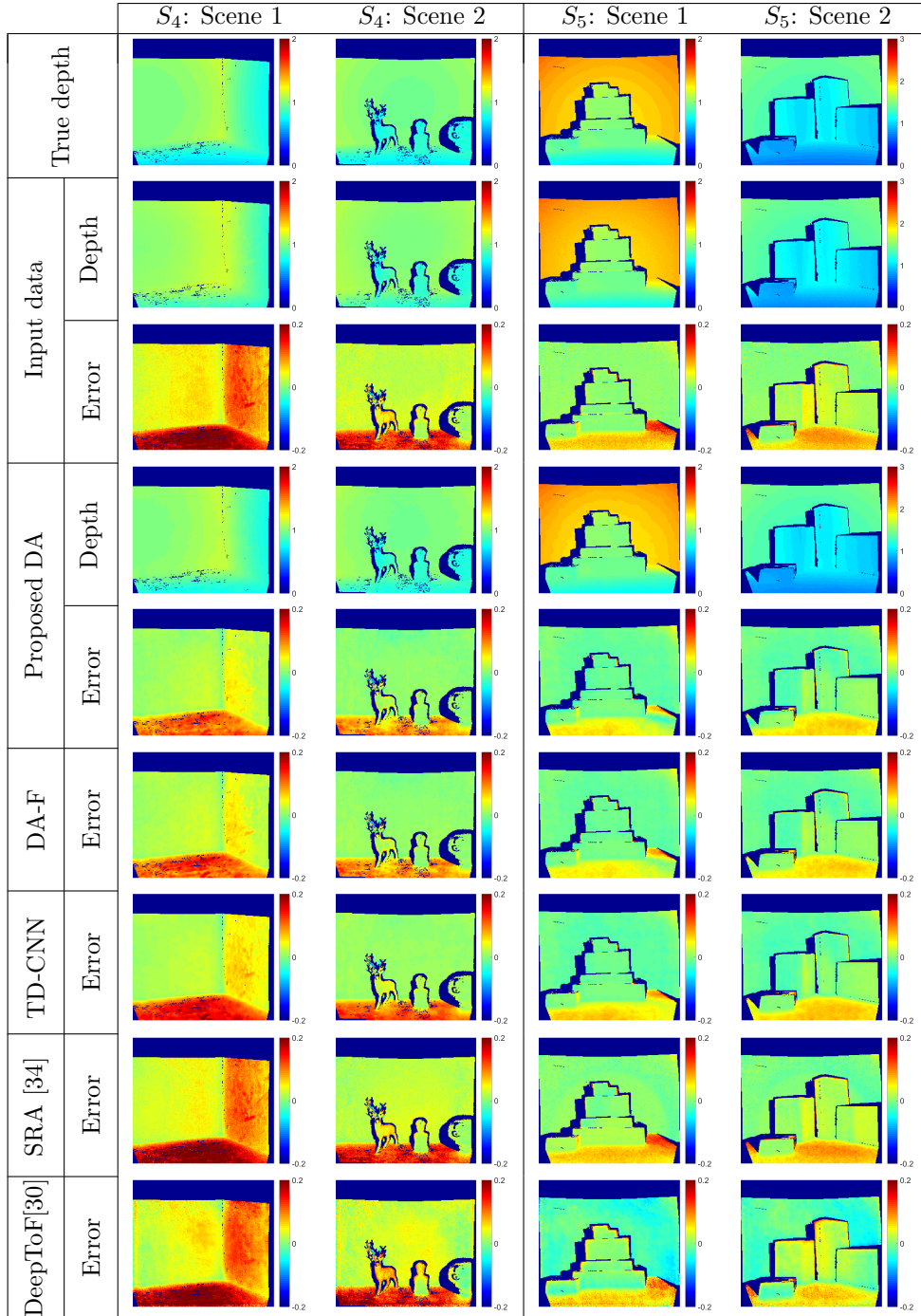


Fig. 5.7: Comparison between the input depth at 60 MHz, the proposed method and the denoised depth maps obtained with some state-of-the-art methods. The figure shows the computed depths with the corresponding error maps, in meters, for some sample scenes from S_4 (first and second columns) and S_5 (third and fourth columns).

Method	S_4 Dataset		S_5 Dataset (<i>box</i>)	
	MAE (cm)	Relative error	MAE (cm)	Relative error
Input (60 Mhz)	5.43	-	3.62	-
Input (20 Mhz)	7.28	-	5.06	-
SRA [34]	5.11	94.1%	3.37	93.1%
DeepToF [30]	5.13	70.5%*	6.68	132%*
[30]+calibration	5.46	75%*	3.36	66.4%*
TD-CNN	3.13	57.6%	1.98	54.7%
Proposed DA on TD-CNN	2.36	43.5%	1.66	46.1%
DA-F (TD-CNN+ [79,101])	2.6	47.9%	1.71	47.2%

Table 5.2: 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 [30] (*) that is compared with 20 MHz since it uses only this frequency.

DeepToF [30] 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 case. TD-CNN is better performing and removes about 45% 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 TD-CNN and it gets close to our approach, removing about 53% of the error.

Some visual results are shown in Fig. 5.7 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 with feature-based domain adaptation DA-F (TD-CNN+ [79,101]) looks visually good. The compared methods are able to remove a smaller amount of MPI. The best of the compared ones is the synthetic trained TD-CNN (Section 4.3.3), [34] have limited performance and [30] 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.

Fig. 5.8 shows the correction obtained with the different methods on a cross-

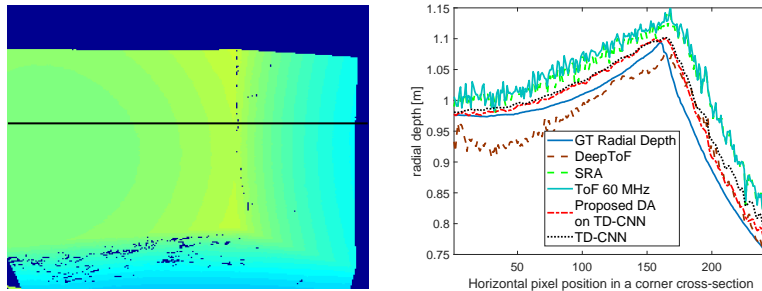


Fig. 5.8: Comparison of different MPI correction methods on a cross-section of a corner scene.

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.

5.4 Domain Adaptation on Semantic Segmentation

A different field in which unsupervised domain adaptation is worth to be investigated is semantic segmentation. Many different approaches for semantic segmentation of images have been proposed [85]. There are many different strategies for this task, but most current state-of-the-art approaches are based on encoder-decoder schemes and on approaches based on variations of the Fully Convolutional Network (FCN) model [107]. Some recent well-known and highly performing methods are DilatedNet [108], PSPNet [109] and DeepLab v2 [110]. In particular, the latter is the model employed for the generator network in this work. All these networks are very efficient in their task, but in order to be trained they require a huge number of couples of color images and related segmented map. For this reason, usually synthetic datasets as GTA5 [89] and SYNTIA [90] are exploited during the training. However, when the synthetic trained network are tested on real data, a degradation in performance can be observed.

The method presented in the next sections aims at applying unsupervised domain adaptation from synthetic data to real data. We focused on semantic segmentation of road scenes, since this is one of the main components of autonomous driving systems, that is one of the most growing area in computer vision. For this

reason, we focused on adapting a segmentation network trained on the synthetic datasets GTA and SYNTHIA to work on the Cityscapes real dataset [86]. All of these three datasets contain images acquired in road environments.

The proposed method exploits an adversarial learning framework, where a segmentation network based on the DeepLab v2 framework is trained using both labeled and unlabeled data thanks to the combination of three different losses. The first is a standard supervised cross-entropy loss exploiting ground truth annotations allowing to perform an initial supervised training phase on synthetic data. The second is an adversarial loss derived from previous methods [94, 111] developed in the context of semi-supervised semantic segmentation (i.e., for dealing with datasets only partially annotated). Finally, the third term is based on a self-teaching framework inspired from [111], where the predicted segmentation is passed through the discriminator to obtain a confidence map and then high confidence regions are considered reliable and used as ground truth for self-teaching the network over the unlabeled real data. We trained the network on both synthetic labeled data (using the first and second component of the loss) and on unlabeled real world data (using the second and third component) thus being able to obtain accurate results on real world datasets even without using labeled real world data.

5.4.1 Architecture of the Proposed Approach

The proposed approach is based on two main modules, i.e., two different Convolutional Neural Networks (CNNs). The first network is the generator (G) of the adversarial learning framework. It performs the semantic segmentation of the given color image. For this module, we exploited the Deeplab v2 network [110] based on the ResNet-101 model whose weights were pre-trained¹ on the MSCOCO dataset [112]. Although we considered the Deeplab v2, notice that our approach does not rely on specific properties of this network and any network for semantic segmentation can be fit inside the proposed learning framework. Fig.5.9 shows a general overview of the procedure used to train G exploiting three different losses.

Starting from the first, the network produces a class probability map representing for each pixel the probability that it belongs to each class c inside the set of

¹We used the weights computed by V. Nekrasov available at: <https://github.com/DrSleep/tensorflow-deeplab-resnet>

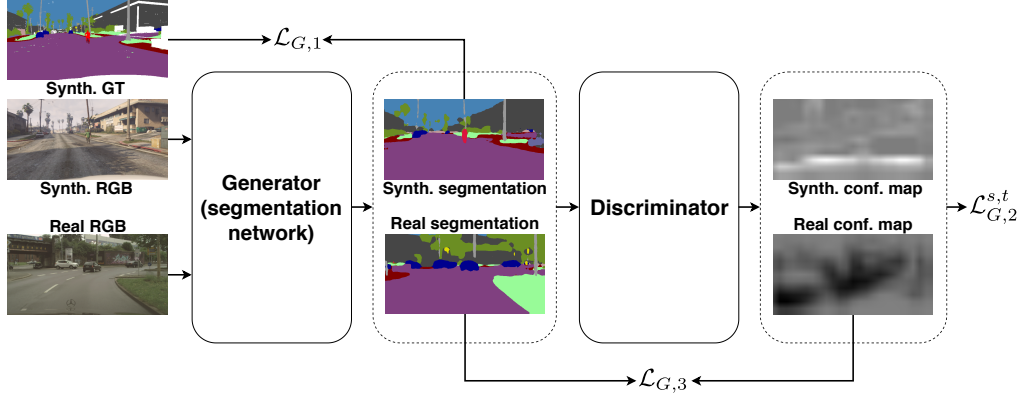


Fig. 5.9: Architecture of the proposed framework for the training of the generator network. A first stage of supervised learning with synthetic data is followed by a second stage using also unlabeled real data to boost the performance of the segmentation network (i.e., the generator) through the combination of 3 losses. $\mathcal{L}_{G,1}$ is a standard cross-entropy loss computed on synthetic data, $\mathcal{L}_{G,2}^{s,t}$ is an adversarial loss referring to a fully-convolutional discriminator network, and $\mathcal{L}_{G,3}$ is a self-teaching loss for unlabeled real data.

possible classes \mathcal{C} . This map can be directly used to train the network in a supervised way exploiting the semantic ground truth data: we used a standard cross-entropy loss ($\mathcal{L}_{G,1}$) for this task. More in detail, given the n -th input image \mathbf{X}_n^s from the source (synthetic) domain, its one-hot encoded ground truth segmentation \mathbf{Y}_n^s and the output of the segmentation network $G(\mathbf{X}_n^s)$, the loss $\mathcal{L}_{G,1}$ on the image \mathbf{X}_n^s can be computed as:

$$\mathcal{L}_{G,1} = - \sum_{p \in \mathbf{X}_n^s} \sum_{c \in \mathcal{C}} \mathbf{Y}_n^{s(p)}[c] \cdot \log (G(\mathbf{X}_n^s)^{(p)}[c]) \quad (5.9)$$

where p is a generic pixel in the considered image \mathbf{X}_n^s , c is a particular class contained in the set \mathcal{C} of possible classes and $\mathbf{Y}_n^{s(p)}[c]$ and $G(\mathbf{X}_n^s)^{(p)}[c]$ are respectively the value in the one-hot encoded ground truth and in the generator network estimate related to the pixel p and the class c .

Notice that this loss can be computed only on the source domain (i.e., on synthetic data) where the pixel-level semantic ground truth is available. However, the target is to adapt the supervised synthetic training to the real world target domain in an unsupervised way. We exploited an adversarial learning framework: a second

CNN is introduced, i.e., a discriminator network (D) that aims at distinguishing segmentation maps produced by the generator from the ground truth ones. Differently from other adversarial learning models, this network produces a per-pixel prediction instead of a single binary value for the whole input image. The discriminator D is made of a stack of 5 convolutional layers each with 4×4 kernels with a stride of 2 and Leaky ReLU activation function. The number of filters (from the first layer to the last one) is 64, 64, 128, 128, 1 and the cascade is followed by a bilinear upsampling to match the original input image resolution. The loss of the discriminator \mathcal{L}_D is a standard cross-entropy loss between the produced map and the one-hot encoding related to the *fake* domain (class 0) or ground truth domain (class 1) depending on the fact that the input has been respectively drawn from the generator or from ground truth data. Mathematically, \mathcal{L}_D is defined as:

$$\mathcal{L}_D = - \sum_{p \in \mathbf{X}_n^{s,t}} \log(1 - D(G(\mathbf{X}_n^{s,t}))^{(p)}) + \log(D(\mathbf{Y}_n^s)^{(p)}) \quad (5.10)$$

Notice that the discriminator has to label with 0 the segmentation maps produced by the generator using both synthetic data from the source domain s (denoted with \mathbf{X}_n^s) or real world data from the target domain t (i.e., \mathbf{X}_n^t). Thus, it allows to exploit also the real world data without ground truth in an unsupervised way, and it tries to distinguish the segmentations produced by the generator G from ground truth segmentation data (that can be only synthetic in our framework). The usage of both types of data is made possible by the similar classes' statistics of source and target datasets. Notice also that, in principle, the task of the discriminator appears to be trivially solvable by distinguishing a Dirac distributed input (i.e., the one-hot encoded annotations) from other prediction distributions. However, we have empirically observed that the generator network produces (and is forced to produce even more by the adversarial training process) segmentation maps which are very close to a Dirac distribution. The second loss term for the training of G is $\mathcal{L}_{G,2}^{s,t}$, that is computed on the generic image $\mathbf{X}_n^{s,t}$ from the discriminator output as:

$$\mathcal{L}_{G,2}^{s,t} = - \sum_{p \in \mathbf{X}_n^{s,t}} \log(D(G(\mathbf{X}_n^{s,t}))^{(p)}) \quad (5.11)$$

This term forces the training of the generator network in the direction of fooling the discriminator producing data that resembles the ground truth statistics. Notice that in this computation the image can be taken from both the source or the target dataset (i.e., it can be both a synthetic or a real world image): in the following of this section, we are going to use $\mathcal{L}_{G,2}^s$ to refer to the loss function computed only on data extracted from the source dataset, while $\mathcal{L}_{G,2}^t$ refers to the loss computed on data from the target dataset. In particular, in the second case, $\mathcal{L}_{G,2}^t$ tries to force the generator to adapt to the target domain and to improve the performance by encouraging cleaner segmentations and global consistency with respect to the segment shapes.

Finally, starting from the idea in [111] we exploited the output of the discriminator D as a confidence measure representing the reliability of the estimations performed by G . This allows to perform a sort of self-training following the idea that the predictions of G are more reliable where D marks them as ground truth with an higher accuracy. This is represented by the third loss component of the generator, defined as:

$$\mathcal{L}_{G,3} = - \sum_{p \in \mathbf{X}_n^t} \sum_{c \in \mathcal{C}} I_{T_u}^{(p)} \cdot W_c^t \cdot \hat{\mathbf{Y}}_n^{(p)}[c] \cdot \log(G(\mathbf{X}_n^t)^{(p)}[c]) \quad (5.12)$$

where $\hat{\mathbf{Y}}_n$ is the estimated one-hot encoded ground truth computed by taking the per-class argmax of the generated probability map $G(\mathbf{X}_n)$. W_c^t , instead, is the weighting function on the source domain defined as:

$$W_c^t = 1 - \frac{\sum_n |p \in \mathbf{X}_n^s \wedge p \in c|}{\sum_n |p \in \mathbf{X}_n^s|}, \quad (5.13)$$

where $|\cdot|$ represents the cardinality of the considered set.

This set of weights serves as a balancing factor when unlabeled data of the target set are used. Without this weighting factor, unlabeled data would lead the model to mislead rare and tiny objects (such as *traffic lights* or *pole*) as frequent and large ones (such as *road*, *building*). Notice that the term comes into play when using unlabeled data of the target domain but the class frequencies have to be computed on the labeled data of the source domain since we need the ground truth labels to evaluate it. This calculation has only to be performed *a priori* and it is not changed

as the learning progresses.

Finally, $I_{T_u}^{(p)}$ is an indicator function defined as:

$$I_{T_u}^{(p)} = \begin{cases} 1, & \text{if } D(G(\mathbf{X}_n^t))^{(p)} > T_u \\ 0, & \text{otherwise} \end{cases} \quad (5.14)$$

with T_u being a threshold for the pixel-wise confidence maps generated by the discriminator in response to the data produced by the generator. We empirically set $T_u = 0.2$ being a reasonable value. This term is intended to enhance the learning process in a self-taught manner using unlabeled data of the target domain.

To conclude, a weighted average of the three losses is used to train the generator exploiting the proposed adversarial learning framework, i.e.:

$$\mathcal{L}_{full} = \mathcal{L}_{G,1} + w^{s,t} \mathcal{L}_{G,2}^{s,t} + w' \mathcal{L}_{G,3} \quad (5.15)$$

We set the weighting parameters empirically to balance between the three components as $w^s = 0.01$, $w^t = 0.001$ to give less weight in case of unlabeled data and $w' = 0.1$.

The discriminator is fed both with ground truth labels and with the generator output computed on a mixed batch containing both labeled and unlabeled data and is trained aiming at minimizing \mathcal{L}_D . Concerning the generator, instead, during the first 5000 steps $\mathcal{L}_{G,3}$ is disabled (i.e., w' is set to 0) thus allowing the discriminator to enhance its capabilities to produce higher quality confidence maps before using them. After this, the training process continues up to 20000 steps with all the three components of the loss enabled.

5.4.2 Datasets

In this section, we introduce the datasets used to evaluate the performance of the proposed unsupervised domain adaptation framework. The target is to show how it is possible to train a semantic segmentation network in a supervised way on synthetic datasets and then apply unsupervised domain adaptation to real data in autonomous driving scenarios. Thus, we used two publicly available synthetic datasets, namely GTA5 [89] and SYNTHIA [90] for the supervised part of the

training, while the unsupervised adaptation and the result evaluation have been performed on the real world Cityscapes [86] dataset. In general we followed the same evaluation scenarios of the competing approaches for fair comparison [91–93].

GTA5 [89] is a huge dataset composed by 24966 photo-realistic synthetic images with pixel level semantic annotation. The images have been recorded from the prospective of a car in the streets of virtual cities (resembling the ones in California) in the open-world video game *Grand Theft Auto 5*. Being taken from a high budget commercial production they have an impressive visual quality and are very realistic. In our experiments, we used 23966 images for the supervised training and 1000 images for validation purposes. There are 19 semantic classes which are compatible with the ones of the Cityscapes dataset. The original resolution of the images is 1914×1052 pxl but we rescaled and cropped them to the size of 375×750 pxl for memory constraints before being fed to the architecture.

SYNTHIA [90] is a very large dataset of photo-realistic images. It has been produced with an ad-hoc rendering engine, allowing to obtain a large variability of the images. On the other hand, the visual quality is not the same of the commercial video game GTA5. We used the *SYNTHIA-RAND-CITYSCAPES* version of the dataset, which contains 9400 images with annotations compatible with 16 of the 19 classes of Cityscapes. These images have been captured on the streets of a virtual European-style town in different environments under various light and weather conditions. As done in previous approaches, we randomly extracted 100 images for validation purposes from the original training set, while the remaining part, composed by 9300 images, is used for the supervised training of our networks. Again, the images have been rescaled and cropped from the original size of 760×1280 pxl to 375×750 pxl. For the evaluation of the proposed unsupervised domain adaptation on the Cityscapes dataset, only the 16 classes contained in both datasets are taken into consideration.

Cityscapes [86] is the target dataset for our domain adaptation framework. It is composed by 2975 high resolution color images captured on the streets of 50 different European cities. They have pixel level semantic annotation with 34 classes overall. Since the labels of the original test set are not available, we exploited the original training set (without the labels) for unsupervised training and used the 500 images in the original validation set as a test set, as done also by other recent

approaches.

More in detail, the semantic labels have been used just for testing purposes, while the labels of training data have not been used since we aim at proposing a fully unsupervised adaptation strategy. As for the other datasets, the original high resolution images have been resized to 375×750 pxl for memory constraints. The testing was instead carried out on the original resolution of 2048×1024 pxl.

	road	sidewalk	building	wall	fence	pole	t light	t sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mean
Ours ($\mathcal{L}_{G,1}$)	45.3	20.6	50.1	9.3	12.7	19.5	4.3	0.7	81.9	21.1	63.3	52.0	1.7	77.9	26.0	39.8	0.1	4.7	0.0	27.9
Ours ($\mathcal{L}_{G,1}, \mathcal{L}_{G,2}^s$)	61.0	18.5	51.6	15.4	12.3	20.5	1.4	0.0	82.6	24.7	61.0	52.1	2.2	78.5	25.9	41.5	0.4	8.0	0.1	29.3
Ours (\mathcal{L}_{full})	54.9	23.8	50.9	16.2	11.2	20.0	3.2	0.0	79.7	31.6	64.9	52.5	7.9	79.5	27.2	41.8	0.5	10.7	1.3	30.4
Hoffman [91]	70.4	32.4	62.1	14.9	5.4	10.9	14.2	2.7	79.2	21.3	64.6	44.1	4.2	70.4	8.0	7.3	0.0	3.5	0.0	27.1
Hung [111]	81.7	0.3	68.4	4.5	2.7	8.5	0.6	0.0	82.7	21.5	67.9	40.0	3.3	80.7	34.2	45.9	0.2	8.7	0.0	29.0

Table 5.3: Mean intersection over union (mIoU) on the different classes of the original Cityscapes validation set. The approaches have been trained in a supervised way on the GTA5 dataset and then the unsupervised domain adaptation has been performed using the Cityscapes training set.

	road	sidewalk	building	wall	fence	pole	t light	t sign	veg	sky	person	rider	car	bus	mbike	bike	mean
Ours ($\mathcal{L}_{G,1}$)	10.3	20.5	35.5	1.5	0.0	28.9	0.0	1.2	83.1	74.8	53.5	7.5	65.8	18.1	4.7	1.0	25.4
Ours ($\mathcal{L}_{G,1}, \mathcal{L}_{G,2}^s$)	9.3	19.3	33.5	0.9	0.0	32.5	0.0	0.5	82.3	76.9	54.7	5.5	64.9	17.0	5.7	3.9	25.4
Ours (\mathcal{L}_{full})	78.4	0.1	73.2	0.0	0.0	16.9	0.0	0.2	84.3	78.8	46.0	0.3	74.9	30.8	0.0	0.1	30.2
Hoffman [91]	11.5	19.6	30.8	4.4	0.0	20.3	0.1	11.7	42.3	68.7	51.2	3.8	54.0	3.2	0.2	0.6	20.1
Hung [111]	72.5	0.0	63.8	0.0	0.0	16.3	0.0	0.5	84.7	76.9	45.3	1.5	77.6	31.3	0.0	0.1	29.4

Table 5.4: Mean intersection over union (mIoU) on the different classes of the original Cityscapes validation set. The approaches have been trained in a supervised way on the SYNTHIA dataset and then the unsupervised domain adaptation has been performed using the Cityscapes training set.

5.4.3 Experimental Results

The target of the proposed approach is to adapt a deep network trained on synthetic data to real world scenes. To evaluate the performance on this task we performed two different sets of experiments. In the first experiment we trained the network using the scenes from the GTA5 dataset to compute the supervised loss $\mathcal{L}_{G,1}$ and the adversarial loss $\mathcal{L}_{G,2}^s$. Then we used the training scenes of the Cityscapes dataset

for the unsupervised domain adaptation: no labels from Cityscapes have been used and when dealing with this dataset we only computed the losses $\mathcal{L}_{G,2}^t$ and $\mathcal{L}_{G,3}$. Finally we evaluated the performance on the validation set of Cityscapes. In the second experiment we performed the same procedure but we replaced the GTA5 dataset with the SYNTHIA one.

The generator network G (that is a Deeplab v2 network) has been trained as proposed in [110] using the Stochastic Gradient Descent (SGD) optimizer with momentum set to 0.9 and weight decay to 10^{-4} . The discriminator D has been trained using the Adam optimizer. The learning rate employed for both G and D started from 10^{-4} and was decreased up to 10^{-6} by means of a polynomial decay with power 0.9. We trained the two networks for 20000 iterations on a NVIDIA Titan X GPU. The longest training inside this work, i.e., the one with all the losses enabled, took about 10 hours to complete.

To measure the performance, we compared the predictions on the Cityscapes validation set with the ground truth labels and computed the mean Intersection over Union (mIoU) as done by most competing approaches [91, 96, 98].

Table 5.3 refers to the first experiment (i.e., using GTA5 for the supervised training). It shows the accuracy of the proposed approach when exploiting different domain adaptation strategies and compares it with some state-of-the-art approaches. By simply training the network in a supervised way on the GTA5 dataset and then performing inference on real world data from the Cityscapes dataset we obtained a mIoU of 27.9%. The adversarial learning framework on synthetic data (i.e., the contribution of $\mathcal{L}_{G,2}^s$) allows to improve the mIoU to 29.3%. By looking more in detail to the various class accuracies it is possible to see that the accuracy has increased on some of the most common classes corresponding to large structures, while the behaviour on low frequency classes corresponding to small objects is more unstable (some improve but others have a lower accuracy). For this reason in the third loss component related to the self-teaching, the class weights have been taken into account. Thanks to this when using the full framework with all the losses the mIoU increases to 30.4% and in particular it is possible to appreciate a large performance boost on many uncommon classes corresponding to small objects and structures.

By comparing with state-of-the-art approaches, it is possible to see how the

method of Hung et al. [111], based on a similar framework, achieves an accuracy of 29%, lower than our approach mostly because it struggles with small structures and uncommon classes. The method of [91] has even lower performance, however it is also based on a different generator network with lower accuracy (i.e, the method of [108]).

Fig. 5.10 shows the output of the different versions of our approach and of the method of [111] on some sample scenes. The supervised training leads to reasonable results but some small objects get lost or have a wrong shape (e.g., the riders in row 1). Furthermore, some regions of the street and of structures like the walls are corrupted by noise (see the street in the last two rows or the fence on the right in row 3). The adversarial loss $\mathcal{L}_{G,2}^s$ reduces these artifacts but there are still issues on the small objects (e.g., the rider in the fifth row) and the boundaries are not always very accurate (see the fence in the third row). The complete model leads to better performance, for example in the images of Fig. 5.10 the people are better preserved and the structures have better defined edges. Finally the approach of [111] seems to lose some structures (e.g., the fence in the third row) and has issues with the small objects (the riders in row 5 get completely lost) as pointed out before.

By using the SYNTHIA dataset as source dataset, the domain adaptation task is even more challenging if compared with the GTA5 case since the computer generated graphics are less realistic. Table 5.4 shows that by training the network G in a supervised way on the SYNTHIA dataset and then performing inference on the real world Cityscapes dataset, a mIoU of 25.4% can be obtained. This value is smaller than the mIoU of 27.9% obtained by training G on the GTA5 dataset. This result confirms that the GTA5 dataset has a smaller domain shift with respect to real world data, when compared with the SYNTHIA dataset (GTA5 data, indeed, have been produced by a more advanced rendering engine with more realistic graphics). Under this training scenario, the proposed adversarial loss $\mathcal{L}_{G,2}^s$ does not bring to noteworthy improvements indeed the mIoU is equal to the *baseline*. On the other hand, by adding the self-taught loss $\mathcal{L}_{G,3}$, a noticeable improvement to a mIoU of 30.2% can be obtained.

Our domain adaptation framework is able to outperform the compared state-of-the-art approaches. The method of Hung et al. [111], that exploits the same generator architecture of our approach, obtains a mIoU equal to 29.4%, lower than

5.4. DOMAIN ADAPTATION ON SEMANTIC SEGMENTATION

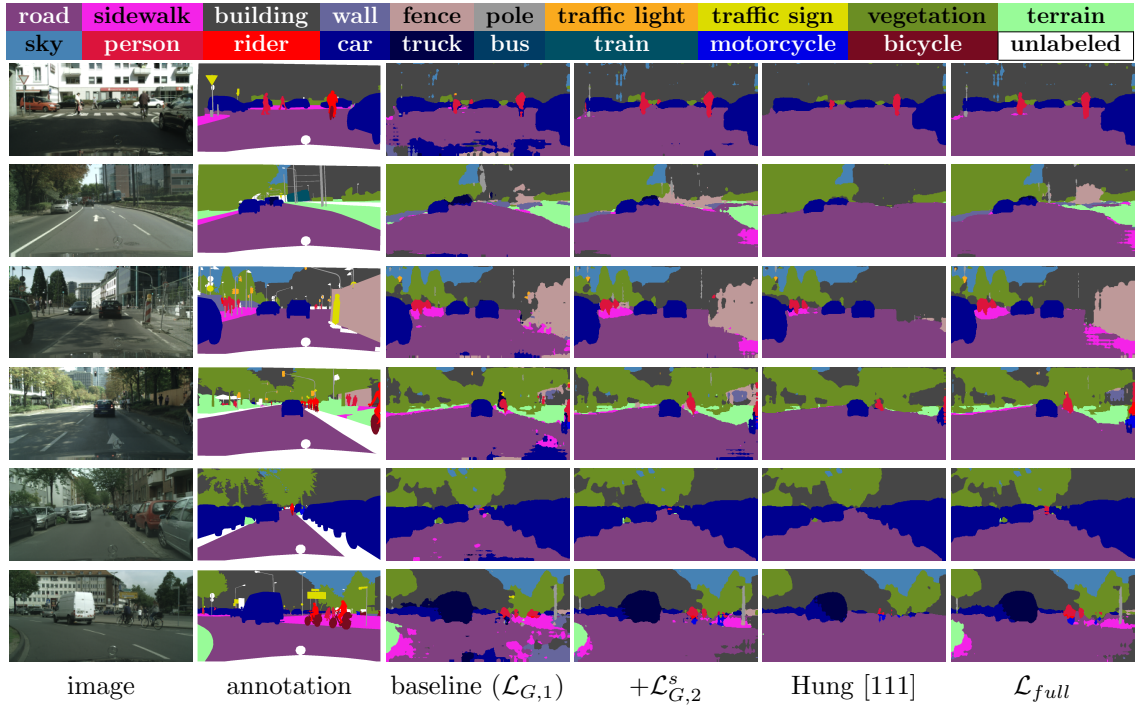


Fig. 5.10: Semantic segmentation of some sample scenes extracted from the Cityscapes validation dataset. The network has been trained using GTA5 with annotations and Cityscapes for the unsupervised part (*best viewed in colors*).

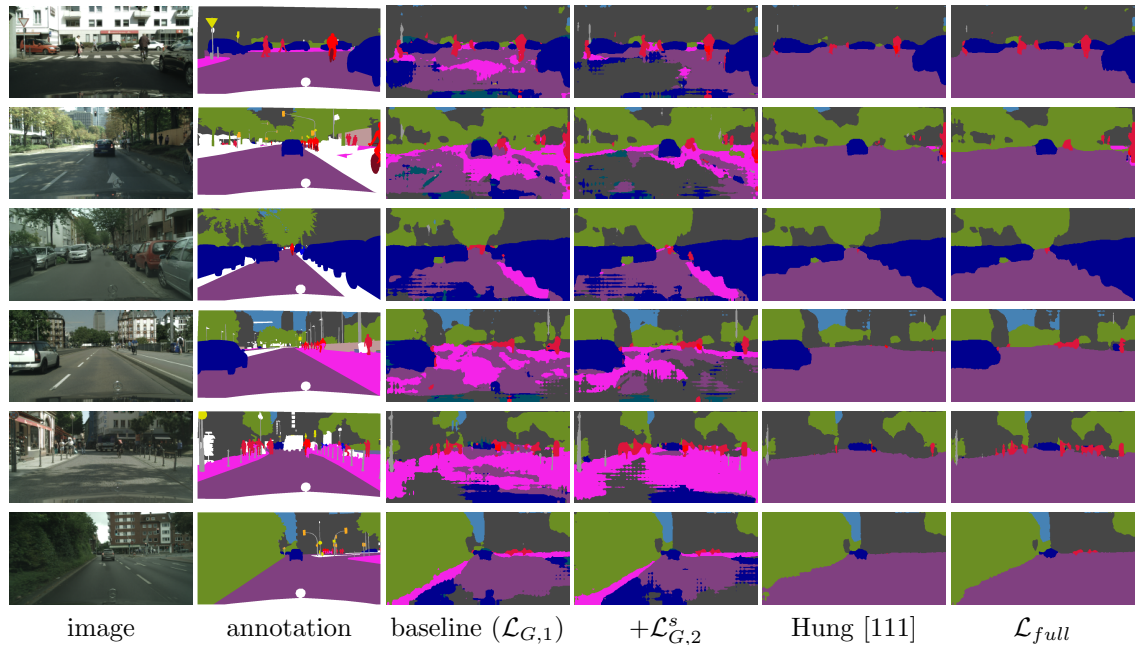


Fig. 5.11: Semantic segmentation of some sample scenes extracted from the Cityscapes validation dataset. The network has been trained using SYNTHIA with annotations and Cityscapes for the unsupervised part (*best viewed in colors*).

our method. The method of [91] appears to be again the less performing approach. In this comparison, it is even less accurate than our *baseline*, but it employs a different segmentation network.

Fig. 5.11 shows the output of the different versions of our approach and of the method of [111] on some sample scenes. The first thing that can be noticed by looking at the qualitative results of the *baseline* supervised version is that by training on the SYNTHIA dataset some classes as *sidewalk* and *road* are highly corrupted. It is evident that a simple synthetic supervised training starting from this dataset would bring to a network which can not be used in an autonomous vehicle scenario. This is probably caused by the not completely realistic representation of streets and sidewalks in the SYNTHIA dataset, where their textures are often very unrealistic. Additionally, while the positioning of the camera in the Cityscapes dataset is always fixed and mounted on-board inside the car, in SYNTHIA the camera is placed in different positions. For example, the pictures can be captured from inside the car, from cameras looking from the top or from the side of the road.

Similarly to the *baseline* approach, the adversarial loss $\mathcal{L}_{G,2}^s$ is unable to adapt the network to the real domain, indeed the class *road* remains very badly detected also after its usage. Differently, Fig. 5.11 shows how unlabelled data and the self-teaching component of the third loss allows to avoid all the artifacts on the *road* surface by reinforcing the segmentation network to capture the real nature of this class in the Cityscapes dataset. Also Hung’s method [111] is able to correctly reconstruct the class *road*, avoiding the noise present in the *baseline*, but it suffers on small classes where it is outperformed by the proposed method. This is clearly visible on rows 4 and 5 of Fig. 5.11, where our method is able to locate more precisely small classes as *person*.

5.4.4 Ablation Study

In this section, we are going to analyze the contributions of the various terms controlling the optimization in the proposed framework. Table 5.5 collects the results of this analysis on the Cityscapes validation split when using GTA5 as source dataset for the supervised part.

As it is possible to notice from Table 5.5, the generator network trained in a

$\mathcal{L}_{G,1}$	$\mathcal{L}_{G,2}^s$	$\mathcal{L}_{G,2}^t$	$\mathcal{L}_{G,3}$	mIoU
✓				27.9
✓	✓			29.3
✓		✓		27.9
✓	✓	✓		29.4
✓			✓	28.7
✓	✓	✓	✓	30.4

Table 5.5: Mean intersection over union (mIoU) of some configurations of our framework on the Cityscapes validation set using GTA5 as source dataset.

supervised way with the standard cross entropy loss (i.e., using only $\mathcal{L}_{G,1}$) is the less performing strategy achieving a mIoU of 27.9%. Some improvements can be obtained by adding the adversarial term $\mathcal{L}_{G,2}^s$ in the loss function, that is by exploiting also adversarial learning on the source dataset. In this case, the segmentation network is more accurate achieving a mIoU of 29.3%. The domain adaptation using adversarial learning on the target dataset only, i.e., $\mathcal{L}_{G,2}^t$ in combination with $\mathcal{L}_{G,1}$ obtains results very similar to the *baseline* approach. However, when $\mathcal{L}_{G,2}^t$ is used in combination with $\mathcal{L}_{G,2}^s$ a slight improvement is noticeable. The exploitation of the self-teaching module $\mathcal{L}_{G,3}$, without adversarial learning, allows to perform some adaptation to the segmentation network obtaining a mIoU of 28.7% (the main issue is the low performance on the road class since it is not able to remove the noise of the baseline method on it). The last row contains the results of the complete version of our approach, where all the aforementioned components are taken in consideration. We can appreciate that the full combination is able to outperform the exploitation of each of the single components and achieves a mIoU of 30.4%.

Chapter 6

Fusion of ToF and Stereo Data

Previous chapters considered the case in which the depth data acquired by a ToF camera are refined without using any form of side information from other sensors. Differently, this chapter considers the joint exploitation of a ToF sensor and a passive stereo vision system. The data coming from these two data sources have complementary strengths and flaws as discussed in Chapter 2. Indeed, the first is able to robustly estimate the 3D geometry independently of the scene content but they have a limited spatial resolution, a high level of noise and a reduced accuracy on low reflective surfaces. The second can acquire a high resolution scene depth but their accuracy strongly depends on the scene content and the acquisition is not very reliable on uniform or repetitive regions. For this reason, it is worth investigating fusion techniques to estimate a more reliable scene depth map exploiting the two depth data sources.

In the next of this chapter, first a review of the literature about stereo-ToF fusion will be reported. Then a novel method, designed during my Ph.D. with other members of the LTTM laboratory at the University of Padova, will be introduced and its performance will be evaluated.

6.1 Literature about Stereo-ToF Fusion

Depth estimation using stereo vision cameras is a long term research field and a large number of different approaches have been proposed and tested on public data

like the Middlebury [113] and KITTI [114] benchmarks. A good review on this topic is [115]. Despite the large amount of research and the continuous improvement of the performance of these methods, the depth estimation accuracy of stereo systems depends on many factors, and in particular on the photometric content of the scene. The estimation is less accurate in regions with fewer details, i.e., when the scene contains a limited amount of texture, or on repetitive patterns. Since the accuracy can vary considerably between different scenes or even different regions of the same scene, it is important to estimate the confidence of the computed data. Until a few years ago, the confidence information for stereo systems used to be computed mostly by analyzing some key properties of the stereo matching cost function. A comprehensive review of this family of approaches is [116]. Recently, machine learning techniques started to be used for this task, first with traditional approaches (e.g., Random Forests), then by using deep learning techniques. A very recent review of machine learning approaches for stereo confidence computation is [117]. An example of approach of this family is [118], that uses a CNN to estimate the confidence information from image patches. A two channel image patch representation is used also by [119], while [120] improves standard confidence metrics by enforcing the local consistency of the confidence maps with a deep network.

On the other hand, ToF cameras represent a quite robust solution for depth acquisition [10, 121–123]. The various low cost depth cameras available on the market can acquire depth information in real-time and are more robust to the scene content with respect to stereo systems, in particular they can estimate the depth also in regions without texture or with repetitive patterns. Nevertheless, ToF cameras have their own limitations, e.g., the resolution is typically lower than standard cameras and they are noisy. These cameras are also affected by other sources of errors like the multi-path interference and the mixed pixel effect. As discussed in the previous chapters, only small real ToF depth datasets with related ground truth exist. For this reason, the confidence of ToF data is typically computed with analytical methods.

ToF cameras and stereo vision systems rely on completely different depth estimation principles. For this reason, they have complementary characteristics and the fusion of the data acquired from the two sources should produce more accurate measures. Several different approaches for the combination of stereo and ToF

data have been proposed. Comprehensive reviews of the topic can be found in [10] and [124].

A possible approach is to model the problem with a MAP-MRF Bayesian formulation and optimize a global energy function with belief propagation. This technique has been used by various works of Zhu et al. [125–127]. A probabilistic formulation has been used in [59] that computes the depth map with a ML local optimization. The approach has been extended in [61] that adds a global MAP-MRF optimization scheme. A second possibility is to use a variational fusion framework. Examples of this family are the methods of [128], that also uses confidence measures for the ToF and stereo vision systems to drive the process, and the works of Chen et al. [129, 130], that combines the variational approach with edge-preserving filtering.

A different approach is proposed in [131], that computes the depth data by solving a set of local energy minimization problems. Another solution is to use a locally consistent framework [132] to fuse the two data sources. The idea has been firstly introduced in [133], then improved in [67] by adding the confidence information for the two data sources.

6.2 Stereo-ToF Fusion Guided by Learned Confidences

In the next of this chapter, a stereo-ToF depth fusion method is presented. It starts from [67] exploiting the locally consistent framework but improves the sensors confidence estimation using a deep learning approach trained on the synthetic dataset SYNHT3 introduced in Section 3.3.2.

The target of the proposed work is to combine the data from a ToF camera with a stereo vision system in order to extract an accurate depth representation. Both devices are able to produce an estimation of depth data from the corresponding viewpoint and the proposed method combines these two representations to provide a dense and more accurate depth map from the point of view of one of the color cameras of the stereo setup.

In this work the experimental evaluation is performed on the synthetic dataset SYNTH3 and the real datasets REAL3 and LTTM5. In these datasets, the stereo

and ToF acquisitions are recorded using geometrically calibrated cameras. By using the calibration data it is possible to project the ToF data on the pixel grid of the reference camera of the stereo vision system in order to use them in the fusion process that assumes that all the data are aligned.

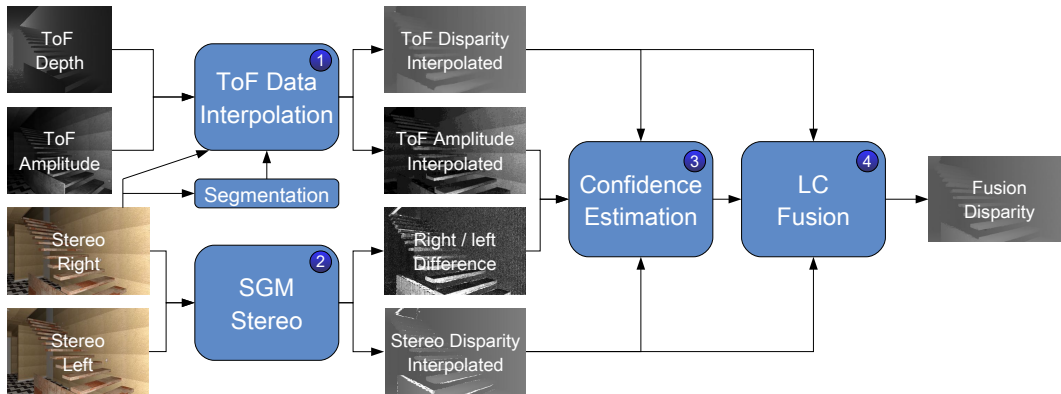


Fig. 6.1: Flowchart of the proposed approach.

The proposed algorithm is divided into four main steps (see Fig. 6.1):

1. The depth information acquired from the ToF sensor is reprojected to the reference color camera viewpoint and interpolated to the same resolution of the color cameras. The interpolation is necessary since ToF sensors have typically a low resolution, specially if compared with modern color cameras. The approach used for this task has been derived from [133]: we used an extended version of the cross bilateral filter where the filter is driven by three terms, the standard spatial Gaussian weighting, the range term computed on the color image and an additional segmentation-based term that depends on a segmented version of the color image computed with Mean-Shift clustering [134]. This procedure allows us to produce a high resolution depth map aligned with the color camera lattice that will be used by the fusion algorithm. Finally, the depth map is converted to a disparity map, since the fusion algorithm works in disparity space. More details on this step can be found in [67, 133].
2. In parallel, the Semi-Global Matching (SGM) stereo vision algorithm [135] is used to compute a high resolution disparity map from the stereo pair. We

selected this algorithm since it provides a good compromise between computation time and performance, however the proposed approach is independent of the selected stereo vision algorithm.

3. After obtaining the two disparity fields, confidence information is jointly estimated for the stereo and ToF disparity maps using the CNN architecture presented in Section 6.2.1.
4. Finally the reprojected and interpolated ToF disparity and the stereo disparity are fused using an extended version of the Locally Consistent (LC) algorithm [67,132]. This step is described in Section 6.2.3.

6.2.1 Confidence Estimation with Deep Learning

A fundamental step in order to reliably fuse the two disparity maps is their per-pixel confidence estimation. To this purpose, we designed and trained a 6-layer CNN that takes in input different clues from ToF and stereo data and jointly uses the information from both devices to infer the two confidence maps. In particular, the proposed CNN takes in input four channels associated to the following clues:

- A difference map Δ encoding for mismatches between corresponding visual cues in the stereo image pair. This is computed by warping on the reference camera the other color image, using the stereo disparity, and subtracting it from the reference image.
- The stereo disparity map D_S .
- The ToF disparity map D_T obtained from the ToF depth map after reprojected on the reference camera and conversion to the disparity space.
- The ToF amplitude image reprojected on the reference camera of the stereo vision system A_T .

Since raw input data correspond to different sources of information coming from heterogeneous sensors, a lightweight pre-processing stage is needed to convert such data into the desired form.

The first clue Δ aims providing a rough measure of the accuracy of the disparities computed by the stereo algorithm. The idea is that accurate disparity estimates are likely to result in pairs of corresponding pixels with similar values in the reference and target stereo images respectively. On the contrary, corresponding pixels computed using inaccurate disparities are likely to hold different values since they correspond to different parts of the scene. In order to compute Δ , both the reference and target stereo images are first converted to grayscale images giving I_R and I_T respectively. The target grayscale image I_T is then reprojected on the reference camera using the stereo disparity, thus obtaining the image I'_T . Finally, the absolute difference between I_R and I'_T is taken, leading to

$$\Delta = \left| I_R - I'_T \right| \quad (6.1)$$

The stereo disparity clue D_S is directly obtained from the stereo disparity map while the ToF disparity clue D_T is derived from the ToF depth map first by reprojecting it on the reference viewpoint, then it is interpolated with a bilateral filter approach and finally it is converted to the disparity space. Similarly, the last clue A is derived by reprojecting the ToF amplitude image onto the reference frame. Finally, the four clues Δ , D_T , D_S and A_T are packed together in a four-channel input tensor where each channel is independently normalized to the unit interval by applying an appropriate scaling factor. Such tensor can be fed to the CNN to produce as output two confidence maps P_T and P_S for the ToF and stereo disparity respectively.

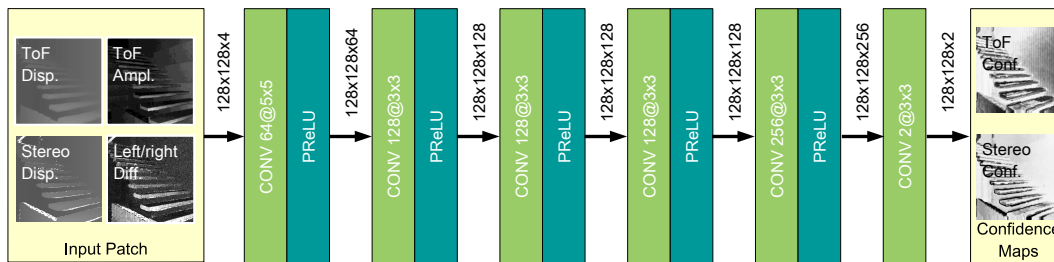


Fig. 6.2: Architecture of the proposed deep learning framework. A 4-channel training patch of size 128×128 [pxl] is fed to a CNN with 6 convolutional layers: the figure shows the number of filters, their spatial kernel sizes and the size of the outputs for each layer.

The proposed CNN architecture is shown in Fig. 6.2. The network is made of a stack of six convolutional layers (CONV) each followed by a Parametric Rectified Linear Unit (PReLU) activation layer, except for the last convolutional layer. The PReLU activation function [136] has been chosen over the standard Rectified Linear Unit (ReLU) activation to prevent the dead-neuron effect caused by negative inputs entering the ReLU zero-slope region. In our experiments, we set the slope of the negative part of the PReLU activation function to 0.02.

The first five layers are assigned an increasing number of filters, namely 64, 128, 128, 128 and 256 respectively. Filter kernels in the first convolutional layer have a spatial size of 5×5 , while kernels in all subsequent layers are 3×3 wide. The last convolutional layer has only two filters in order to produce, as output, a 2-channel tensor, the two channels encoding for the estimated ToF and stereo confidence respectively. To produce an output with the same resolution of the input, no pooling layers are used. At the same time, to cope with the size reduction at the boundaries due to the convolution operation, each convolutional layer applies a suitable padding to its input along each spatial dimension, where padded values are set to be equal to the values at the boundary.

6.2.2 Training of the Convolutional Neural Network

The proposed architecture has been trained on the synthetic dataset SYNTH_{train} described in Section 3.3.2. Although this dataset is smaller if compared with other machine learning datasets, it is the largest dataset for ToF and stereo data fusion containing depth ground truth depth information. We decided to train the network on patches randomly selected from the various scenes instead of using whole images in order to increase the number of training examples. In particular, we generated a large set of training examples by randomly extracting 30 patches of size 128×128 [pxl] from each of the 40 scenes contained in the training set. Moreover, to increase the robustness and variability of the training data, we also augmented the dataset by applying random rotations of $\pm 5^\circ$ as well as horizontal and vertical flipping. Following the augmentation process, a set of about 6000 patches has been generated starting from the 1200 patches initially extracted from the original dataset, thus forming the actual input data used for the training. The training data has been

further split into a training set and a validation set. Validation data has been used to select the network layout and hyper-parameters. Some ablation studies and results obtained with different network architectures are presented in Section 6.2.4, in general deeper and more complex architectures led to a smaller training error but there is no improvement in the validation error and in the fusion results due to overfitting on the not too large training dataset.

The two target confidence maps needed for training have been derived by taking the negative exponential of the absolute error between the ground truth depth information converted to disparity values D_{GT} and the ToF and stereo disparities D_T and D_S respectively, according to the following formulation:

$$\begin{aligned} P_T^* &= e^{-|D_T - D_{GT}|} \\ P_S^* &= e^{-|D_S - D_{GT}|} \end{aligned} \tag{6.2}$$

The network has been trained to minimize a quadratic loss function computed as the Mean Squared Error (MSE) between the predicted ToF and stereo confidence maps P_T and P_S and their corresponding target confidences from Eq. 6.2

$$Loss = \sum (P_T - P_T^*)^2 + \sum (P_S - P_S^*)^2 \tag{6.3}$$

where the two summations are taken over the spatial dimensions. Using a single network minimizing a loss function that combines both ToF and stereo error provided better results than training two separate networks to infer ToF and stereo confidences separately.

The optimization has been performed with the AdaDelta algorithm [137]. The process has been carried out using a batch size of 32 and an initial learning rate equal to 0.01. In each convolutional layer, the kernel weights have been initialized following the procedure proposed by He et al. in [136], while all bias values have been initially set to zero.

Both the CNN model as well as the whole optimization and evaluation framework have been implemented using the TensorFlow library [138]. The training stage runs for 500 epochs and takes about 8 hours on a desktop PC with an Intel i7-4790 CPU and an NVIDIA Titan X (Pascal) GPU.

6.2.3 Fusion of Stereo and ToF Disparity

The confidence estimated by the deep learning framework of Section 6.2.1 can be used to combine the two depth fields coming from the two sensors. The fusion of the upsampled ToF data with the stereo disparity is performed using an extended version of the Locally Consistent (LC) approach.

This method was firstly introduced in [132] for the refinement of stereo disparity data. It refines the disparity estimation by propagating, within an active support centered on the considered point f , the plausibility $\mathcal{P}_{f,g}(d)$ of the disparity assignment coming from other points g inside the active support. The plausibility of a disparity hypothesis d depends on the color and spatial consistency of the considered pixels:

$$\mathcal{P}_{f,g}(d) = e^{-\frac{\Delta_{f,g}}{\gamma_s}} \cdot e^{-\frac{\Delta_{f,g}^\psi}{\gamma_c}} \cdot e^{-\frac{\Delta_{f',g'}^\psi}{\gamma_c}} \cdot e^{-\frac{\Delta_{g,g'}^\omega}{\gamma_t}} \quad (6.4)$$

where f, g and f', g' refer to the coordinates in the left and right image respectively, Δ accounts for spatial proximity, Δ^ψ and Δ^ω encode color similarity, and the parameters γ_s, γ_c and γ_t control the relative relevance of the various terms (a detailed description can be found in [132]). The overall plausibility $\Omega_f(d)$ of a disparity hypothesis d is computed by aggregating the plausibility for the same disparity value propagated from neighboring points, i.e.:

$$\Omega_f(d) = \sum_{g \in \mathcal{A}} \mathcal{P}_{f,g}(d). \quad (6.5)$$

Finally, a winner-takes-all strategy is used to compute the optimal disparity value.

A first extension of the approach has been presented in [133] to account for multiple disparity hypotheses as in the case of our setup. The approach of [133] allows to obtain quite good results in the fusion of the two disparity fields but has the key limitation that assigns the same weight to the two data sources without accounting for their reliability.

For this reason the method has been further extended in [67] by assigning different weights to the plausibilities according to the estimated confidence value for

each depth acquisition system computed at each pixel location g :

$$\Omega'_f(d) = \sum_{g \in \mathcal{A}} \left(P_T(g) \mathcal{P}_{f,g,T}(d) + P_S(g) \mathcal{P}_{f,g,S}(d) \right) \quad (6.6)$$

where $\Omega'_f(d)$ is the plausibility at point f for disparity hypothesis d , $\mathcal{P}_{f,g,T}(d)$ is the plausibility propagated by neighbouring points g according to ToF data and $\mathcal{P}_{f,g,S}(d)$ is the one according to stereo data. Finally, $P_T(g)$ and $P_S(g)$ are the ToF and stereo confidence values at location g respectively. Another improvement to the LC method introduced in [67] is the depth estimation at subpixel precision that allows to obtain a better accuracy. In [67] the confidence information is computed with a deterministic algorithm based on the noise model for the ToF sensor and on the cost function analysis for the stereo system, while in the proposed approach the confidence is estimated with the deep learning architecture of Section 6.2.1, that ensures that the estimated confidences maps are coherent with respect to each other and to the true error. For the experimental results of this work the parameters have been set to $\gamma_s = 8$, $\gamma_c = 6$ and $\gamma_t = 4$. Finally notice how the proposed framework can easily be extended to setups with more than two input channels in order to perform the fusion of multiple sensors based on different technologies.

6.2.4 Experimental Results

The proposed approach has been evaluated on three different datasets. We started by evaluating its performance on the SYNTH3 synthetic dataset and then moved to the experiments using data collected with real cameras. For the real world experiments we used both the REAL3 dataset and the LTTM5 dataset from [61].

Evaluation on the SYNTH3 Dataset

For this set of experiments, the proposed fusion algorithm has been trained and evaluated on synthetic data from the SYNTH3 dataset described in Section 3.3.2. The SYNTH3 test set, SYNTH3_{test} , contains 15 different scenes with very different properties including different acquisition ranges, textured and un-textured surfaces, complex geometries and strong reflections. The algorithm takes in input the 512×424 [pxl] depth and amplitude maps from the ToF sensor and the two 960×540 [pxl]

color images from the cameras (the color cameras resolution has been halved with respect to the original input data). The output is computed from the point of view of the right camera at the color data resolution of 960×540 [pxl]. For performance evaluation, it has been cropped to consider only on the region that is framed by all the three cameras and compared with ground truth data. Ground truth information has been computed by extracting the depth data from the Blender rendering engine and converting it to the disparity space.

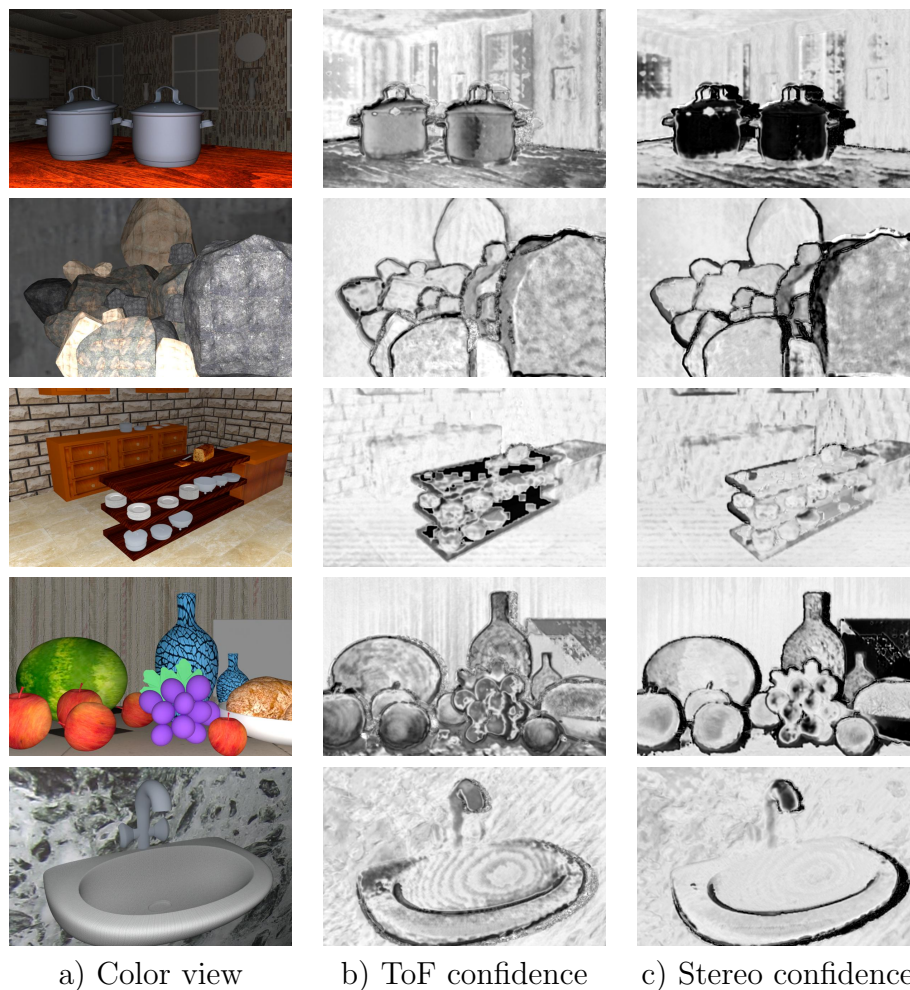


Fig. 6.3: Confidence information estimated by the proposed method for some sample scenes: a) color view; b) estimated ToF confidence; c) estimated stereo confidence. Brighter areas correspond to higher confidence values, while darker pixels to lower ones.

Before evaluating the performance of the fusion scheme we analyze the confidence information computed with the deep learning approach of Section 6.2.1 that will be used to control the fusion process. Fig. 6.3 shows the color image and the confidence maps for a few sample scenes. The second column shows the ToF confidence, the proposed approach is able to assign a low confidence (darker pixels in the figure) to the areas with a larger error. A first observation is that in most of the confidence maps the error is larger in proximity of the edges. It is a well-known issue of ToF sensors due to the limited resolution and to the mixed pixels effect. Furthermore the CNN is also able to detect that the ToF error is higher on dark surfaces due to the lower reflection (e.g., on the dark furniture in row 3). The MPI distortion is more challenging to be detected however, by looking at the fruits in row 4, it is possible to see that the confidence is lower in their bottom part touching the table, similarly in the angle between the wall and the sink in row 5, where the multi-path is generated by rays bouncing from one surface to the other.

Concerning the stereo confidence, results are also good. As in the previous case, the limited accuracy on edges is correctly recognized. Furthermore, surfaces with uniform patterns (e.g., the flat panel on the right in row 4) or reflective ones (e.g., the pots in row 1) have lower confidence as expected.

The confidence information is then used to drive the fusion algorithm. The output disparity for some sample scenes is shown in Fig. 6.4. Column 1 shows a color view of the scene while column 2 contains the ground truth disparity data. The up-sampled, filtered and reprojected ToF data are shown in column 3 while column 4 contains the corresponding error map. The error is computed as the ground truth disparity minus the estimated disparity. Notice how ToF data are in general more accurate than the stereo one although some limitations of ToF sensors are visible. In particular, the data in proximity of edges are not too accurate. Furthermore the acquisition on low-reflective surfaces is more noisy and the multi-path error affects some regions close to boundaries between touching surfaces.

Columns 5 and 6 show the disparity and the error map for the SGM stereo vision algorithm. For this work we used the OpenCV implementation of the SGM stereo algorithm with pointwise Birchfield-Tomasi metric, 8 paths for the optimization and a window size of 7×7 [pxl]. Edge regions are challenging also for stereo vision even if they are more accurate than the ToF acquisitions due to the higher resolution.

On the other hand, some regions proved to be critical for the stereo algorithm, e.g., regions with a limited amount of texture (like the flat panel on the right in row 4) or strongly reflective regions (e.g., the pots in row 1).

Finally, the fused disparity maps and their relative error are shown in columns 7 and 8. The fusion algorithm tries to extract the most accurate information from both sources and generally it provides depth maps with less artifacts on edges but at the same time free from the various artifacts of the stereo acquisition.


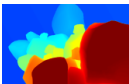
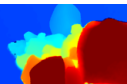
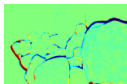
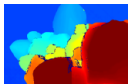

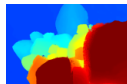


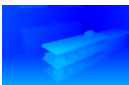
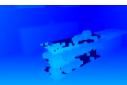
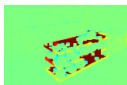
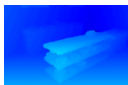

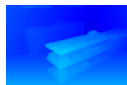



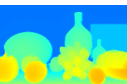
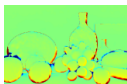
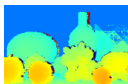




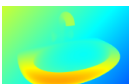
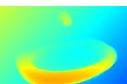
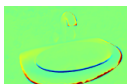
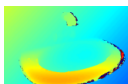
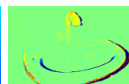
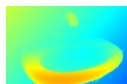
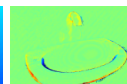

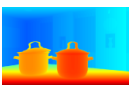
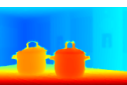
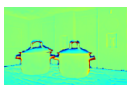
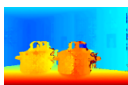
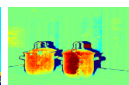
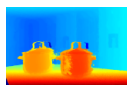

Input Scene		ToF		Stereo		Fusion	
Color view	Ground truth	Disparity	Error	Disparity	Error	Disparity	Error
							
							
							
							
							

Fig. 6.4: Results of the proposed fusion framework on 5 sample scenes (one for each row). The disparity images are depicted in the range between 0 (dark blue) to 200 [pxl] (dark red). The error images are depicted in the range between -10 (dark blue) to 10 [pxl] (dark red), the absence of error is represented in green (best viewed in color).

The numerical evaluation of the performance is shown in Table 6.1 and confirms the visual evaluation. The table shows both the Mean Absolute Error (MAE) and the Mean Squared Error (MSE) in disparity space averaged on all the 15 scenes. For a fair comparison, we considered as valid pixels for the results only the ones having a valid disparity value in all the compared disparity maps (stereo, ToF and fused disparities). By looking at the averaged MSE values, the ToF sensor has a

<i>Method</i>	<i>MAE</i> [pxl]	<i>MSE</i> [pxl ²]
Interpolated ToF	0.66	4.75
SGM Stereo	0.79	13.54
Marin et al. [67]	0.64	4.20
Proposed Method	0.53	3.92

Table 6.1: Mean Absolute Error (MAE) and Mean Squared Error (MSE) in disparity units with respect to the ground truth for the ToF and stereo data, the proposed method and [67] on the SYNTH3 dataset. The error has been computed only on non-occluded pixels for which a disparity value is available in all the methods.

high accuracy with a MSE of 4.75 [pxl²], much smaller than the MSE of 13.54 [pxl²] of the stereo system. The MAE is 0.66 and 0.79 [pxl] respectively, with a more limited gap due to the fact that the stereo system has some large errors that have a larger impact with the squared measure.

However, confidence data allow to select at most pixel locations the best source and thus to exploit the strengths of both stereo and ToF acquisitions. The proposed approach is able to obtain a MSE of 3.92 [pxl²] and a MAE of 0.53 [pxl], a very good result with a noticeable improvement with respect to both sensors. Comparison with state-of-the-art approaches on this dataset is limited by the lack of available implementations of the competing approaches. However, we compared our approach with the highly performing method of Marin et al. [67]. This approach uses the same LC fusion framework used in the proposed method, but it uses different analytical confidence measures for the ToF and stereo data. It has a MSE of 4.20 [pxl²], higher than the one of the proposed method. The method of [67] outperforms most state-of-the-art approaches, so also the performance of the proposed method are expected to be competitive with the better performing schemes, as demonstrated by the comparison on the LTTM5 dataset in Section 6.2.4.

Evaluation on the REAL3 Dataset

The testing on synthetic data does not take into account all the potential issues that can arise when working with real world data and sensors. For this reason, we tested the proposed approach also on real world data using the REAL3 dataset presented in Section 3.2. Notice that, due to the limited size of the real world

dataset, in this experiment we used the network trained on the synthetic dataset to compute the confidence maps used to drive the fusion process. As pointed out in Section 3.2, the real world dataset contains 8 different scenes (see Fig. 3.4 for their thumbnails). The scenes are simpler than the synthetic ones due to the challenges in practical data acquisition (specially for what concerns the acquisition of ground truth information), however they contain regions with different amount of texture information, repeating patterns critical for stereo approaches, different materials, bright and dark objects and some complex geometries (e.g., in the plant scene). Similarly to the synthetic data case, the algorithm takes in input the 512×424 [pxl] depth and amplitude maps from the Kinect v2 sensor and the two 960×540 [pxl] color images obtained by subsampling by a factor of 2 and rectifying the two color views from the ZED camera (see sections 6.2 and 3.2). In this case the output is computed on the point of view of the left camera at the 960×540 [pxl] resolution of color data and compared with the ground truth from the same viewpoint. The estimated disparities have also been cropped to highlight only the region that is framed by all the three cameras.

We start the evaluation from the confidence information: in this case, the task is more challenging since the CNN is trained on synthetic data and then evaluated on the real data, which have slightly different properties. However, the proposed deep network proved to have quite good generalization properties and the estimated confidence, although not as precise as in the synthetic case due to the *domain shift* issue, is able to underline the key sources of error as can be seen from the examples in Fig. 6.5. Confidence information for ToF data is shown in the second column, it is possible to note that the CNN properly predicts a higher ToF error in proximity of edges. Also other critical aspects are properly identified, for example in row 2 it is possible to see how black areas in the pattern have a weaker reflection and lead to a less accurate acquisition while in row 3 the CNN properly detects that the acquisition of the plant is very critical for the ToF sensor. The third column contains the confidence maps for stereo data, notice how the CNN is able to recognize that highly textured regions are properly acquired while uniform surfaces like the white walls are critical for the stereo algorithm. However, the confidence estimation on real data appears to be noisier than in the synthetic case, and this is due to the slightly different nature between simulated and real sensor data. These results

could be improved by using domain adaptation methods as the one proposed in Section 5.3.

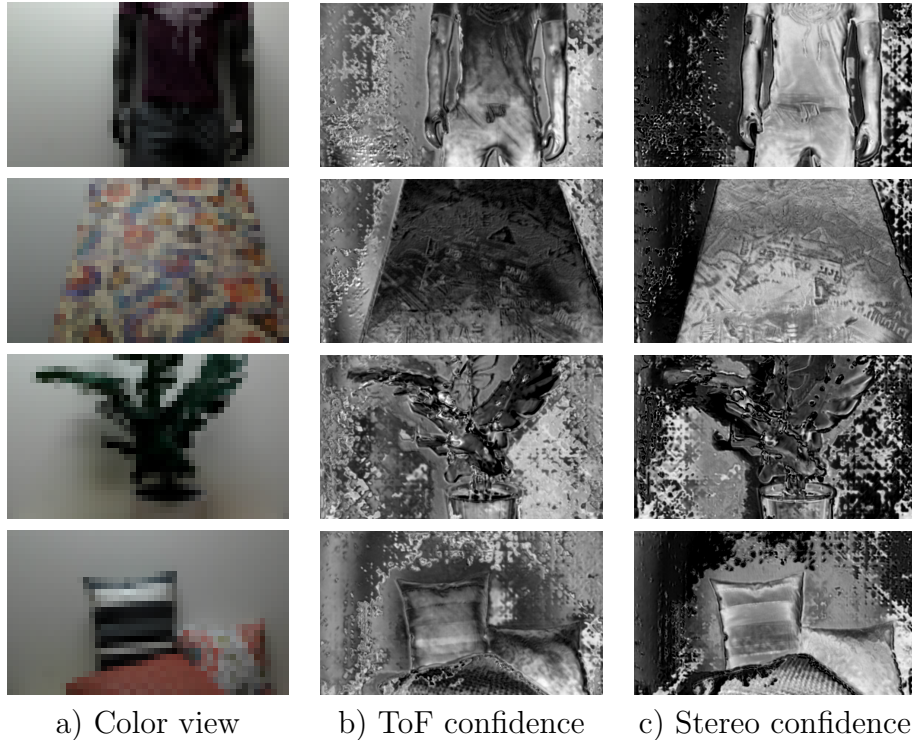


Fig. 6.5: Confidence information computed by the proposed deep learning architecture for some sample scenes: a) Color view; b) Estimated ToF confidence; c) Estimated stereo confidence. Brighter areas correspond to higher confidence values, while darker pixels to low confidence ones.

The numerical results of the fusion algorithm are reported in Table 6.2 while Fig. 6.6 shows the output depth maps and the error maps for some sample scenes. The figure is organized as in the previous experiment. Column 1 and 2 show a color view of the scene and the ground truth disparity data. The up-sampled, filtered and reprojected ToF data are shown in column 3, while column 4 contains the corresponding error map. It is possible to notice that also in this case ToF data are not very precise in proximity of edges, but there is a small amount of error also on flat surfaces due to the noise of the sensor and to inaccuracies in the reprojection operation (in this case it is based on calibration information while previously the cameras were ideally placed).

Columns 5 and 6 show the disparity estimated by the stereo vision algorithm and the corresponding error map. Stereo data have sharper edges and a good accuracy on the objects in the foreground but there are artifacts on low-textured regions, specially on the white walls on the background.

The fused disparity map and its relative error are shown in columns 7 and 8. The fusion algorithm reliably fuses the information coming from the two sensors being able to properly reconstruct the edges using the stereo data but also correctly estimating the background that instead is better acquired by the ToF sensor.

Input Scene		ToF		Stereo		Fusion	
Color view	Ground truth	Disparity	Error	Disparity	Error	Disparity	Error

Fig. 6.6: Results of the proposed fusion framework on some sample scenes from the REAL3 dataset. The disparity images are depicted in the range between 0 (dark blue) to 100 [pxl] (dark red). In the ground truth disparity maps, the unlabeled pixels are highlighted in dark blue. The error images are depicted in the range between -10 (dark blue) to 10 [pxl] (dark red), the absence of error is represented in green (best viewed in color).

The numerical evaluation of the performance is shown in Table 6.2 and confirms the visual analysis. The table shows the MAE and the MSE in disparity space averaged on all the 8 scenes. For a fair comparison, we considered as valid pixels for the results only the ones having a valid disparity value in all the compared disparity maps (stereo, ToF and fused disparities). By looking at the MAE values, the ToF sensor has a high accuracy with an error of 2.55 [pxl], much smaller than the MAE of 7.98 [pxl] of the stereo system (and the MSE difference is even larger). This is

a challenging situation for fusion algorithms since it is difficult to improve the data from the best sensor without affecting it with errors from the other one. However, the use of confidence data helps to properly combine both sources of information obtaining a MAE of 1.65 [pxl] with a noticeable improvement with respect to both sensors. The method of Marin et al. [67] on this dataset has a MAE of 2.19 [pxl], again higher than the one obtained with the proposed method.

<i>Method</i>	<i>MAE</i> [pxl]	<i>MSE</i> [pxl ²]
Interpolated ToF	2.55	10.76
SGM Stereo	7.98	201.64
Marin et al. [67]	2.19	8.82
Proposed Fusion	1.65	8.35

Table 6.2: MAE and MSE in disparity units with respect to the ground truth for the ToF and stereo data, the proposed method and [67] on the REAL3 dataset. The error has been computed only on non-occluded pixels for which a disparity value is available in all the methods.

Evaluation on the LTTM5 Dataset

Finally, we tested the proposed approach on the LTTM5 dataset. This dataset has been introduced in [61] and contains 5 different scenes acquired with a MESA SR4000 ToF sensor and two Basler color cameras (the scene thumbnails are in the first column of Fig. 6.7). Even if it is smaller than the other two datasets and the ToF data has been acquired with a camera with lower performance (the resolution is just 176×144 [pxl]), this dataset represents an interesting benchmark since it has been used for the evaluation of several works and allows to perform the comparison with different state-of-the-art methods from the literature. Furthermore it contains object with various shapes and characteristics that allow to evaluate the method in various situations including depth discontinuities, materials with different reflectivity and both textured and un-textured surfaces. In order to process this dataset the algorithm takes in input the 176×144 [pxl] depth and amplitude maps from the MESA sensor and the two 1032×778 [pxl] color images from the Basler cameras and computes the output from the point of view of the left camera at the same resolution of color data. For the confidence estimation, we used the CNN

trained on synthetic data from the SYNTH3 training set as for the other datasets.

Fig. 6.7 shows the confidence information for the ToF and stereo sensors on this dataset. This situation is even more challenging since the ToF camera used for this dataset has very different properties from the simulated one used in the training. The accuracy of confidence information is lower, however the proposed approach is able to detect some key issues. Concerning ToF data it is possible to notice the lower confidence in proximity of edges and that the depth information is less reliable on the complex geometries of the objects if compared with the walls and table. Stereo data are also less reliable on edges and on regions with a lower amount of texture.

Concerning the results of the fusion of the two sensors, Fig. 6.8 shows the output depth maps and the error maps for the 5 scenes of the dataset. It is possible to notice the good accuracy of fused data on edges and how the algorithm is able to properly choose the best data source in many situations avoiding the artifacts of the two acquisition devices. For example, the repeating pattern on the green box causes errors in the stereo reconstruction that are not present in the fused data. On the other hand, the upper part of the table is very critical for the ToF sensor due to the multi-path and to the surface orientation. In the fused disparity, even if not perfect, it is better reconstructed thanks to the information coming from stereo vision.

Table 6.3 reports the numerical values for the error and the comparison with some state-of-the-art methods from the literature.

The compared state-of-the-art methods are based on different strategies: the method of [139] uses an iterative approach and bilateral filtering. Then we considered two approaches based on probabilistic MAP-MRF schemes, i.e., [125] and [61]. Finally, there are the two previous approaches based on the LC framework, i.e. [133] and [67]. For a fair comparison, we considered as valid pixels for the results only the ones having a valid disparity value in all the compared disparity maps (stereo, ToF and fused disparities from the various methods). By looking at the average values, on this dataset the ToF and stereo sensors have a similar MAE of 1.53 and 1.45 while the MSE is much lower for the ToF sensor (this is due to the fact that the stereo data has some large errors while ToF error is more uniformly distributed).

The proposed approach achieves a MAE of 0.89 [pxl], that is better than all



Fig. 6.7: Confidence information computed by the proposed deep learning architecture for the scenes in the LTTM5 dataset: a) Color view; b) Estimated ToF confidence; c) Estimated stereo confidence. Brighter areas correspond to higher confidence values, while darker pixels to low confidence ones.

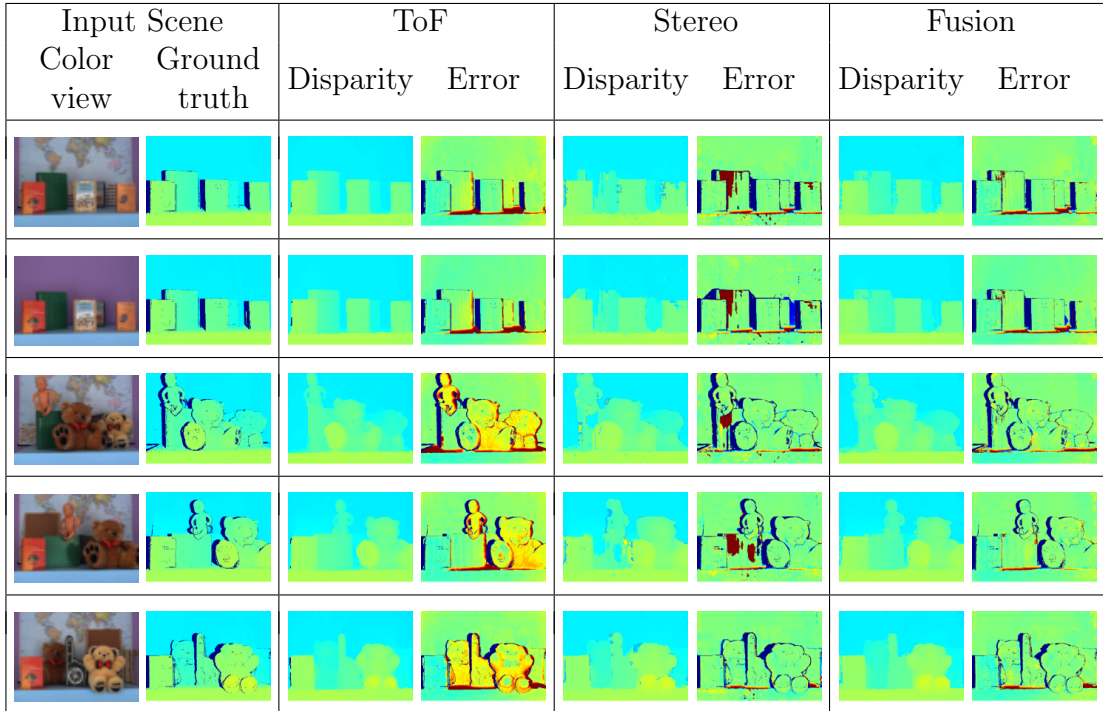


Fig. 6.8: Results of the proposed fusion framework on the 5 scenes of the LTTM5 dataset. The disparity images are depicted in the range between 0 (dark blue) to 100 [pxl] (dark red). The error images are depicted in the range between -10 (dark blue) to 10 [pxl] (dark red), the absence of error is represented in green (best viewed in color).

<i>Method</i>	<i>MAE</i> [pxl]	<i>MSE</i> [pxl ²]
Interpolated ToF	1.53	11.68
SGM Stereo	1.45	20.42
Dal Mutto et al. (LC) [133]	1.36	10.06
Marin et al. [67]	1.15	7.67
Yang et al. [139]	1.59	10.98
Zhu et al. [125]	1.59	11.13
Dal Mutto et al. (MRF) [61]	1.43	12.21
Proposed Fusion	0.89	7.40

Table 6.3: MAE and MSE averaged on the 5 scenes of the LTTM5 dataset (in disparity units) with respect to the ground truth, computed only on non-occluded pixels for which a disparity value is available in all the methods.

the proposed approaches with a large margin. The best among the compared approaches is [67], that has a MAE about 25% higher, while all the other compared approaches have a larger error. If using the MSE as error metric the gap with [67] is smaller while it remains large with respect to all the other approaches. This is due to the fact that [67] relies strongly on ToF data that has a better MSE while the proposed approach makes a more balanced use of the two sources of information. In any case, the proposed approach has the best performance among all the compared ones according to both measures.

Ablation Studies

Finally, we performed some further tests in order to evaluate the impact on the fusion accuracy of the information coming from the various input sources. Specifically, we made an additional set of experiments where, in turn, we selectively removed one of the four input sources of Section 6.2.1 in order to better evaluate its contribution to the final output. The results are shown in Table 6.4 and indicate how, on average, the combination of all inputs offers the best performance. More in detail, the first two rows show how each of the two disparities contains relevant information for the corresponding sensor. Both the removal of the ToF or stereo disparity leads to a quite large decrease in terms of fusion accuracy (around 20%). The impact of the ToF amplitude is smaller, but it has a noticeable effect on the SYNTH3 and LTTM5 datasets. The main issue with ToF amplitude is that it depends a lot on the employed sensor while the ToF simulator is not able to model in a completely accurate way the amplitude data acquired by real world sensors. Finally, the difference map Δ between the reference image and the target one reprojected over it proved to be very useful specially in real world datasets where the stereo matching is less reliable. Concluding, even if not all information types are fundamental for all datasets, the combination of all the four sources is the best solution in order to have an approach with very good performance on both real and synthetic data.

Another idea we exploited is to jointly estimate the confidence of the two sensors instead of independently computing the two confidence maps. We evaluated the impact of this approach by trying to estimate the ToF and stereo confidence separately with two different CNNs with a single output (keeping fixed the other

<i>Ablation Study</i>	<i>SYNTH3</i>	<i>REAL3</i>	<i>LTTM5</i>
All inputs (proposed method)	0.53	1.65	0.89
Without ToF disparity (no D_T)	0.64	2.04	1.179
Without Stereo disparity (no D_S)	0.66	2.11	1.19
Without ToF amplitude (no A_T)	0.55	1.6	0.915
Without LR difference (no Δ)	0.52	2.84	1.19
Separate estimation ToF/stereo conf.	0.61	1.93	1.156
Select highest confidence (HC)	0.43	1.90	1.11
Weighted average (WA)	0.46	2.44	1.07

Table 6.4: Mean Absolute Error (MAE) [pxl] on the fused depth maps (in disparity units) when removing different input channels, with separate stereo and ToF confidence estimation and with different fusion strategies.

parameters). As shown by the last row of the table, this approach leads to worse performance on all the three datasets, demonstrating that the joint estimation allows to obtain more coherent confidence maps and thus a better accuracy of the fused data.

Finally, in order to evaluate the impact of the Locally Consistent (LC) fusion algorithm we tried also to exploit the confidence data estimated with the proposed deep learning architecture into simpler fusion strategies. We tried two simple solutions, the first is the selection at each location of the source with the highest confidence (HC) and the second is the usage of a weighted average (WA) of the ToF and stereo disparities with the weights given by the estimated confidences at each pixel location. The obtained results are in the last two rows of Table 6.4. On synthetic data, where the confidence information is very reliable and the noise on the data is limited, even by just selecting at each pixel location the source with the highest estimated confidence it is possible to obtain very good results with a MAE of 0.43, even better than the one achieved by the LC algorithm. Also the weighted average driven by confidence allows to obtain a very good result with a MAE of 0.46. This proves the reliability of the proposed confidence estimation algorithm. While on synthetic data the LC refinement is not really necessary, the discussion is quite different on real world data. On the REAL3 dataset the selection of the source with the highest confidence and the weighted average achieve a MAE of 1.9 and 2.44 respectively, quite higher than the result of the full version of the proposed

approach with LC (which achieves a MAE of 1.6 [pxl]). A similar discussion holds for the LTTM5 dataset (the absolute errors are 1.11 [pxl] for HC and 1.07 [pxl] for WA against 0.89 [pxl] for LC). This proves how the smoothing and regularization of the choices performed by LC is very useful when data are more noisy and less reliable as it happens in real world acquisitions. On the other hand, simpler fusion strategies might be preferable when a fast computation is needed or data are very reliable.

Chapter 7

Conclusions

This thesis collects the results of the research that I have carried out during my three years of Ph.D. at University of Padova from October 2016 to September 2019. In particular, my work has been focused on the development of methods for the denoising of depth data acquired ToF cameras eventually supported by stereo vision systems.

An initial step of my work was to collect depth datasets acquired with commercial depth sensors with the related true depth maps. Thanks to these we were able to train data driven denoising methods and to test their performance.

A part of my research was devoted to the reduction of the distortion due to *multi-path interference* and the zero-mean noise on stand alone ToF cameras. Two methods for this task have been proposed. The first is based on a hardware customization of the ToF projector, that is modified to spatially modulate the ToF light signal. This customization allows to estimate a depth map of the scene using a *structured light* principle on the ToF device. This depth map is then fused, using a *maximum likelihood* criterion, with a second depth map, estimated with the *time-of-flight* principle, to have a more accurate depth estimation. A second approach exploits a *convolutional neural network* that takes as input data acquired by a multi-frequency ToF sensor to correct MPI and reduce the noise of the depth recordings. Due to the complexity of gathering enough data for the training of such network, a synthetic dataset has been used for this task. However, this showed the *domain shift* issue in training a network on synthetic data and testing it on real data. For

this reason, we have started to investigate how to improve the performance of the denoiser network on real data without using real ground truth. We designed a novel *unsupervised domain adaptation* technique for this task. This approach showed well established performance at adapting the denoiser to work on real data. Moreover, we developed another method for *unsupervised domain adaptation* in the task of semantic segmentation, another field where it is expensive to collect real data with ground truth information, which in this case is the semantic annotation of the given color images.

Another approach to improve the accuracy of the scene depth estimation is to fuse the data collected by multiple depth sensors simultaneously. In particular, we focused on the case in which a ToF camera and a stereo vision system are used together. We made this choice since these two typologies of sensors share complementary strengths and flaws. In this field, we proposed a method to combine the two data sources using a fusion method that enforces the *local consistency* of the data and which is guided by the reliability measures of the two depth data estimated by a *convolutional neural network*. This method proved to have state-of-the-art performance on both synthetic and real datasets.

Future work will be focused on domain adaptation methods, as the one we presented for ToF depth data refinement, in the case of stereo-ToF fusion. Other techniques for domain adaptation could be implemented for example by translating depth data from the synthetic to the real domain or working directly in the network feature space to reduce the *domain shift* issue. Regarding the proposed STM-ToF method, the future work will involve a possible implementation of a real device able to reproduce the analysed model for MPI correction. Furthermore, more advanced fusion techniques for the structured light and ToF depth maps could be investigated in order to solve the current limitations. New methods for MPI correction could exploit neural networks for the estimation of the light back-scattering vector. This will allow to get the scene impulse response as done by the SRA method, that is instead implemented with an analytical approach. Regarding the stereo-ToF fusion research, a novel end-to-end network could be implemented and tested in order to avoid the use of the complex LC fusion algorithm.

Appendix A

Error Propagation Analysis

Here, the error propagation analysis used for the evaluation of the error acting on the stereo and ToF depth acquisition is introduced. The error propagation analysis links the amount of uncertainty of the output of a given function $f(\cdot)$ with the amount of uncertainty of its arguments. Given a function $f(\cdot)$ that is continuous and has continuous first and second order derivatives with respect to the random arguments $\{V_i\}_{i=0}^{N-1}$ on some of their neighbourhoods, then the random variable $w = f(\{V_i\}_{i=0}^{N-1})$ is asymptotically *Normal*. If these arguments of $f(\cdot)$ are random variables respectively with mean μ_{V_i} and variance $\sigma_{V_i}^2$, whose deviation from the mean is symmetric and bell-shaped, and assuming that they are each other independent, then the variance of the function $f(\cdot)$ can be approximated as [140]:

$$\sigma_f^2 = \sum_{i=0}^n \left(\frac{\partial f}{\partial V_i} \right)^2 \sigma_{V_i}^2. \quad (\text{A.1})$$

Appendix B

Noise Variance due to Photon Shot Noise on ToF Recordings

By using the error propagation introduced in Appendix A, an evaluation of the noise acting on ToF depth estimation can be retrieved. In the following of this analysis, each sample of the ToF correlation function, measuring the amount of received photons in the integration time and described by Eq. 2.7, will be labeled as $N_i = c(\omega_r \tau_i)$ with $\omega_r \tau_i = \frac{2\pi}{4}i$ for $i = 0, \dots, 3$. Due to the random nature of the light that is assumed to be affected by *photon shot noise*, it is possible to assume that the ToF correlation samples have a *Poisson distribution* with mean μ_{V_i} and variance $\sigma_{V_i}^2$ equal to N_i [141], i.e., the number of photons accumulated during the correlation sample acquisition.

From Eq. A.1, the noise variance of ϕ due to the noise acting on the correlation function recording is

$$\sigma_\phi^2 = \sum_{i=0}^3 \left(\frac{\partial \phi}{\partial N_i} \right)^2 \sigma_{N_i}^2. \quad (\text{B.1})$$

Since ϕ can be computed from Eq. 2.8, where the function `atan2` is just an *arctangent* (`atan`) function with periodicity extended from a maximum range of π to a maximum range of 2π (just a constant π term is added if required), the following analysis will be carried out in the case of `atan` for simplicity, but it is still valid for the `atan2` case.

APPENDIX B. NOISE VARIANCE DUE TO PHOTON SHOT NOISE ON TOF RECORDINGS

By computing ϕ as

$$\phi = \text{atan}\left(\frac{N_3 - N_1}{N_0 - N_2}\right), \quad (\text{B.2})$$

defining $X = \frac{N_3 - N_1}{N_0 - N_2}$, we can express it as $\phi = \text{atan}(X)$ and so Eq. B.1 can be reformulated as

$$\begin{aligned} \sigma_\phi^2 &= \sum_{i=0}^3 \left(\frac{\partial \text{atan}(X)}{\partial N_i}\right)^2 \sigma_{N_i}^2 \\ &= \left(\frac{\partial \text{atan}(X)}{\partial X}\right)^2 \sum_{i=0}^3 \left(\frac{\partial X}{\partial N_i}\right)^2 \sigma_{N_i}^2 \\ &= \left(\frac{1}{1+X^2}\right)^2 \sum_{i=0}^3 \left(\frac{\partial X}{\partial N_i}\right)^2 \sigma_{N_i}^2. \end{aligned} \quad (\text{B.3})$$

Since $X = \tan \phi$ and $\frac{1}{1+\tan^2 \phi} = \cos^2 \phi$, we can rewrite (B.3) as:

$$\begin{aligned} \sigma_\phi^2 &= \cos^4 \phi \sum_{i=0}^3 \left(\frac{\partial X}{\partial N_i}\right)^2 \sigma_{N_i}^2 \\ &= \cos^4 \phi \left[\frac{(N_3 - N_1)^2}{(N_0 - N_2)^4} N_0 + \frac{1}{(N_0 - N_2)^2} N_1 + \dots \right. \\ &\quad \left. \dots + \frac{(N_3 - N_1)^2}{(N_0 - N_2)^4} N_2 + \frac{1}{(N_0 - N_2)^2} N_3 \right] = \\ &= \cos^4 \phi \left[\frac{(N_3 - N_1)^2}{(N_0 - N_2)^4} N_0 + \frac{1}{(N_0 - N_2)^2} N_1 + \dots \right. \\ &\quad \left. \dots + \frac{(N_3 - N_1)^2}{(N_0 - N_2)^4} N_2 + \frac{1}{(N_0 - N_2)^2} N_3 \right] = \\ &= \cos^4 \phi \left[\frac{(N_3 - N_1)^2 (N_0 + N_2)}{(N_0 - N_2)^4} + \frac{N_1 + N_3}{(N_0 - N_2)^2} \right] \end{aligned} \quad (\text{B.4})$$

By using Eq. 2.7 in Eq. B.4, it comes out that

$$\begin{aligned}
\sigma_{\phi}^2 &= \cos^4 \phi \left[\frac{(2A \cos(\phi + \frac{3\pi}{2}))^2 (2B)}{(2A \cos \phi)^4} + \frac{2B}{(2A \cos \phi)^2} \right] = \\
&= \cos^4 \phi \left[\frac{4A^2 (1 - \cos^2 \phi) (2B)}{(2A \cos \phi)^4} + \frac{2B}{(2A \cos \phi)^2} \right] = \\
&= \cos^4 \phi \frac{4A^2 (1 - \cos^2 \phi) (2B) + 4A^2 \cos^2 \phi (2B)}{(2A \cos \phi)^4} = \\
&= \frac{B}{2A^2}.
\end{aligned} \tag{B.5}$$

Given the effect of the photon shot noise on the ϕ evaluation and that the final depth can be computed by means of Eq. 2.5, the noise variance on ToF depth is

$$\sigma_{ps}^2 = \left(\frac{c_l}{4\pi f_{mod}} \right)^2 \frac{B}{2A^2} \tag{B.6}$$

Appendix C

STM-ToF Correlation Function Evaluation and Error Propagation Analysis

In this appendix, the correlation function for the STM-ToF system introduced in Section 4.2.1, i.e., Eq. 4.2, will be derived.

Referring to the acquisition model of Section 4.2.1, the signal emitted by pixel (x, y) of the projector when the sample $\omega_r \tau_i$ of the correlation function has to be computed is given by the combination of the standard ToF modulation of Eq. 2.3 with the modulating pattern, $L_{x,y}(\omega_r \tau_i)$, of Eq. 4.1:

$$\begin{aligned} s_t(t, \omega_r \tau_i) &= L_{x,y}(\omega_r \tau_i) \cdot s_{stdToF,t}(t) \\ &= \frac{1}{4} a_t \left(1 + \cos(l\omega_r \tau_i - \theta_{x,y}) \right) \left(1 + \sin(\omega_r t) \right). \end{aligned} \quad (\text{C.1})$$

Each pixel of the ToF camera receives a light signal that can be modeled as:

$$\begin{aligned} s_r(t, \omega_r \tau_i) &= b_r + \frac{1}{4} a_r \left(1 + \cos(l\omega_r \tau_i - \theta_{x,y}) \right) \left(1 + \sin(\omega_r t - \phi_d) \right) + \dots \\ &\dots + b_{r,g} + a_{r,g} \cdot \cos(\omega_r \tau_i - \phi_g) \end{aligned} \quad (\text{C.2})$$

where b_r is the light offset due to the ambient light, $a_r = \alpha a_t$, with α equal to the channel attenuation, $b_{r,g}$ and $a_{r,g}$ are respectively the light offset and amplitude

of the global component of the light, and ϕ_d is the phase displacement between the transmitted and received direct part of the signal. The scene depth d can be computed from ϕ_d through the well known relation $d = \frac{\phi_d c_l}{2\omega_r}$ where c_l is the speed of light. The second line of Eq. C.2 contains the global component of the received light signal, the one related to MPI and it is assumed to be not influenced by the projected pattern in case of diffuse reflections [57].

The ToF pixels are able to compute the correlation function between the received signal and a reference one, e.g., a rectangular wave at the same modulation frequency $rect_{\omega_r}(t) = H(\sin(\omega_r t))$, where $H(\cdot)$ represents the Heaviside function. The correlation function sampled in $\omega_r \tau_i \in [0; 2\pi)$ can be modeled in this acquisition scenario as

$$c(\omega_r \tau_i) = \int_0^{\frac{1}{f_{mod}}} s_r(t, \omega_r \tau_i) H(\sin(\omega_r t + \omega_r \tau_i)) dt \quad (C.3)$$

and by substituting Eq. C.2 inside C.3, we obtain Eq. 4.2:

$$c(\omega_r \tau_i) = B + A \cos(\omega_r \tau_i + \phi_d) + A_g \cos(\omega_r \tau_i + \phi_g) + \frac{\pi A}{2} \cos(l\omega_r \tau_i - \theta) + \dots \\ + \frac{A}{2} \left[\cos((l-1)\omega_r \tau_i - \phi_d - \theta) + \cos((l+1)\omega_r \tau_i + \phi_d - \theta_{x,y}) \right] \quad (C.4)$$

where $B = \frac{1}{f_{mod}} \left(\frac{b_r}{2} + \frac{a_r}{8} + \frac{b_{r,g}}{2} \right)$ is an additive constant that represents the received light offset, $A = \frac{a_r}{4\pi f_{mod}}$ is proportional to the power of the direct component of the received light, $A_g = \frac{a_{r,g}}{\pi f_{mod}}$ is proportional to the power of the global component of the received light. The correlation function values $c(\omega_r \tau_i)$ are a measure of the number of photons received by the pixel during the considered integration time.

C.1 Error Propagation Analysis on STM-ToF

In Section 4.2.2, we have presented a model for the estimation of the impact of the photon shot noise on ToF correlation samples on the depth estimation. In this section, we present the mathematical derivation based on error propagation analysis

(Appendix A) that we used to compute Eq. 4.5.

In the following of this analysis, each sample of the correlation function described by Eq. 4.2 will be labeled as $N_i = c(\omega_r \tau_i)$ with $\omega_r \tau_i = \frac{2\pi}{9}i$ for $i = 0, \dots, 8$. Due to the random nature of the light that is assumed to be affected by *photon shot noise*, it is possible to assume that the ToF correlation samples have a *Poisson distribution* with mean μ_{N_i} and variance $\sigma_{N_i}^2$ equal to N_i [141], i.e., the number of photons accumulated during the correlation sample acquisition.

Since $\phi_d = \frac{\varphi_4 - \varphi_2}{2}$, as a first step we are going to estimate the noise variance for φ_k with $k = 1, \dots, 8$, using the error propagation analysis given by Eq. A.1, that is

$$\sigma_{\varphi_k}^2 = \sum_{i=0}^8 \left(\frac{\partial \varphi_k}{\partial N_i} \right)^2 \sigma_{N_i}^2. \quad (\text{C.5})$$

Since from Fourier analysis

$$\varphi_k = \arctan \left(- \frac{\sum_{i=0}^8 N_i \sin \frac{2\pi}{9} ki}{\sum_{i=0}^8 N_i \cos \frac{2\pi}{9} ki} \right), \quad (\text{C.6})$$

by taking $X_k = - \frac{\sum_{i=0}^8 N_i \sin \frac{2\pi}{9} ki}{\sum_{i=0}^8 N_i \cos \frac{2\pi}{9} ki}$, we can express it as $\varphi_k = \arctan(X_k)$ and so Eq. C.5 can be reformulated as

$$\begin{aligned} \sigma_{\varphi_k}^2 &= \sum_{i=0}^8 \left(\frac{\partial \arctan(X_k)}{\partial N_i} \right)^2 \sigma_{N_i}^2 \\ &= \left(\frac{\partial \arctan(X_k)}{\partial X_k} \right)^2 \sum_{i=0}^8 \left(\frac{\partial X_k}{\partial N_i} \right)^2 \sigma_{N_i}^2 \\ &= \left(\frac{1}{1 + X_k^2} \right)^2 \sum_{i=0}^8 \left(\frac{\partial X_k}{\partial N_i} \right)^2 \sigma_{N_i}^2. \end{aligned} \quad (\text{C.7})$$

Since $X_k = \tan \varphi_k$ and $\frac{1}{1 + \tan^2 \varphi_k} = \cos^2 \varphi_k$, we can rewrite (C.7) as:

$$\sigma_{\varphi_k}^2 = \cos^4 \varphi_k \sum_{i=0}^8 \left(\frac{\partial X_k}{\partial N_i} \right)^2 \sigma_{N_i}^2. \quad (\text{C.8})$$

After evaluating the partial derivatives $\frac{\partial X_k}{\partial N_i}$ and computing the summation in

APPENDIX C. STM-TOF CORRELATION FUNCTION EVALUATION AND
ERROR PROPAGATION ANALYSIS

(C.8), it results that the error variance for the estimation of the phase φ_k is equal to:

$$\sigma_{\varphi_k}^2 = \frac{2}{9I_k^2} \left[B - \frac{1}{2} I_{2k} \cos(2\varphi_k) \cos(\varphi_{2k}) - \frac{1}{2} I_{2k} \sin(2\varphi_k) \sin(\varphi_{2k}) \right], \quad (\text{C.9})$$

where we labeled with I_i the amplitude of the sinusoidal wave at frequency i in the correlation function given by Eq. 4.2. Moreover, the variances of the phase estimation error for phases φ_2 , φ_3 and φ_4 are respectively

$$\sigma_{\varphi_2}^2 = \frac{8}{9A^2} \left[B - \frac{A}{4} \cos(-3\phi_d - \theta) \right], \quad (\text{C.10})$$

$$\sigma_{\varphi_3}^2 = \frac{8}{9\pi^2 A^2} \left[B - \frac{\pi A}{4} \cos(3\theta) \right], \quad (\text{C.11})$$

$$\sigma_{\varphi_4}^2 = \frac{8}{9A^2} \left[B - \frac{D}{2} \cos(2\phi_d - 2\theta + \phi_{FF}) \right]. \quad (\text{C.12})$$

where

$$D = \sqrt{A + A_g + 2A \cdot A_g \cos(\phi_d - \phi_g)} \quad (\text{C.13})$$

$$\phi_{FF} = \text{atan2}(A \cos \phi_d + A_g \cos \phi_g, A \sin \phi_d + A_g \sin \phi_g). \quad (\text{C.14})$$

At this point it is possible to evaluate also the variance of the error for ϕ_d and θ , since $\phi_d = (\varphi_4 - \varphi_2)/2$ and $\theta' = -(\varphi_2 + \varphi_4)/2$ and since ϕ_2 and ϕ_4 are not independent, because they are computed from the same samples of the correlation function, we have

$$\begin{aligned} \sigma_{\phi_d}^2 &= \sum_{i=1}^n \left(\frac{\partial \phi_d}{\partial N_i} \right)^2 \sigma_{N_i}^2 = \frac{1}{4} \sum_{i=1}^n \left(\frac{\partial [\varphi_4 - \varphi_2]}{\partial N_i} \right)^2 \sigma_{N_i}^2 \\ &= \frac{1}{4} \left[\sum_{i=1}^n \left(\frac{\partial \varphi_4}{\partial N_i} \right)^2 \sigma_{N_i}^2 + \sum_{i=1}^n \left(\frac{\partial \varphi_2}{\partial N_i} \right)^2 \sigma_{N_i}^2 - 2 \sum_{i=1}^n \frac{\partial [\varphi_4 \cdot \varphi_2]}{\partial N_i} \sigma_{N_i}^2 \right] \\ &= \frac{1}{4} \left[\sigma_{\varphi_4}^2 + \sigma_{\varphi_2}^2 - 2\sigma_{\varphi_2 \varphi_4} \right] \end{aligned} \quad (\text{C.15})$$

In the same way it arises that

$$\sigma_{\theta'}^2 = \frac{1}{4} \left[\sigma_{\varphi_4}^2 + \sigma_{\varphi_2}^2 + 2\sigma_{\varphi_2\varphi_4} \right] \quad (\text{C.16})$$

For the evaluation of $\sigma_{\varphi_2\varphi_4} = \sum_{i=1}^n \frac{\partial[\varphi_4\varphi_2]}{\partial N_i} \sigma_{N_i}^2$ operations similar to the retrieval of $\sigma_{\varphi_k}^2$ can be applied and it follows that:

$$\begin{aligned} \sigma_{\varphi_2\varphi_4} = \frac{4}{9A^2} & \left[\cos \varphi_4 \cos \varphi_2 \left(\frac{A}{2} \cos \varphi_2 - \frac{\pi A}{2} \cos \varphi_3 \right) + \dots \right. \\ & \dots + \sin \varphi_4 \cos \varphi_2 \left(\frac{A}{2} \sin \varphi_2 + \frac{\pi A}{2} \sin \varphi_3 \right) + \dots \\ & \dots + \cos \varphi_4 \sin \varphi_2 \left(-\frac{A}{2} \sin \varphi_2 + \frac{\pi A}{2} \sin \varphi_3 \right) + \dots \\ & \left. \dots + \sin \varphi_4 \sin \varphi_2 \left(\frac{A}{2} \cos \varphi_2 + \frac{\pi A}{2} \cos \varphi_3 \right) \right]. \end{aligned} \quad (\text{C.17})$$

Putting together the solutions for $\sigma_{\varphi_2}^2$, $\sigma_{\varphi_4}^2$ and $\sigma_{\varphi_4+\varphi_2}$ through Eq. C.15 and C.16 it is possible to compute the variances of the direct phase and pattern phase offset estimation error. In particular, it is worth noticing that the mean value of these error variances (without considering the sinusoidal terms) are:

$$\bar{\sigma}_{\phi_d}^2 = \bar{\sigma}_{\theta'}^2 = \frac{4}{9A^2} B. \quad (\text{C.18})$$

If we compare the estimation noise variances for the pattern phase offset retrieval from the second and fourth harmonics or using only the third harmonic it comes out that:

$$\begin{aligned} \theta' = -(\varphi_2 + \varphi_4)/2 & \implies \bar{\sigma}_{\theta'}^2 = \frac{4}{9A^2} B \\ \theta = -\varphi_3 & \implies \bar{\sigma}_{\theta}^2 = \bar{\sigma}_{\varphi_3}^2 = \frac{8}{9\pi^2 A^2} B, \end{aligned} \quad (\text{C.19})$$

from which arises that the second formula gives an estimation of the phase offset that is about four times less noisy than the first one and for this reason we used the third harmonic to compute this parameter.

For the evaluation of the variance of the noise acting on the depth estimate employing the STM-ToF acquisition (the standard Whyte approach [51]), we have

to consider the linear relation that links the direct phase and the depth, indeed $d_{noMPI} = \frac{c}{4\pi f_{mod}} \phi_d$ and it results that

$$\sigma_{d_{noMPI}}^2 = \left(\frac{c}{4\pi f_{mod}} \right)^2 \sigma_{\phi_d}^2. \quad (C.20)$$

and by considering the mean variance of the noise we have

$$\bar{\sigma}_{d_{noMPI}}^2 = \left(\frac{c}{4\pi f_{mod}} \right)^2 \frac{4}{9A^2} B. \quad (C.21)$$

that is Eq. 4.5.

C.2 Structured Light Depth Estimation with Implicit Phase Unwrapping

In this section we are going to present the derivation for the implicit phase unwrapping in the *structured light* (SL) depth estimation that we used in Section 4.2.3.

As mentioned in Section 4.2.3, in case the phase offset θ_{ref} and θ_{target} are already phase unwrapped in θ_{ref}^{PU} and θ_{target}^{PU} , then the depth map with the SL approach can be estimated as

$$\begin{aligned} d_{SL} &= d_{ref} \left(1 + \frac{Q}{b} (\theta_{ref}^{PU} - \theta_{target}^{PU}) \right)^{-1} \\ &= d_{ref} \left(1 + \frac{Q}{b} (\theta_{ref} + 2\pi k_{ref} - \theta_{target} - 2\pi k_{target}) \right)^{-1} \end{aligned} \quad (C.22)$$

where $\theta_{target}, \theta_{ref} \in [-\pi; \pi)$ are the phase offsets directly accessible and $2\pi k_{target}$ and $2\pi k_{ref}$ are the offsets which correct the phase wrapping.

In order to unwrap the phase offsets, and so estimate $2\pi k_{target}$ and $2\pi k_{ref}$, it is usually required to project multiple patterns with lower frequencies on the scene. We avoid this by exploiting the depth map computed with the ToF sensor, d_{ToF} . First of all, we consider the pattern phase offset θ_{ToF}^{PU} that originates the depth map d_{ToF} in case of a SL acquisition:

$$\theta_{ToF}^{PU} = \theta_{ref}^{PU} - \frac{b}{Q} \cdot \frac{d_{ref} - d_{ToF}}{d_{ToF}}. \quad (C.23)$$

If we consider

$$\theta_{ToF} = \theta_{ref} - \frac{b}{Q} \cdot \frac{d_{ref} - d_{ToF}}{d_{ToF}}. \quad (\text{C.24})$$

we have that

$$\theta_{ToF}^{PU} = \theta_{ToF} + 2\pi k_{ref}. \quad (\text{C.25})$$

Since the only differences between θ_{ToF}^{PU} and θ_{target}^{PU} are the fluctuations due to noise, by assuming that the noise is smaller than half of the phase wrapping distance, we obtain that

$$|\theta_{ToF}^{PU} - \theta_{target}^{PU}| < \pi. \quad (\text{C.26})$$

By using together Eq. C.22 and C.23, we have that

$$d_{SL} = d_{ref} \left(1 + \frac{d_{ref} - d_{ToF}}{d_{ToF}} + \frac{Q}{b} (\theta_{ToF}^{PU} - \theta_{target}^{PU}) \right)^{-1} \quad (\text{C.27})$$

Since we have assumed that $|\theta_{ToF}^{PU} - \theta_{target}^{PU}| < \pi$, recalling that $\theta_{target}^{PU} = \theta_{target} + 2\pi k_{target}$ and $\theta_{ToF}^{PU} = \theta_{ToF} + 2\pi k_{ref}$, it comes out that

$$\begin{aligned} \theta_{ToF}^{PU} - \theta_{target}^{PU} &= (\theta_{ToF}^{PU} - \theta_{target}^{PU})_{[-\pi; \pi]} \\ &= (\theta_{ToF} + 2\pi k_{ref} - \theta_{target} - 2\pi k_{target})_{[-\pi; \pi]} \\ &= (\theta_{ToF} - \theta_{target})_{[-\pi; \pi]} \end{aligned} \quad (\text{C.28})$$

From Eq. C.27 and C.28 it turns out that:

$$d_{SL} = d_{ref} \left(1 + \frac{d_{ref} - d_{ToF}}{d_{ToF}} + \frac{Q}{b} (\theta_{ToF} - \theta_{target})_{[-\pi; \pi]} \right)^{-1} \quad (\text{C.29})$$

that is Eq. 4.8 presented in Section 4.2.3 and it doesn't require any explicit phase unwrapping operations.

C.3 Error Estimation of the Structured Light Approach

For the estimation of the error in the *structured light* (SL) approach we consider the model of Eq. 4.8 and we assume that θ_{ref} is noiseless since it has to be captured

APPENDIX C. STM-TOF CORRELATION FUNCTION EVALUATION AND ERROR PROPAGATION ANALYSIS

only once and multiple acquisitions can be repeated in order to remove the noise. The only remaining source of randomness is θ_{target} and by error propagation we can obtain:

$$\begin{aligned}\sigma_{d_{SL}}^2 &= \left(\frac{\partial d_{SL}}{\partial \theta_{target}} \right)^2 \sigma_{\theta_{target}}^2 \\ &= \left(Q \frac{d_{target}^2}{d_{ref} b} \right)^2 \sigma_{\theta_{target}}^2\end{aligned}\tag{C.30}$$

from Eq. C.30 it is possible to notice that the depth estimation accuracy improves if we increase the baseline between the sensor and the projector and it degrades with the increase of d_{target} , that is depth that we are going to estimate. This is a common behavior for SL systems. The reference distance d_{ref} has no effect in the accuracy since Q is directly proportional to d_{ref} itself. The phase estimation variance $\sigma_{\theta_{target}}^2$ can be retrieved from the second line of Eq. C.19. It comes out that the mean error variance for the SL depth estimation is

$$\bar{\sigma}_{d_{SL}}^2 = \left(Q \frac{d_{target}^2}{d_{ref} b} \right)^2 \frac{8B}{9\pi^2 A^2}\tag{C.31}$$

Bibliography

- [1] G. Agresti and P. Zanuttigh, “Combination of spatially-modulated tof and structured light for mpi-free depth estimation,” in *Proceedings of European Conference on Computer Vision Workshops (ECCVW)*, 2018.
- [2] G. Agresti and P. Zanuttigh, “Deep learning for multi-path error removal in tof sensors,” in *Proceedings of European Conference on Computer Vision Workshops (ECCVW)*, 2018.
- [3] G. Agresti, H. Schaefer, P. Sartor, and P. Zanuttigh, “Unsupervised domain adaptation for tof data denoising with adversarial learning,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5584–5593, 2019.
- [4] M. Biassetton, U. Michieli, G. Agresti, and P. Zanuttigh, “Unsupervised domain adaptation for semantic segmentation of urban scenes,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2019.
- [5] G. Agresti, L. Minto, G. Marin, and P. Zanuttigh, “Deep learning for confidence information in stereo and tof data fusion,” in *Proceedings of International Conference on Computer Vision Workshops (ICCVW)*, Oct 2017.
- [6] G. Agresti, L. Minto, G. Marin, and P. Zanuttigh, “Stereo and tof data fusion by learning from synthetic data,” *Information Fusion*, vol. 49, pp. 161–173, 2019.
- [7] B. Curless, “Overview of active vision techniques,” in *Proceedings of ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, vol. 99, 2000.
- [8] Stereolabs, “Zed stereo system website.” <https://www.stereolabs.com>, Accessed July 29th, 2019.
- [9] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, 2000.
- [10] P. Zanuttigh, G. Marin, C. Dal Mutto, F. Dominio, L. Minto, and G. M. Cortelazzo, *Time-of-Flight and Structured Light Depth Cameras*. Springer, 2016.
- [11] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [12] A. Fusiello, E. Trucco, and A. Verri, “A compact algorithm for rectification of stereo pairs,” *Machine Vision and Applications*, vol. 12, no. 1, pp. 16–22, 2000.
- [13] H. Hirschmuller and D. Scharstein, “Evaluation of stereo matching costs on images with radiometric differences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1582–1599, 2008.

BIBLIOGRAPHY

- [14] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [15] H. Hirschmüller, “Accurate and efficient stereo processing by semi-global matching and mutual information,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 807–814, IEEE, 2005.
- [16] W. Luo, A. G. Schwing, and R. Urtasun, “Efficient deep learning for stereo matching,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [17] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, “End-to-end learning of geometry and context for deep stereo regression,” in *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 66–75, 2017.
- [18] M. Poggi, D. Pallotti, F. Tosi, and S. Mattocchia, “Guided stereo matching,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 979–988, 2019.
- [19] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, “High-resolution stereo datasets with subpixel-accurate ground truth,” in *Proceedings of German conference on pattern recognition*, pp. 31–42, Springer, 2014.
- [20] Intel, “Intel realsense ds435 active stereo system website.” <https://www.intelrealsense.com/depth-camera-d435>, Accessed July 31st, 2019.
- [21] G. C. Birch, A. L. Dagel, B. A. Kast, and C. S. Smith, “3d imaging with structured illumination for advanced security applications,” tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2015.
- [22] J. Geng, “Structured-light 3d surface imaging: a tutorial,” *Advances in Optics and Photonics*, vol. 3, no. 2, pp. 128–160, 2011.
- [23] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, “A state of the art in structured light patterns for surface profilometry,” *Pattern recognition*, vol. 43, no. 8, pp. 2666–2680, 2010.
- [24] F. Remondino and D. Stoppa, *TOF range-imaging cameras*, vol. 68121. Springer, 2013.
- [25] Microsoft, “Kinect azure.” <https://azure.microsoft.com/it-it/services/kinect-dk/>, Accessed August 16th, 2019.
- [26] Sony Corporation, “Sony depthsensing website.” <https://www.sony-depthsensing.com>, Accessed August 14th, 2019.
- [27] M. Lindner, I. Schiller, A. Kolb, and R. Koch, “Time-of-flight sensor calibration for accurate range sensing,” *Computer Vision and Image Understanding*, vol. 114, no. 12, pp. 1318–1328, 2010.
- [28] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, “Probabilistic tof and stereo data fusion based on mixed pixels measurement models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2260–2272, 2015.
- [29] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [30] J. Marco, Q. Hernandez, A. Muñoz, Y. Dong, A. Jarabo, M. H. Kim, X. Tong, and D. Gutierrez, “Deeptof: off-the-shelf real-time correction of multipath interference in time-of-flight imaging,” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, p. 219, 2017.

-
- [31] Q. Guo, I. Frosio, O. Gallo, T. Zickler, and J. Kautz, “Tackling 3d tof artifacts through learning and the flat dataset,” in *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.
- [32] S. Su, F. Heide, G. Wetzstein, and W. Heidrich, “Deep end-to-end time-of-flight imaging,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6383–6392, 2018.
- [33] L. Zhang, B. Curless, and S. M. Seitz, “Spacetime stereo: Shape recovery for dynamic scenes,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. II–367, IEEE, 2003.
- [34] D. Freedman, Y. Smolin, E. Krupka, I. Leichter, and M. Schmidt, “Sra: Fast removal of general multipath for tof sensors,” in *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 234–249, Springer, 2014.
- [35] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330–1334, 1998.
- [36] M. Gupta and S. K. Nayar, “Micro phase shifting,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 813–820, IEEE, 2012.
- [37] “CloudCompare 3d point cloud and mesh processing software open source project.” <https://www.danielgm.net/cc/>. Accessed August 24th, 2019.
- [38] The Blender Foundation, “Blender website.” <https://www.blender.org/>, Accessed July 29th, 2019.
- [39] Blender Swap, “Blend swap website.” <https://www.lendwsap.com/>, Accessed August 22nd, 2019.
- [40] S. Meister, R. Nair, and D. Kondermann, “Simulation of Time-of-Flight Sensors using Global Illumination,” in *Vision, Modeling and Visualization* (M. Bronstein, J. Favre, and K. Hornmann, eds.), The Eurographics Association, 2013.
- [41] The LuxRender Project, “Luxrender website.” <https://luxcorerender.org/>, Accessed July 29th, 2019.
- [42] J. Sell and P. O’Connor, “The xbox one system on a chip and kinect sensor,” *IEEE Micro*, vol. 34, no. 2, pp. 44–53, 2014.
- [43] T. Edeler, K. Ohliger, S. Hussmann, and A. Mertins, “Time-of-flight depth image denoising using prior noise information,” in *Proceedings of IEEE International Conference on Signal Processing*, pp. 119–122, IEEE, 2010.
- [44] F. Lenzen, H. Schäfer, and C. Garbe, “Denoising time-of-flight data with adaptive total variation,” in *Proceedings of the International Symposium on Visual Computing*, pp. 337–346, Springer, 2011.
- [45] M. Georgiev, R. Bregović, and A. Gotchev, “Time-of-flight range measurement in low-sensing environment: Noise analysis and complex-domain non-local denoising,” *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2911–2926, 2018.
- [46] R. Whyte, L. Streeter, M. J. Cree, and A. A. Dorrington, “Review of methods for resolving multi-path interference in time-of-flight range cameras,” in *IEEE Sensors*, pp. 629–632, IEEE, 2014.

BIBLIOGRAPHY

- [47] S. Fuchs, “Multipath interference compensation in time-of-flight camera images,” in *Proceedings of International Conference on Pattern Recognition (ICPR)*, pp. 3583–3586, IEEE, 2010.
- [48] S. Fuchs, M. Suppa, and O. Hellwich, “Compensation for multipath in tof camera measurements supported by photometric calibration and environment integration,” in *Proceedings of International Conference on Computer Vision Systems*, pp. 31–41, Springer, 2013.
- [49] D. Jiménez, D. Pizarro, M. Mazo, and S. Palazuelos, “Modeling and correction of multipath interference in time of flight cameras,” *Image and Vision Computing*, vol. 32, no. 1, pp. 1–13, 2014.
- [50] A. Kadambi, R. Whyte, A. Bhandari, L. Streeter, C. Barsi, A. Dorrington, and R. Raskar, “Coded time of flight cameras: sparse deconvolution to address multipath interference and recover time profiles,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 167, 2013.
- [51] R. Whyte, L. Streeter, M. J. Cree, and A. A. Dorrington, “Resolving multiple propagation paths in time of flight range cameras using direct and global separation methods,” *Optical Engineering*, vol. 54, no. 11, p. 113109, 2015.
- [52] N. Naik, A. Kadambi, C. Rhemann, S. Izadi, R. Raskar, and S. 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*, pp. 73–81, 2015.
- [53] S. Achar, J. R. Bartels, W. L. Whittaker, K. N. Kutulakos, and S. G. Narasimhan, “Epipolar time-of-flight imaging,” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 37, 2017.
- [54] A. Bhandari, A. Kadambi, R. Whyte, C. Barsi, M. Feigin, A. Dorrington, and R. Raskar, “Resolving multipath interference in time-of-flight imaging via modulation frequency diversity and sparse regularization,” *Optics letters*, vol. 39, no. 6, pp. 1705–1708, 2014.
- [55] K. Son, M.-Y. Liu, and Y. 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)*, pp. 3390–3397, 2016.
- [56] A. A. Dorrington and R. Z. Whyte, “Time of flight camera system which resolves direct and multi-path radiation components,” Jan. 23 2018. US Patent 9,874,638.
- [57] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar, “Fast separation of direct and global components of a scene using high frequency illumination,” *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 935–944, 2006.
- [58] Y. Xu, L. Ekstrand, J. Dai, and S. Zhang, “Phase error compensation for three-dimensional shape measurement with projector defocusing,” *Applied Optics*, vol. 50, no. 17, pp. 2572–2581, 2011.
- [59] C. Dal Mutto, P. Zanuttigh, and G. Cortelazzo, “A probabilistic approach to tof and stereo data fusion,” in *3DPVT*, (Paris, France), May 2010.
- [60] J. Zhu, L. Wang, J. Gao, and R. Yang, “Spatial-temporal fusion for high accuracy depth maps using dynamic mrfs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 899–909, 2010.
- [61] C. D. Mutto, P. Zanuttigh, and G. M. Cortelazzo, “Probabilistic tof and stereo data fusion based on mixed pixels measurement models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2260–2272, 2015.

-
- [62] M. O’Toole, F. Heide, L. Xiao, M. B. Hullin, W. Heidrich, and K. N. Kutulakos, “Temporal frequency probing for 5d transient analysis of global light transport,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 87, 2014.
- [63] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [64] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [65] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 839–846, IEEE, 1998.
- [66] D. Chan, H. Buisman, C. Theobalt, and S. Thrun, “A noise-aware filter for real-time depth upsampling,” in *Proceedings of Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.
- [67] G. Marin, P. Zanuttigh, and S. Mattocchia, “Reliable fusion of tof and stereo depth driven by confidence measures,” in *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 386–401, Springer, 2016.
- [68] M. Zhang and B. K. Gunturk, “Multiresolution bilateral filtering for image denoising,” *IEEE Transactions on Image Processing*, vol. 17, no. 12, pp. 2324–2333, 2008.
- [69] M. Gupta, S. K. Nayar, M. B. Hullin, and J. Martin, “Phasor imaging: A generalization of correlation-based time-of-flight imaging,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 5, p. 156, 2015.
- [70] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss functions for image restoration with neural networks,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.
- [71] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.
- [72] M. Long, Y. Cao, J. Wang, and M. I. Jordan, “Learning transferable features with deep adaptation networks,” in *Proceedings of International Conference on Machine Learning (ICML)*, pp. 97–105, JMLR. org, 2015.
- [73] M. Long, H. Zhu, J. Wang, and M. I. Jordan, “Deep transfer learning with joint adaptation networks,” in *Proceedings of International Conference on Machine Learning (ICML)* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 2208–2217, PMLR, 06–11 Aug 2017.
- [74] P. Morerio, J. Cavazza, and V. Murino, “Minimal-entropy correlation alignment for unsupervised deep domain adaptation,” *arXiv preprint arXiv:1711.10288*, 2017.
- [75] B. Sun and K. Saenko, “Deep coral: Correlation alignment for deep domain adaptation,” in *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 443–450, Springer, 2016.
- [76] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, “Revisiting batch normalization for practical domain adaptation,” *arXiv preprint arXiv:1603.04779*, 2016.
- [77] F. M. Cariucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, “Autodial: Automatic domain alignment layers,” in *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 5077–5085, IEEE, 2017.

BIBLIOGRAPHY

- [78] S. Roy, A. Siarohin, E. Sangineto, S. R. Bulo, N. Sebe, and E. Ricci, “Unsupervised domain adaptation using feature-whitening and consensus loss,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9471–9480, 2019.
- [79] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *Proceedings of International Conference on Machine Learning (ICML)*, pp. 1180–1189, 2015.
- [80] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [81] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proceedings of the Neural Information Processing Systems Conference (NIPS)*, pp. 2672–2680, 2014.
- [82] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, “Unsupervised pixel-level domain adaptation with generative adversarial networks,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, p. 7, 2017.
- [83] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, p. 5, 2017.
- [84] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, “Generate to adapt: Aligning domains using generative adversarial networks,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8503–8512, 2018.
- [85] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, “A survey on deep learning techniques for image and video semantic segmentation,” *Applied Soft Computing*, vol. 70, pp. 41 – 65, 2018.
- [86] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The Cityscapes dataset for semantic urban scene understanding,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, 2016.
- [87] G. Brostow, J. Fauqueur, and R. Cipolla, “Semantic object classes in video: A high-definition ground truth database,” *Pattern Recognition Letters*, pp. 88–97, 2009.
- [88] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, “The Mapillary vistas dataset for semantic understanding of street scenes,” in *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 4990–4999, 2017.
- [89] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for data: Ground truth from computer games,” in *Proceedings of European Conference on Computer Vision (ECCV)* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), vol. 9906, pp. 102–118, Springer International Publishing, 2016.
- [90] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3234–3243, 2016.
- [91] J. Hoffman, D. Wang, F. Yu, and T. Darrell, “FCNs in the wild: Pixel-level adversarial and constraint-based adaptation,” *arXiv preprint arXiv:1612.02649*, 2016.

-
- [92] S. Sankaranarayanan, Y. Balaji, A. Jain, S. Nam Lim, and R. Chellappa, “Learning from synthetic data: Addressing domain shift for semantic segmentation,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3752–3761, 2018.
- [93] Y. Zhang, P. David, and B. Gong, “Curriculum domain adaptation for semantic segmentation of urban scenes,” in *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 2020–2030, 2017.
- [94] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, “Cycada: Cycle-consistent adversarial domain adaptation,” in *Proceedings of International Conference on Machine Learning (ICML)*, 2018.
- [95] Y. C. Chen, Y. Y. Lin, M. H. Yang, and J. B. Huang, “Crdoco: Pixel-level domain transfer with cross-domain consistency,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1791–1800, 2019.
- [96] Y.-H. Tsai, W.-C. Hung, S. Schuler, K. Sohn, M.-H. Yang, and M. Chandraker, “Learning to adapt structured output space for semantic segmentation,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7472–7481, 2018.
- [97] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, “Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [98] Y. Chen, W. Li, and L. Van Gool, “Road: Reality oriented adaptation for semantic segmentation of urban scenes,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7892–7901, 2018.
- [99] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, “Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2517–2526, 2019.
- [100] P. Z. Ramirez, A. Tonioni, S. Salti, and L. Di Stefano, “Learning across tasks and domains,” *arXiv preprint arXiv:1904.04744*, 2019.
- [101] Z. Ren and Y. J. 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.
- [102] A. Tonioni, M. Poggi, S. Mattoccia, and L. Di Stefano, “Unsupervised adaptation for deep stereo,” in *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 1605–1613, 2017.
- [103] A.I. Wiki, “A beginner’s guide to generative adversarial networks (gans.)” <https://skymind.ai/wiki/generative-adversarial-network-gan>, Accessed November 24th, 2019.
- [104] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, “Least squares generative adversarial networks,” in *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 2813–2821, IEEE, 2017.
- [105] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *arXiv preprint*, 2017.

BIBLIOGRAPHY

- [106] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, “Tensorflow: a system for large-scale machine learning,” in *Symposium on Operating Systems Design and Implementation (OSDI)*, vol. 16, pp. 265–283, 2016.
- [107] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015.
- [108] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in *Proceedings of International Conference on Learning Representation (ICLR)*, 2016.
- [109] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2881–2890, 2017.
- [110] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, 2018.
- [111] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou³⁴, Y.-Y. Lin, and M.-H. Yang¹⁵, “Adversarial learning for semi-supervised semantic segmentation,” in *Proceedings of British Machine Vision Conference (BMVC)*, 2018.
- [112] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 740–755, Springer, 2014.
- [113] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, “High-resolution stereo datasets with subpixel-accurate ground truth,” in *Proceedings of German Conference on Pattern Recognition*, pp. 31–42, Springer, 2014.
- [114] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [115] B. Tippetts, D. Lee, K. Lillywhite, and J. Archibald, “Review of stereo vision algorithms and their suitability for resource-limited systems,” *Journal of Real-Time Image Processing*, pp. 1–21, 2013.
- [116] X. Hu and P. Mordohai, “A quantitative evaluation of confidence measures for stereo vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2121–2133, 2012.
- [117] M. Poggi, F. Tosi, and S. Mattoccia, “Quantitative evaluation of confidence measures in a machine learning world,” in *ICCV*, 2017.
- [118] M. Poggi and S. Mattoccia, “Learning from scratch a confidence measure,” in *Proceedings of British Machine Vision Conference (BMVC)*, 2016.
- [119] A. Seki and M. Pollefeys, “Patch based confidence prediction for dense disparity map,” in *Proceedings of British Machine Vision Conference (BMVC)*, 2016.
- [120] M. Poggi and S. Mattoccia, “Learning to predict stereo reliability enforcing local consistency of confidence maps,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

-
- [121] M. Hansard, S. Lee, O. Choi, and R. Horaud, *Time-of-Flight Cameras: Principles, Methods and Applications*. Springer Briefs in Computer Science, Springer, 2013.
- [122] F. Remondino and D. Stoppa, eds., *TOF Range-Imaging Cameras*. Springer, 2013.
- [123] S. A. Gudmundsson, H. Aanaes, and R. Larsen, “Fusion of stereo vision and time of flight imaging for improved 3d estimation,” *Int. J. Intell. Syst. Technol. Appl.*, vol. 5, pp. 425–433, 2008.
- [124] R. Nair, K. Ruhl, F. Lenzen, S. Meister, H. Schäfer, C. Garbe, M. Eisemann, M. Magnor, and D. Kondermann, “A survey on time-of-flight stereo fusion,” in *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications* (M. Grzegorzec, C. Theobalt, R. Koch, and A. Kolb, eds.), vol. 8200 of *Lecture Notes in Computer Science*, pp. 105–127, Springer Berlin Heidelberg, 2013.
- [125] J. Zhu, L. Wang, R. Yang, and J. Davis, “Fusion of time-of-flight depth and stereo for high accuracy depth maps,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [126] J. Zhu, L. Wang, J. Gao, and R. Yang, “Spatial-temporal fusion for high accuracy depth maps using dynamic mrfs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 899–909, 2010.
- [127] J. Zhu, L. Wang, R. Yang, J. E. Davis, and Z. Pan, “Reliability fusion of time-of-flight depth and stereo geometry for high quality depth maps,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1400–1414, 2011.
- [128] R. Nair, F. Lenzen, S. Meister, H. Schaefer, C. Garbe, and D. Kondermann, “High accuracy tof and stereo sensor fusion at interactive rates,” in *Proceedings of European Conference on Computer Vision Workshops (ECCVW)*, 2012.
- [129] B. Chen, C. Jung, and Z. Zhang, “Variational fusion of time-of-flight and stereo data using edge selective joint filtering,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2017.
- [130] B. Chen, C. Jung, and Z. Zhang, “Variational fusion of time-of-flight and stereo data for depth estimation using edge selective joint filtering,” *IEEE Transactions on Multimedia*, pp. 1–1, 2018.
- [131] G. Evangelidis, M. Hansard, and R. Horaud, “Fusion of Range and Stereo Data for High-Resolution Scene-Modeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2178 – 2192, 2015.
- [132] S. Mattoccia, “A locally global approach to stereo correspondence,” in *Proceedings of International Conference on Computer Vision Workshops (ICCVW)*, pp. 1763–1770, IEEE, 2009.
- [133] C. Dal Mutto, P. Zanuttigh, S. Mattoccia, and G. Cortelazzo, “Locally consistent tof and stereo data fusion,” in *Proceedings of European Conference on Computer Vision Workshops (ECCVW)*, pp. 598–607, Springer, 2012.
- [134] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [135] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

BIBLIOGRAPHY

- [136] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015.
- [137] M. D. Zeiler, “Adadelta: an adaptive learning rate method,” *arXiv preprint arXiv:1212.5701*, 2012.
- [138] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.
- [139] Q. Yang, R. Yang, J. Davis, and D. Nister, “Spatial-depth super resolution for range images,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.
- [140] H. H. Ku *et al.*, “Notes on the use of propagation of error formulas,” *Journal of Research of the National Bureau of Standards*, vol. 70, no. 4, 1966.
- [141] R. Lange, P. Seitz, A. Biber, and S. C. Lauxtermann, “Demodulation pixels in ccd and cmos technologies for time-of-flight ranging,” in *Sensors and camera systems for scientific, industrial, and digital photography applications*, vol. 3965, pp. 177–189, International Society for Optics and Photonics, 2000.

List of my Publications

Journal Papers

- G. Agresti, L. Minto, G. Marin, and P. Zanuttigh, “Stereo and ToF Data Fusion by Learning from Synthetic Data,” in Elsevier Information Fusion, vol. 49, pp. 161–173, 2019.
- G. Marin, G. Agresti, L. Minto, and P. Zanuttigh, “A Multi-Camera Dataset for Depth Estimation in an Indoor Scenario,” in Elsevier Data in Brief (accepted).
- M. Poggi, G. Agresti, F. Tosi, P. Zanuttigh, and S. Mattoccia, “Confidence Estimation for ToF and Stereo Sensors and its Application to Depth Data Fusion,” in IEEE Sensors Journal (accepted).
- U. Michieli, M. Biassetton, G. Agresti, and P. Zanuttigh, “Adversarial Learning and Self-Teaching Techniques for Domain Adaptation in Semantic Segmentation,” in IEEE Transactions on Intelligent Vehicles (submitted).

Conference Papers

- G. Agresti, L. Minto, G. Marin, and P. Zanuttigh, “Deep Learning for Confidence Information in Stereo and ToF Data Fusion,” in Proceedings of International Conference on Computer Vision Workshops (ICCVW), 2018.
- G. Agresti and P. Zanuttigh, “Combination of Spatially-Modulated ToF and Structured Light for MPI-Free Depth Estimation,” in Proceedings of European Conference on Computer Vision Workshops (ECCVW), 2018.
- G. Agresti and P. Zanuttigh, “Deep Learning for Multi-Path Error Removal in ToF Sensors,” in Proceedings of European Conference on Computer Vision Workshops (ECCVW), 2018.
- G. Agresti and S. Milani, “Material Identification Using RF Sensors and Convolutional Neural Networks,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.
- M. Biassetton, U. Michieli, G. Agresti and P. Zanuttigh, “Unsupervised Domain Adaptation for Semantic Segmentation of Urban Scenes,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.
- G. Agresti, H. Schaefer, P. Sartor, and P. Zanuttigh, “Unsupervised Domain Adaptation for ToF Data Denoising with Adversarial Learning,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

Acknowledgments

Firstly, I want to thank my advisor prof. Pietro Zanuttigh for supporting and guiding me in my Ph.D study. I thank Sony Europe for funding my Ph.D. and all the people at Sony Eutec who helped me in my research and made me feel part of the company during my stay in Stuttgart, in particular Oliver, Martina, Henrik, Markus, Yalcin, Vincent, Bi, Matthias, Piergiorgio and Francesco.

I thank Chiara, Daniel, Davide, Federico, Francesco, Giulia, Leonardo, Michele, Paolo, Sebastiano, Silvia, Umberto and the other students and post-docs at DEI for all the coffee breaks and the spritz after work. You made these years lighter and unforgettable.

I would like to thank my parents Maria and Angelo, my brothers Filippo and Salvatore, my sisters in law Barbara and Ornella for supporting me and for their constant presence in all these years which brought me here. Thanks to my little nephews Emanuele, Francesca, Miriam and Gabriel for letting me feel a child when I play with you. A special thank to Silvana for being always there for me and for encouraging me day by day. I would like to thank Silvana's family for making me feel at home.