

La borsa di dottorato è stata cofinanziata con risorse del
Programma Operativo Nazionale Ricerca e Innovazione 2014-2020 (CCI 2014IT16M2OP005),
Fondo Sociale Europeo, Azione I.1 "Dottorati Innovativi con caratterizzazione Industriale"



UNIONE EUROPEA
Fondo Sociale Europeo



Unsupervised deep learning encoders/decoders for laparoscopic 3D depth estimation and surface reconstruction.



UNIVERSITÀ DEGLI STUDI
DI SALERNO

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Supervisor

Ch.mo Prof. Roberto Tagliaferri

Candidate

dott. Francesco Bardozzo

School of Big Data Management

Director

Ch.mo Prof. Valerio Antonelli

February, 2021

DECLARATION

I here by declare that the thesis entitled “Unsupervised deep learning encoder-s/decoders for laparoscopic 3D depth estimation and surface reconstruction.” submitted by me, for the award of the degree of *Doctor of Philosophy* at the University of Salerno is a record of bonafide work carried out by me under the supervision of Prof. Roberto Tagliaferri, University of Salerno (IT).

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Salerno, Italy

Date: 20/02/2021

Signature

(dott. Francesco Bardozzo)

CERTIFICATE

This is to certify that the thesis entitled “Unsupervised deep learning encoder-decoders for laparoscopic 3D depth estimation and surface reconstruction.” submitted by dott Francesco Bardozzo, University of Salerno (IT) for the award of the degree of *Doctor of Philosophy*, is a record of bonafide work carried out by him under my supervision, as per the University of Salerno code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place: Salerno, Italy

Date: 21/02/2021

Signature

(Prof. Roberto Tagliaferri)

ABSTRACT

Accurate and real-time methodologies for a non-invasive three-dimensional representation and reconstruction of internal patient surfaces is one of the main research fields in computer-assisted surgery and endoscopy. Mono and stereo endoscopic images of soft tissues and observable organs are converted into three-dimensional representation by the estimation of depth maps. In this Thesis, I developed and tested self-supervised stacked and Siamese encoder/decoder neural networks suited to work during a minimally invasive surgery, while respecting the constraint of real-time and in mono and/or stereo configurations. Computed depth-maps from monocular and stereo endoscopic images are tested with three different test sets known in literature and with a new challenging test set generated through simulated environments. I introduced also a fuzzy adaptive algorithm to fine tune depth estimations perturbed by specular reflection and other artifacts. The models I present outperform mono and stereo depth estimation obtained by state-of-the-art models on similar researches. Extensive robustness and sensitivity analyses on more than 30 000 frames are performed. This Research Thesis leads to important results and improvements in monocular and stereo real-time depth estimations of soft tissues by showing, also, their efficacy and applicability in real surgical workd-flows.

Keywords: *Unsupervised depth estimation, Unsupervised 3D soft-tissue and surface reconstruction, Fuzzy adaptive depth post processing, 3D laparosopocy, Computer-aided medicine.*

ACKNOWLEDGEMENT

With immense pleasure and a deep sense of respect and gratitude, I wish to express my sincere thanks to my supervisor Prof. Roberto Tagliaferri (University of Salerno - Italy) without his teachings, his motivation and his continuous encouragement, this research would not have been successfully completed. I thank Dr. Toby Collins and Dr. Alexandre Hostetter (IRCAD - Strasbourg - France) for their rigorous and very important advice on setting up my research. I also thank Prof. Pietro Lió (University of Cambridge - United Kingdom), Prof. Humberto Bustince (University of Navarra - Spain) from whom I was able to learn several aspects of applied scientific methodologies in disparate research fields and with whom I published important researches during my PhD period. I would like to thank the SCAI CINECA group for their confidence in my project and for providing me with access to the MARCONI100 supercomputer.

I am grateful to Doctor Antonello Forgione, Ing. Renzo Zaltieri (MILANO Value Biotech) and Prof. Luigi Troiano (University of Salerno) for their help in providing me with essential elements for studying the application domain and the key aspects in which to focus my research.

I express my sincere thanks to Prof. Valerio Antonelli, Coordinator of my PhD Course, for his kind words of support and encouragement.

I like to acknowledge the support rendered by my colleagues in several ways throughout my research work.

I wish to extend my profound sense of gratitude to my parents (Prof. Maria Consiglia Montanino, Ing. Luigi Giacinto Bardozzo, and Prof. Lavinia Angela Bardozzo) for all the sacrifices they made during my research and also providing me with moral support and encouragement.

Last but not the least, I would like to thank my partner Dolores Durante for her constant encouragement and moral support along with patience, understanding and love.

Domicella (Avellino) Italy

02/02/2021

Francesco Bardozzo

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGEMENT	ii
LIST OF FIGURES	vi
LIST OF TABLES	xi
LIST OF TERMS AND ABBREVIATIONS	xiii
1 Introduction	2
1.1 Application Domain	3
1.2 Research challenges	5
1.3 Research objectives	7
1.4 Research advances and Thesis organization	8
2 Related work and some insights	9
2.1 Dense depth estimation models	9
2.1.1 Depth-map reconstruction with monocular images	10
2.1.2 Depth-map reconstruction with binocular images	11
2.2 Pre- and post-processing models	12
2.3 Model proposal positioning	13
3 Datasets	15
3.1 Training set	15
3.2 Test sets	16
3.2.1 Test set without ground truth	16
3.2.2 In Silico and In Vivo test sets with ground truth	16
3.3 Simulated test set	20
3.3.1 Initial setup	20
3.3.2 Video tipologies and artifacts simulations	22

3.3.3	Implementation details and configurations	23
4	Unsupervised models for Mono and Stereo Depth Estimation	25
4.1	<i>Mono depth CNN</i> architecture	26
4.2	<i>Stereo depth CNN</i> architecture and <i>Stereo depth pipeline</i> configurations .	28
4.3	Unsupervised loss functions	28
4.3.1	<i>Mono depth CNN</i> loss	28
4.3.2	<i>Stereo depth CNN</i> loss	29
4.4	Training configurations	30
4.5	<i>Mono depth CNN</i> and <i>Stereo depth CNN</i> implementation details	31
4.6	Training time performances	32
4.7	Prediction time performances	33
5	Depth map corrections with fuzzy post processing.	34
5.1	Computation of the fuzzy integral image CF_{12}	35
5.1.1	Computation of non-fuzzy integral image S with SAT	35
5.1.2	Algorithm 1: Fuzzy integral image computation with Fuzzy-SAT .	36
5.2	Adaptive binarization based on fuzzy integral images	39
5.3	GPU optimization of CF_{12} -algorithm (Algorithm 1 and Algorithm 2) . . .	42
5.4	Fuzzy post-processing of depth map estimations	42
6	Results and Applications	44
6.1	Experimental setup	44
6.1.1	Evaluations on datasets with ground truth	45
6.1.2	Results reported previously	45
6.2	Evaluation without Ground Truth	46
6.3	Evaluation with Ground Truths	47
6.3.1	Comparisons on Test Set B	47
6.3.2	Comparisons on Test Set C	47
6.4	Evaluations on the simulated Test Set D	51
6.4.1	Experiment 1: Change-ligth-fix-position vs fix-position-change-light	51
6.4.2	Experiment 2: Scene ensemble vs Scene ensemble with smoke . .	51
6.4.3	Experiment 3: Normal condition vs three levels of smoke	52

6.4.4	Additional samples and comparisons also with fuzzy pre processing.	55
6.5	Visual results	55
7	Conclusions	59
	REFERENCES	60
	LIST OF PUBLICATIONS	72

Appendices

Appendix A	Insights on Chapter 4	75
A.1	Additional collection of visual and numerical results on Test Set D.	75
A.2	Visual results: additional information.	75
A.3	Learning of the stereo displacement function	75
Appendix B	Insights on Chapter 5	77
B.1	Fuzzy Theory	77
	B.1.1 Fuzzy measure	77
	B.1.2 Uniform fuzzy measure	77
	B.1.3 Aggregation function	77
	B.1.4 Sugeno integral definition	78
	B.1.5 Generalized Sugeno integral	78
B.2	GPU-oriented implementation in python/numba of CF_{12} -optimized-algorithm	79

LIST OF FIGURES

3.1	In this Figure left-right channel of <i>In Vivo</i> samples for unlabeled training and test set are provided. These scenes are publicly accessible on the Hamlyn Center Vision repository and in particular are provided by Ye et al. (2017)	17
3.2	In this Figure left-right channel samples with <i>In Silico</i> ground truth for different scenes are provided. These two scenes are publicly accessible on the Hamlyn Center Vision repository and in particular are provided by Pratt et al. (2010)	18
3.3	In this Figure three left-right channel <i>In Vivo</i> samples with structured-light-based ground truth for different scenes are provided. This dataset is under gentle concession of <i>MICCAI Endovis Grand Challenge 2019</i>	19
21figure.caption.4		
3.5	In this Figure three left channel samples with ground truth for LIGHT challenge scenes (at three different endoscopic light positions) of <i>Test Set D</i> are provided.	24
4.1	In Figure 4.1 show the <i>Mono depth CNN</i> architecture. The input layer ($\mathbf{I_X}$), where $\mathbf{X}$ could be the left or right channel image is shown in gray. The <i>Convolutional</i> layers $\mathbf{C_n}$ are shown in violet for the (a) encoder, in blue for the (b) boottleneck and in red for the (c) decoder. The <i>Max Pooling</i> layers ($\mathbf{M_1}$, $\mathbf{M_2}$, $\mathbf{M_3}$) are shown in yellow. The <i>upsampling</i> layers ($\mathbf{U_1}$, $\mathbf{U_2}$, $\mathbf{U_3}$) are shown in orange. The variable $\mathbf{F}$ corresponds to the number of filter and is equal to 32. The two two skip connections $\mathbf{C_1} \oplus \mathbf{C_9}$ and $\mathbf{C_2} \oplus \mathbf{C_8}$ are shown with dotted arrows.	26

- 4.2 *Stereo depth CNN* is designed with two encoder/decoder networks in a siamese configuration (**SN1**) which are connected with a stereo matching module (**SM**). The encoder/decoder network has the shape of the *Mono depth CNN* described in Figure 4.1. Instead, the **SM** is an additional architectural module composed of fully connected layers and a convolutional layer C_{12} at the end (in orange). The *Stereo depth CNN* takes in input two depth maps (D_l) and (D_r) for left and right channel respectively. The left branch C_{10}^l and right branch C_{10}^r concatenation $\otimes$ is shown in red. The fully connected layers (FC_1, FC_2, FC_3) are indicated in violet. The output depthmap is indicated in gray as D_s 27
- 4.3 In Figure 4.3 *Stereo depth pipeline* is shown. In particular, two *Mono depth CNNs* are both disposed in a siamese configuration (**SN1**) and stacked on the *Stereo depth CNN* (**SN2 + SM** - see also Figure 4.2). In this way the *Stereo depth pipeline* takes the form of **SN1 + (SN2 + SM)**. At the prediction time the *Stereo depth pipeline* takes in input (I_l) and (I_r) for left and right channel providing in output (D_s). 27
- 5.1 Overview of the complete pipeline from the original depth map to the binarized depth map by going through the Algorithm 1 (Box **(a)**) and then Algorithm 2 (Box **(b)**). In Figure 5.1 Box **(b)**, the k -th local search window w_n used for the locally adaptive thresholding is shown. It is important to underline that, as it is described in Algorithm 2, only the 4 values in the orange rectangles are used for binarization. These 4 values are not necessarily adjacent like in the j -th operative window of Figure 5.2. The dashed red arrows show the local window sliding directions, from up to down, from left to right. The local window has a fixed size of $n_a * n_a$. The binarized depth map in output will be used for original depth map post processing as it is described in Section 5.4. 35

- 5.2 Overview of Algorithm 1. From the original Depth Map in input (Box **(a)**) the algorithm computes both the non-fuzzy integral image S and dynamically transforms it into a fuzzy integral image ($F_{CF_{12}}$) (see Section 5.1.1 and 5.1). The blue square in Box **(a)** represents current value of $p(x, y) \in I$ (in red) and its neighboring pixels defined in j -th operative window. For each pixel $p(x, y)$ and for every j -th operative window (yellow box), the 4 values in S are mapped with the associated fuzzy measures through $F_{CF_{12}}(x, y) = f_i(\vec{ov}, \vec{m})$. The output of fuzzy-based integral functional computation is saved in the fuzzy integral image $F_{CF_{12}}$. This computation is described in formula 5.3. In the j -th operative window, the fixed values of $\min v_1$ and $\max v_4$ are represented in violet and blue. For the decision of v_2 and v_3 , the green arrow indicates the P_{swap} action as described in Procedure 5.5. 37
- 5.3 In Figure 5.3 - Box **(a)**, the left reference image $\mathbf{I}^1$ is shown. In Box **(b)**, its ground truth is reported. The *Stereo depth pipeline* depth estimation $\mathbf{D}_S$ is shown in Box **(c)**. In Figure **(d)**, the dashed black square shows a set of features which are classified with value 0 on the binary mask $\mathbf{M}_k$. In Figure (c), the same set of features is indicated with the dashed black square. While, in $f(\mathbf{D}_S)$, according to the procedure described in Section 5.4, these features are lowered in intensity (Dashed black square of Box **(e)**). In Box **(f)**, the 3D Mesh of the window depicted in the red dashed square of Box **(a)(e)** is shown with two different rotation degrees. In detail, this Mesh is obtained from the depth estimation of Box **(e)**. While, the pixel colours of Box **(a)** are remapped into the Mesh in Box **(f)**. In this case fuzzy-post processing is able to completely remove specular reflections. 43

6.1	Visual comparisons on a sample of <i>Test set B</i> : In this Figure a sample of depth estimation for my models are in comparison with Ye et al. (2017) 's M_1 , M_2 , M_3 and M_4 . Box (a): <i>Test Set B</i> left/right input sample. Box (b): the ground truth of the input sample . Box (c): the depth estimation in output from <i>Mono depth CNN</i> . Box (d): the depth map in output from <i>Stereo depth pipeline</i> . Box (e): the depth map estimated by the <i>Stereo depth pipeline</i> is post-processed. Box (f): Mono and stereo depth estimation in output from the models of Ye et al. (2017)	48
6.2	Visual comparisons on a sample of <i>Test set C</i> : In this Figure a sample of depth estimation for my models are in comparison with Ye et al. (2017) 's M_1 , M_2 , M_3 and M_4 models. Box (a): <i>Test Set B</i> left/right input sample. Box (b): the ground truth of the input sample . Box (c): the depth estimation in output from <i>Mono depth CNN</i> . Box (d): the depth map in output from <i>Stereo depth pipeline</i> . Box (e): the depth map estimated by the <i>Stereo depth pipeline</i> is post-processed. Box (f): Mono and stereo depth estimation in output from the models of Ye et al. (2017)	50
6.3	In this Figure three different levels of noise (smoke) are added to three samples of <i>Test Set D</i> . <i>Stereo depth pipeline</i> predictions are computed with two different trained <i>Stereo depth CNNs</i> . In particular, depth predictions are provided with <i>Stereo depth CNN</i> trained until epoch 30 (its depths predictions are indicated with D_s^{30}) and another one it is trained until epoch 60 (its depth predictions are indicated with D_s^{60}). The depths are without fuzzy post processing. In Table 6.6 an analytical comparison between depth predictions D_s^{30} and D_s^{60} is provided.	54

6.4	Visual comparisons on a sample of <i>Test set D</i> with <i>SMOKE challenge</i> perturbations and evident specular reflections: In this Figure a sample of depth estimation for my models are in comparison with Ye et al. (2017) 's M_1 , M_2 , M_3 and M_4 models. Box (a): <i>Test Set B</i> left/right input sample. Box (b): the ground truth of the input sample . Box (c): the depth estimation in output from <i>Mono depth CNN</i> . Box (d): the depth map in output from <i>Stereo depth pipeline</i> . Box (e): the depth map estimated by the <i>Stereo depth pipeline</i> is post-processed. Box (f): Mono and stereo depth estimation in output from the models of Ye et al. (2017)	56
6.5	In Figure 6.5, the estimated 3D surface reconstruction of a frame from DEFORM challenge set is provided. The estimated 3D reconstruction is shown with different 45 rotation degrees from right-to-left (a) and left-to-right (b). The original left-right image and their ground truths are in box (c) and (d) respectively. The 3D reconstructed surfaces on a black background were taken directly from the tool I developed. Their brightness and shading are dependent on the 3D Mesh reprojection, so there are no further modifications to the 3D model in post-processing or color adjustments. Moreover, it is not applied fuzzy post-processing on their estimated depths. The pitted holes on the borders and on 3D models are the missing or incorrect triple of points that the tool was unable to reconstruct from the depth map. A finer 3D reconstruction could address these holes by paying for additional computation time.	58
A.1	In Figure A.1 - Box (a), the stereo images (left and right channel) are shown for the blue point displaced. In Figure A.1 - Box (b), the intuitive process of learning the displacement function is shown. The displacement function f_δ is shown with the black arrow across the learning epochs.	76

LIST OF TABLES

4.1	. In Table 4.1, <i>Mono depth CNN</i> parameters, connections and layer shapes are shown.	32
4.2	. In Table 4.2, <i>Stereo depth CNN</i> parameters, connections and layer shapes are shown.	33
5.1	I provide for each test set with ground truth described in Chapter 3 - Section 3.2.2, the CF_{12} -hyper-parameters table used for <i>Stereo depth CNN</i> post processing ($f(\mathbf{D}_S)$). The fuzzy kernel size is determined by Equation 5.9 on parameters $\mathbf{a}_1$ and $\mathbf{a}_2$ (see Section 5.2), in this case $\mathbf{a}_1 = \mathbf{a}_1$. The sensitivity parameter is t . Further, π is the percentage of depth intensity reduction and ρ is the Gaussian kernel filter size.	42
6.1	Table 6.1 shows comparisons via <i>SSIM</i> and <i>PSNR</i> on <i>Test Set A</i> for different traditional and deep learning models (see Chapter 3 - Section 3.2.2).	46
6.2	Comparisons on <i>Test Set B</i> (See also Chapter 3 - Section 3.2.2). The best results both for mono and stereo models are in bold.	49
6.3	Table of comparisons on <i>Test Set C - MICCAI 2019 EndoVis Grand Challenge</i> dataset with SfM -based GT. The best results both for mono and stereo models are in bold	51
6.4	Comparisons on <i>Test Set C - NAVI and LIGHT</i> challenge between my models and those of Ye et al. (2017) . The best results both for mono and stereo models are in bold.	52
6.5	My models are compared to those one of Ye et al. (2017) on an ensemble of challenging scenes from <i>Test Set D</i> (NAVI,LIGHT,DEFORM - see also Section ??). The scene ensemble (SE) is analysed with and without SMOKE perturbations. For mono and stereo analyses the best results are in bold.	53

6.6	Table of comparisons between depth estimations D_s^{30} ad D_s^{60} on samples with smoke of Figure 6.3.	55
-----	--	----

According to the Italian tradition, this doctoral thesis is written as a monograph, therefore many concepts, theories, experimental results although they have been reported in the respective publications, in this work will be deepened and, where possible, rearranged in order to produce a document as much homogeneous as possible. Furthermore, as shown in the list of publications, my research work did not stop only at the contents described in the thesis but also embraced other themes such as the research and discovery of spatial multi-omic oscillations within bacterial metabolic networks, etc ...

CHAPTER 1

Introduction

My thesis is oriented on a specific research field which is aimed at improving methodologies and applications for depth estimation and non-rigid 3D reconstruction of soft-tissues and organs in 3D laparoscopy. This topic falls into the main category of applications related to computer assisted medicine (CAM). In general, CAM aims to improve several aspects in surgery for what is concerning patient/surgeon clinical work-flow. CAM and also, computer-aided diagnosis, interact in all their procedural steps with human features which have specific and personalized characteristics. If on one hand, every step in an surgery work-flow (for example, the way in which a surgeon plan an operation, the type of cancer to be removed, the surgical tools) have some common points, on the other hand, every surgery work-flow is patient specific and deeply different at several application levels. This is the main reason why computer-assisted medicine presents several still open challenges. Currently, due to the intrinsic and peculiar difficulties of modern surgery, such as minimally invasive surgery, the extraction of depth maps observing soft-tissues and organs and their possible 3D reconstruction is an open challenge. Although, several solutions have been produced, none appear to be the definitive one. Far it is from me to think that the solutions I am about to propose are the definitive ones, however, as I will explain, they will lead to significant research advances. Studying and deepening this work, given its possible application to human health and the benefits that patient and surgeon could derive, made me reflect a lot on its underlying importance. In my thesis I focused my attention on laparoscopic surgery for multiple reasons related to a targeted data collection and by following common and well-known research lines. Laparoscopic surgery is like *E. Coli* for biologists and bioinformaticians; it is a complex case study with its intrinsic difficulties but also with a large collection of researches to be compared which are produced by thousands of research groups during the years. However, I urge the reader to do not consider the choice of laparoscopic application domain as a limiting factor due to limited intervention area (abdomen and pelvic cavities). In fact, I will describe some applications and strategies that could be generalized on different minimally invasive surgeries guided by depth estimations and non-rigid 3D reconstructions. The first question that might intrigue the reader is: why depth estimations and non-rigid 3D reconstructions are so important in

minimally invasive surgery? The answer tells a little about the history of surgical interventions and their evolution over the years that goes beyond this thesis. However, the need of minimally-invasive surgery embraces the why of computer-assisted medicine for depth estimation and 3D reconstruction cause of several aspects related each other. I can find a simple explanation by making the following differences: **(i)** In traditional surgery, open interventions require major cutting of skin and underlying tissues to reach the region of interest (ROI). An operation of this type would not require additional instruments for the calculation of the depth because it would work through the direct observation of the surgeon on ROIs, but on the other hand it could cause many post-operative problems for the patient and a long hospital stay. **(ii)** In minimally invasive surgery (MIS) the access to ROIs is indirect and mediated by laparoscopic instruments or robots. Thus, MIS procedures requires minimal incisions to insert instruments and abdominal gas insufflation to provide an adequate internal working volume without additional cuts. In this way, MIS generally avoids excessive bleeding and reduces operative risks, recovery time, pain and complications due to infections. The indirect surgical access with robotic arms or handheld instruments is guided by a visual device (endoscopic camera) that project the intervention areas and ROIs on monitors installed in operation rooms. Without additional instruments (like controlled light projectors or additional sensors (i.e. Time-Of-Flight Sensors), it is possible to leverage the two-dimensional information provided by the endoscopic images (in mono or stereo) to produce depth estimations and 3D reconstructions. On the other hands, MIS has several surgery limitations, including reduced spatial awareness, field of view, and lack of haptic feedback. Furthermore, the surgeon works with very few degree of freedom (DoF) (Bensignor et al. 2016) and with a reduced depth perception (Bogdanova et al. 2016).

In Section 1.1, the application domain is specified and the problem context is summarized. The main research challenges are described in Section 1.2. Furthermore, research objectives are described in Section 1.3. The research advances introduced in this work are summarized in Section 1.4. The thesis structure is reported in Section 1.4

*

1.1 Application Domain

In the following list of points I have collected the main application domains oriented to improve the clinical work-flow and contextually support the surgeon in safeguarding the patient:

- *3D measurements*: The environment of minimally invasive surgery is limited by space, so 3D exploration and measurements (anatomical zoom, roto-translation, focus on particular details, etc.) can cope with the few degrees of freedom induced by the use of handheld instruments or robotic arms. Once the laparoscope

is inserted, for example in the abdominal cavity, the depth information can help to optimally choose the access points of the other robotic arms (Park et al. 2016, Mahmoud et al. 2016). 3D information is not only used passively. For example, it is possible to improve and guide certain automatic actions such as suturing a specific area of the soft-tissues (Lu et al. 2019). Obviously, this can help in speeding up the suturing time avoiding additional complications.

- *3D surgical planning in real-time:* Despite all the preoperative process and the traditional surgical planning oriented by MRI or CT, a series of unexpected variables can coexist during an operation and can lead the surgeon to make very important and timely choices. In these cases, an enhanced sensory feedback related to ROI volumes (Wagner et al. (2018) as well as for evaluating the proximity and distance between structures (Foti et al. 2020, Luo et al. 2020) helps in obtaining quantitative and qualitative information. For example, depth information can help to track organs in a perturbed environment (Collins et al. 2016) and to prevent the accidental robotic arm collisions on surfaces (Jo et al. 2019).
- *3D depth perception and surface reconstruction:* Patient tissues and organ diseases have different features, volume and locations, thus increasing the need for accurate, adaptive and personalized 3D representations (Collins and Bartoli 2012, Sinha et al. 2017). The reconstructed 3D surface allows for a better perception of spatiality and intervention objectives (Maier-Hein et al. 2013, Westebring-van der Putten et al. 2008). The more accurate is the depth information, the more reliable is the 3D surface reconstruction .
- *Pre-operative or intra-operative 3D registration:* In augmented reality (AR)-guided surgery, 3D surface information allows performing a surface-to-surface or surface-to-volume registration of pre-operative or intra-operative models (Bernhardt et al. 2017, Chhatkuli et al. 2014). These type of registrations support accurate and patient-specific operations, allowing the surgeon to both evaluate and plan strategies in real-time (Prevost et al. 2019, Leonard et al. 2018). In this case, registration to pre-operative/intra-operative models requires accurate depth reconstruction which, often, is achieved by exploiting multi-view methods. The latters are based on fusing depth maps (Lo et al. 2008), structure-from-motion (Malti et al. 2011) or simultaneous localization and mapping (SLAM) (Mahmoud et al. 2016) approaches. In general, traditional models require various preliminary information on the adopted camera that weighs on the clinical workflow (Garg et al. 2016, Prevost et al. 2019).

1.2 Research challenges

In a certain sense, CAM has sought to overcome problems described in Section 1.1 by increasing the surgeon's visual perception with a certain accuracy (Luo et al. 2020, Chadebecq et al. 2020). From this point of view, my research work points to this main objective: how to improve this accuracy? Although, on one hand, I have already hinted that in MIS the least invasive tool for obtaining 3D information comes from laparoscopic images, on the other hands, this approach is medically and technically challenging. The main reasons come from the different conditions/perturbations and hardware limitations that could invalidate the depth maps and make the 3D reconstruction completely dysfunctional and erroneous; I have found four main challenging typologies that could affect depth estimations: (*Type A*) - Intrinsic, (*Type B*) - Extrinsic, (*Type C*) - Induced, (*Type D*) - Technologic. In *Type A* there are all the conditions and perturbations observed by the laparoscope and derived from intrinsic organs/soft-tissues properties. In *Type B* there are all the conditions and perturbations generated during the images acquisition and caused by the nature of the laparoscope for example: type of optics, lens distortions and different light systems adopted. In *Type C* are collected all the actions performed by the surgeon or external factors that could induce additional perturbations in the different way laparoscope acquires images with respect standard conditions. In *Type D* there are all the technical constraints and challenges emerged both from the algorithmic and engineering point of view.

Perturbations of *Type A*, *Type B*, *Type C* and *Type D* could be mixed generating combined challenges. It is possible to verify that for each of these 4 different types of challenges there are different classes of algorithms, research fields and research lines (Maier-Hein et al. 2014, Bernhardt et al. 2017, Ali et al. 2021). However, going into detail, it is not possible to list all the specific perturbations and technical challenges, however below I will synthesize the most common ones found in the Literature with some brief comments on some recent proposed solutions:

- *Non-rigid tissues and organs* - (*Type A and C*) Presence of organ and soft tissue deformations is the biggest challenge in depth estimation for 3D reconstruction and registration (Bernhardt et al. 2017). Deformations can be defined according to different scales. They can be induced by the surgeon and range from the simple displacement of an internal element (eg hepatic flap) to the complete removal of one of the ROI parts (eg tumor resection). Furthermore, the same tissues and organs are not rigid, for example due to the gas insufflation their arrangement with respect to an initial state (such for example recorded in a pre-operative image) moves and expands. Additionally, both tissues and organs show varying degrees of motion due to pulsations and spasms (Maier-Hein et al. 2014). Recently several papers have come to the conclusion that for a non-rigid registration a non-rigid

3D reconstruction is mandatory ([Kajihara et al. 2019](#)).

- *Specular reflections and other artifacts - (Type A)*: The endoscopic light is not reflected homogeneously due to the intrinsic characteristics of the surfaces and organs (eg wet and irregular). Many depth estimation algorithms work on an exemplification of this problem by considering the light reflected equally on all the points of the surface (Lambertian reflectance) ([Maier-Hein et al. 2013](#)). In general the typical surface reflectance in real endoscopic video generates specular reflections. In my opinion this is one the most relevant artifacts that could be found in an endoscopic video. A concrete classification of other possible artifacts is provided in [Ali et al. \(2021\)](#). However, as I will show in Chapter 2, there are few works in the Literature that combine deep learning for depth estimation and artifact removal directly on depth maps.
- *Poor lighting - (Type B)*: *In Vivo* problems in acquiring accurate depth maps can arise from a lack of homogeneous distribution of light on the surface ([Maier-Hein et al. 2013](#)). This can be derived from poor lighting conditions, which are generated due to the limited working-volume and hardware miniaturization. Some works are introduced to solve this specific issue, for example in [Liu et al. \(2017\)](#)
- *Mono or stereo laparoscope characteristics - (Type B)*: In general, stereo 3D depth estimation is more reliable than mono depth estimations, providing both a wider field of view and a greater stability ([Mayer et al. 2016](#), [Lin et al. 2016](#), [Stoyanov et al. 2010](#)). However, advances in laparoscopy robotic engineering both tend to reduce lens distortions ([Qin and Hua 2016](#)) and to miniaturize instruments and camera. A limited baseline (distance between the left and right camera) could affect traditional stereo matching algorithms and depth predictions, however, some valid solutions based on subpixels accuracy are introduced in the literature ([Maier-Hein et al. 2013](#), [Sabater et al. 2011](#)). In most neural network models, similarly to traditional models, a sufficiently large baseline, in widening the field of view, carries more information increasing accuracy ([Brandao et al. 2019](#)). However, an increasingly smaller baseline leads these models to converge towards a mono-based deep learning model. Generally, the latter works better than mono-based traditional models ([Zhou et al. 2020](#), [Laga et al. 2020](#))
- *Blood covering, occlusion and smoke - (Type C)*: In general, as a consequence of the surgical intervention in the different operative phases, the internal scene and the consequent estimation of the depth map is perturbed by three main conditions: **(i)** the haemorrhages (which alter the observability both of target surfaces and the ends of surgical tools), **(ii)** the total or partial occlusions (due to poor maneuverability, some elements of the surgical scene can be placed between the

camera and the ROIs for a reasonable amount of time), **(iii)** the smoke generated by electrocautery. From the removal of smoke to the estimation of depth in partial occlusion conditions, the literature is very wide for what concerns traditional algorithms and models based on neural networks, for both stereo and mono images (Vishal et al. 2020, Wang et al. 2016, Heiselman et al. 2017, Maier-Hein et al. 2014).

- *Real-time and other technical limitations or constraints- (Type D)*: In addition to the accuracy problems, which are very challenging in CAM, real time constraint must be respected (Maier-Hein et al. 2013, Mahmoud et al. 2016). Algorithms that estimate depth, during the surgical workflow, cannot slow down the surgical work-flow and cannot provide deferred information while the surgeon operates for obvious practical and functional reasons. Current technological advances in the use of neural networks paired with next-generation graphics cards (such as the NVIDIA RTX generation), are approaching in real time even on mid-budget GPUs, when optimized and balanced software solutions are adopted.

1.3 Research objectives

During my research I found a clear diachronicity between the research carried out in the medical field and the same type of research carried out, for example, in the automotive field. Neural networks applied for real-time depth estimation are *de facto* the most promising reality in the automotive field for various topics like autonomous vehicles navigation, pedestrian collision detection, etc. (Laga et al. 2020). While, in the endoscopic (and laparoscopic) field, there are very few neural networks-based depth estimation models in the Literature, which become fewer if we focus on encoders/decoders (and autoencoders) self-supervised and / or unsupervised approaches (a complete overview is provided in Chapter 2.) The main reason is that there are no comprehensive datasets with ground truth that can allow unsupervised models to be tested on large-scale samples. On the other hand, in the automotive field there are wide dataset collections with ground truth (for example KITTI Vision Benchmark Suite (Menze and Geiger 2015, Fritsch et al. 2013, Geiger et al. 2012, 2013)). After all, in the automotive field it is much more difficult to find all the complications that we might encounter in a surgical environment and if there are they are very small. The lack of exhaustive datasets in the surgical laparoscopic field derives from the intrinsic nature of the problem: it is not possible for ethical and moral reasons to create ground truth of internal surfaces on living or dead patients. Generally, the datasets available in the literature are without ground truth and in the most part of the cases are built on animal experimentations (like pigs), for example in Ye et al. (2017). Recently, always on pigs, some research centers, under strict restrictions and controls, are trying to create datasets with

ground truth (or fiducials) ([Modrzejewski et al. 2019](#)). However I focus more on that on Chapter 3. For the reasons above, my research can be divided into three related research objectives:

- **(1):** The first objective is characterized by the development and evaluation of unsupervised neural network models (stereo and mono) capable of estimating the depths of soft tissues and organs. Consequently, it is possible to obtain a non-rigid 3D reconstruction.
- **(2):** The second objective is focused on leveraging a fuzzy local adaptive thresholding algorithm (optimized on GPUs) to improve the neural network depth estimations in post-processing. The algorithm has two functions in the pipeline, **(i)** the first is to be able to identify errors on the depth map which are generated by specular reflections or similar artifacts and **(ii)** the second is to locate the elements closest to the laparoscopy and weigh them accordingly.
- **(3):** The third objective is focused on creating a close to reality and more exhaustive simulated dataset to benchmark the models found in the literature or proposed by me.

1.4 Research advances and Thesis organization

In pursuing the research objectives described in Section 1.3 and facing the research challenges described in Section 1.2, I achieved some important research advances, with respect to the state of the art, both in theoretical models and in their applications which can be summarized in the following highlights:

- An advanced deep learning based model able to produce very accurate and close-to-real-time depth estimations.
- A precise procedure for correcting depth maps with respect to specular reflections and luminous objects proximal to the camera;
- The creation of a comprehensive and challenging simulated dataset for performing reliable model benchmarks
- Finally, the testing of models in close-to-real-time situations to evaluate their effectiveness within a surgical workflow. In this video [YouTube Video](#) it is possible to see my model working in simulated and real surgery scenarios.

In Chapter 2 traditional and neural-network related works are introduced. In Chapter 3, the datasets used for training and evaluations are described. In Chapter 4 the models designed in this Thesis are shown. In Chapter 6, the results achieved in this project and several comparisons are described. Finally, the conclusions are in Chapter 7.

CHAPTER 2

Related work and some insights

As explained in Chapter 1, there are different research branches and thousands of methods applied over the years for the depth estimation, for example, based on the calculation of the distance of the markers / fiducials (positioned in specific areas of the ROI) or on the estimation of the three-dimensionality of surfaces and organs, etc. In Chapter 1 - Section 1.1 several possible applications were described. Instead, in this chapter, in Section 2.1, the depth estimation related works are discussed. Furthermore, in Section 2.2, some artifact removal models are described because related to my depth map post processing procedure.

2.1 Dense depth estimation models

The most immediate organization of this section could be achieved through the separation of the methods based on the extraction of features from the methods based on feature learning. But this type of separation would not be able to grasp the differences and potential of the various methods proposed.

The idea is therefore to use a better known separation in the medical imaging community, that is to separate methods that produce dense depth maps from those that produce sparse depth maps. The sparse methods are not completely comparable with my models and so they will be treated only marginally. Subsequently, within the dense methods, a further separation of the models is performed considering those for monocular and stereo images and keeping the traditional algorithms separate from the methods based on supervised, semi-supervised and unsupervised learning. In detail, only encoders/decoders (and autoencoders) methods are enumerated. This choice is driven by the application domain and the number of related works found in the Literature. However, throughout the text we will also marginally discuss deep learning based approaches that do not necessarily involve autoencoders or encoder / decoder architectures. Importantly, much of the text in Sections 2.1.1, 2.1.2 is from [Bardozzo et al. \(2021\)](#). This paper has been submitted to *Medical Image Analysis - Elsevier*.

2.1.1 Depth-map reconstruction with monocular images

2.1.1.1 Traditional methods

In the literature, traditional methods are considered, for example, *SLAM*, *SfX*, where X could be a shading, motion, texture visual cues, Bayesian methods and classical variational methods from computer vision. An example of the latter group is the paper of [Stoyanov et al. \(2010\)](#), where the authors propose a semi-dense reconstruction approach, where sparse points are found with a traditional approach and considered as *diffusion points*; then, from these points the disparity information is propagated with a region growing method. In [Mahmoud et al. \(2016\)](#), real-time endoscope position in 3D space, with respect to an estimated 3D point cloud for soft tissues and organs (also textureless), is obtained leveraging ORB features [Chien et al. \(2016\)](#) and *SLAM* methodology combined with dense multi-view stereo depth maps.

2.1.1.2 Supervised and semi-supervised DL-based methods

Recently, promising methods have been introduced in several computer vision fields and, in particular, in industrial and automotive applications, by combining traditional methods with deep learning. For example, [Kobler et al. \(2017\)](#) designed a model to combine variational methods with deep learning. Moreover, in most cases, traditional monocular methods are used in the training phase as proxy labels to replace unlabeled data (compensating for the lack of ground truth), or in pre- or post processing for more accurate predictions ([Tateno et al. 2017](#), [Duan et al. 2019](#), [Bahraini et al. 2019](#), [Tosi et al. 2019](#)). For example, the DL-based method of [Hager et al. \(2018\)](#) is semi-supervised and it improves depth estimations by leveraging the information obtained with a Structure-from-Motion based method ([Leonard et al. 2018](#)).

2.1.1.3 Unsupervised/self-supervised DL-based methods

A well-known unsupervised model based on autoencoders (*AE*) is proposed by [Garg et al. \(2016\)](#); in this case, adjacent frames at time t_i and t_{i+1} in a mono video are viewed as displaced stereo pairs (I_L and I_R). I_L is considered as input image, I_R is considered as target. The *AE* reconstruction loss (photometric error) leverages the displacement field obtained by backward warping the predicted depth map with respect to the target image. In detail, a displacement field is defined as the direction (backward/foreward) and the distance that every pixel has to warp to match the target image ([Garg et al. 2016](#), [Schwarz 2007](#)). Later, [Godard et al. \(2017\)](#) introduced a similar work for *AE* depth estimation with left-right consistency loss. In a similar way, [Yusiong et al. Yusiong and Naval \(2019\)](#) introduced an *Inception*-like stacked *AE* for monocular depth estimation. In general, single or stacked *AE* could be considered as the building blocks for stereo/si-

amense networks. Given the lack of stereo information that can be exploited at least in the training phase (Tosi et al. 2019), a mono depth estimation problem is often defined ill posed and leaves some challenges opened.

2.1.2 Depth-map reconstruction with binocular images

2.1.2.1 Traditional methods

Traditional methods leverage techniques like block matching, graph cuts, semi-global matching, Bayesian methods and variational methods (Hamzah and Ibrahim 2016). For example, Geiger et al. (2010) proposed a Bayesian method for an efficient stereo matching and sparse reconstruction. Furthermore, in Wang et al. (2018), endoscopic depth maps are reconstructed through an estimate of variational disparity in a multi-resolution pyramidal representation of stereo images. The disparity estimation method goal is to minimize a modification of the global energy cost function. In Yamaguchi et al. (2014), the depth maps are estimated by developing a semi-global block matching algorithm based on the joint evidence of the stereo and video pairs. In general, if these traditional monocular and stereo models provide sparse disparity estimations, on the other hand they also provide possible solutions to their sparsity.

2.1.2.2 Supervised and semi-supervised DL-based methods

A first attempt of a deep neural network-based 3D non-rigid organ reconstruction can be found in Antal (2016, 2017). The approach is based on training a six layer deep neural network with heart beating stereo image pixel intensities combined with three-dimensional information (Pratt et al. 2010, Stoyanov et al. 2010). The network provides the 3D coordinates of each pixel. Unfortunately, for the reasons above explained, three-dimensional and/or depth map GT is often unavailable leading to the need of unsupervised/self-supervised or semi-supervised models. For example, Luo et al. (2019) cope with the lack of ground truth by replacing it with proxy disparity labels to improve AE depth predictions (Tosi et al. 2019). Pseudo labels are computed with different traditional algorithms and used in the supervised training phase as a sort of *loss driver*. In particular, Luo et al. (2019) adopted some transformations of the ad-census stereo algorithm originally designed by Zabih and Woodfill (1994)..

2.1.2.3 Unsupervised/self-supervised DL-based methods

Ye et al. (2017), inspired by Garg et al. (2016), introduced the first work on binocular laparoscopic depth estimation based on self-supervised Siamese networks (Ye et al. 2017). In their work, two equal AE with shared weights (defined as Siamese) are designed. Xu et al. (2019) introduced further advances exploiting a Siamese-like structure

with the addition of skip connections as in a *ResNet*-like architecture. Furthermore, [Goldman et al. \(2019\)](#) introduced additional improvements by adapting a *ResNet*-like approach to a [Godard et al. \(2017\)](#) architecture to learn stereo and infer mono. Similarly to models described in [Laga et al. \(2020\)](#), most of the models compared in this work and in other application domains have some limitations that should be addressed: for example, these models are too sensitive to complex lighting conditions and surface reflectance properties. Moreover, as will be discussed in Section 6, estimated depths are often smoothed, noisy and show incorrect or deformed object positions. Moreover, these related models ([Ye et al. 2017](#), [Luo et al. 2019](#), [Xu et al. 2019](#), [Godard et al. 2017](#)) show low-depth resolutions, non-continuous depth levels on continuous surfaces, and generate artifacts. For example, the depth maps of [Ye et al. \(2017\)](#) have black bars on the left or right side, as shown in Figure 6.4. In the same Figure, it is possible to observe that the black bars induce distortions in adjacent areas, displacements of the object positions with respect to the original scene and narrowing the field of view for depth estimation.

2.2 Pre- and post-processing models

Artifacts generated by specular reflections and other types of artifacts in traditional and deep learning models are generally removed during the pre-processing with the aim of improving final results. Recent deep learning encoder/decoder and autoencoder architectures are quite stable to different types of perturbations induced by the presence of artifacts. For this reason many of the deep learning models described in Section 2.1 are able to make reasonable depth estimations in low light conditions, specular reflections, occlusions, smoke, bleeding, etc... without the need for pre- or post-processing for artifacts removal. See also [Ye et al. \(2017\)](#), [Xu et al. \(2019\)](#), [Luo et al. \(2019\)](#), [Laga et al. \(2020\)](#). This does not mean that they are infallible: in my Thesis I will show that in some conditions a soft post-processing can "slightly" improve the depth estimates done by my models. The literature is very extensive in this regard and goes beyond the scope of this Thesis. Moreover, although the removal of artifacts is a much-discussed topic in the literature, the same removal of artifacts has been frequently applied to original images and not to depth maps as in this Thesis. I proceed by following some key points and showing what kind of depth map post-processing I have chosen to improve depth accuracy. A first example could be found in [Barbalata and Mattos \(2014\)](#). In this case, a specular reflection removal algorithm based on filters and binary masks is introduced to improve the classification of laryngeal tumours. From this work, I got the idea of applying an additional filter and obtaining a binary mask to be superimposed on the depth map for its correction but, in my case, the procedure had to be further refined. Recently, [Mahmood and Durr \(2018\)](#) used digital image painting as pre-processing to improve depth estimates obtained with adversarial neural networks. Few years later, [Ali](#)

et al. (2021) introduce generative adversarial networks that are trained both to isolate artifact typologies and to apply video endoscopy restoration. In this case, the artifacts are separated into different types and consequently processed in different ways. In general, modern and deep learning based techniques have shifted the focus from pixel colour distribution analyses and image painting to more affordable feature driven methods focused on deterministic adaptive algorithms and/or, only partly, on gradient-based models. This because some transformations induced by traditional models, such as the inpainting ones, or deep learning models, cause the alteration of the original content of the image by inducing a sort of reconstruction/estimation of the corrupted parts.

In Li et al. (2019) and Tan et al. (2017), the specular reflections are removed with a model based on adaptive robust principal component analysis. Furthermore, in Asif et al. (2020) specular reflections of human colorectal endoscopic images are detected by using the intrinsic image layer separation model oriented to the extraction of high gradient area. Always in 2020, I published a paper in Information Fusion Journal that is capable of foreground/background segmentation with a very high accuracy. This method is based on adaptive fuzzy integral-based images obtained from the generalizations of Sugeno and Choquet integrals (Bardozzo et al. 2020). I found this work very interesting also for its important applicative contribution to this Thesis. In fact, with a correct parametrization, this novel algorithm is able to detect proximal elements to the endoscopic camera and specular reflections. This can help in post-processing to fix depth estimations. The fuzzy algorithm works on gray scale depths instead of pre-processing RGB images by recognizing the brighter area affected by specular reflections or with a relevant proximity to the camera. I omit the part of the related works for the specific background/foreground segmentation present in the literature because they have already been published in Bardozzo et al. (2020). However I will provide specific details on the implementation of this algorithm and its results combined with my deep learning model in Chapter 5.

2.3 Model proposal positioning

In this work I propose two mono and stereo models (deep learning networks) that are positioned in the class of the unsupervised ones (see also Section 2.1.2.3). Although they may take their cue from Siamese autoencoders of Ye et al. (2017), Xu et al. (2019), they are more oriented in the encoder / decoder classes by taking advantage of the learning of the displacement field (left-right consistency loss) used in Godard et al. (2017, 2019). Compared to the previous state-of-the-art autoencoders (mentioned above), the encoder/decoder models I propose present a reduced number of convolutional layers, resulting in a certain sense, wider than deep. They do not present batch normalization like the previous ones and exploit, in an engineered way, a specific positioning of the skip

connections between encoders and decoders. The same loss in both stereo and mono models has been changed with respect to previous architectures and empirically configured following an extensive series of experiment grids on Marconi100 (supercomputer of SCAI CINECA). The Siamese configuration (especially in the stereo model) not only presents (in late integration) parameter sharing, but converges by stereo correspondence into a single image (depth estimation). For this task, I designed a specific combined loss function oriented to minimize typical reconstruction error functions and to obtain more stable solutions on the edges. The concept of late integration is inspired by the X-CNN Cross-modal convolutional neural networks of [Veličković et al. \(2016\)](#) (even if left and right channels in cross-relation are not effectively different modalities.) Moreover, the encoders / decoders in the stereo pipeline and along the stereo model have been arranged in a stacked structure which ends with the application of an advanced stereo matching layer consisting of a series of concatenation and dense layers which is inspired but not equal to [Žbontar and LeCun \(2016\)](#). There are several works on the use of stacked autoencoders or encoders / decoders, but the ones that have particularly aroused my interest are those of [Maimaitimin et al. \(2017\)](#), [Yusiong and Naval \(2019\)](#). As I will describe in the following chapters, many novelties in terms of model configurations and design can be located not only in the architectural choices with respect state-of-the-art but, also, in those oriented to the specific application domain. Finally, the interaction with the fuzzy algorithm to correct the depth map estimations, although it falls within a specific range of models described above, is completely innovative due to its configuration and within this application domain. Many details and comparisons can be found directly in the Chapters dedicated to models (Chapter 5) and to the processing (Chapter 4).

CHAPTER 3

Datasets

The major part of laparoscopic and surgical depth estimation models have been usually designed to work on datasets without ground truth showing unsupervised or semi-supervised characteristics. This is because, as I described in the Introduction 1, the acquisition of datasets with ground truth is very limited on humans (and often on animals). The unlabeled training set used in this Thesis is described in Section 3.1 and the test set without ground truth is described in Section 3.2.1. On the other hand, to evaluate the sensitivity and robustness of my models, I tested the results also on some available datasets with ground truth. The latters are described in Section 3.2.2. However, as I will describe in the next sections, these datasets have some ground truth highly imprecise. For this reason, in Section 3.3, I introduced a novel simulated test set with ground truth. As far as is my knowledge, this type of dataset has been never proposed in the Literature. It is my opinion that the use of this type of test set can improve model benchmarking for depth estimation and 3D reconstruction in terms of accuracy, robustness and sensitivity.

3.1 Training set

My models have been trained on the human partial nephrectomy dataset available on the repository of the Hamlyn Center Imperial College of London. The training set is composed of 34.240 pairs of rectified stereo images of size $j = 384 \times k = 192$ pixels sampled from 11 *in Vivo* surgery operations. The data set used is not heterogeneous and, because it was sampled at a high frame-rate, many of the adjacent sampled frames are redundant. This could lead networks in training stages to give more importance to the most frequent features rather than the rarer ones. [Ye et al. \(2017\)](#) tried to solve this problem using random sampling strategies. However, instead of adopting this strategy, I preferred to solve the problem from the point of view of the model by choosing an optimizer capable of handling redundancy. In Figure 3.1 sample selected from the training images are provided. In Section 3.2.1, the unlabeled test set samples are described.

3.2 Test sets

This Section describes the two different types of test sets adopted, their subjects and the acquisition methods. In general, to avoid bias, test sets with and without ground truth are separated from the training set, ensuring the independence of the images used for training from those used for testing.

3.2.1 Test set without ground truth

The images belonging to the *unlabelled* test sets are usually of animals and humans. In my case, the test set is composed of 7.191 unlabeled stereo images always provided by the Hamlyn Center (see also Section 3.1). This images are of the same type of those used for training. The same test set is used in the works of (Xu et al. 2019, Luo et al. 2019, Ye et al. 2017). This is a fairly large test set compared to other datasets and also in this case all the images are already stereo rectified. As it is possible to observe in Figure 3.1, the dataset is challenging because of the presence of real specular reflections, blood, partial instrument occlusions and smoke.

3.2.2 In Silico and In Vivo test sets with ground truth

The other solutions work around the creation of new test sets with ground truth paying attention to some relevant characteristics and challenges for the application domain. There are several researches in which test sets with ground truth are acquired considering three-dimensional models (phantom). In these cases, the ground truths are acquired with the same acquisition techniques used for humans and animals (computer tomography (CT), etc...) (Pratt et al. 2010) or, for example, with laser scanners (Penza et al. 2018). Other invasive techniques, exclusively on animals, provide the possibility of operating with an open abdomen and/or by using additional sensors for the acquisition of ground truth (such as structured lights (Lin et al. 2015)). In my Thesis, I tested my models on the following publicly available datasets with ground truth. This choice is driven by the fact that these test sets are used in other papers with the same purpose of depth estimation (for example in Luo et al. (2019)).

- *Test Set B*: This is the case of data sets built on phantom models. The first data set considered is of Pratt et al. (2010). In particular, two stereo videos of "ghost of the silicon heart deformed by heart pulses" are recorded. As a result, a series of associated CT (computed tomography) scans are acquired and used as ground truth. Videos are recorded at a frame rate of 1/25 for a total of 3389 images for the first video and 2427 for the second video.
- *Test Set C*: In 2019, *MICCAI Grand Challenge* proposed a new test set of 45 stereo images with ground truth suited for depth estimation and 3D surface re-

Training Set and Test Set A (without ground truth)

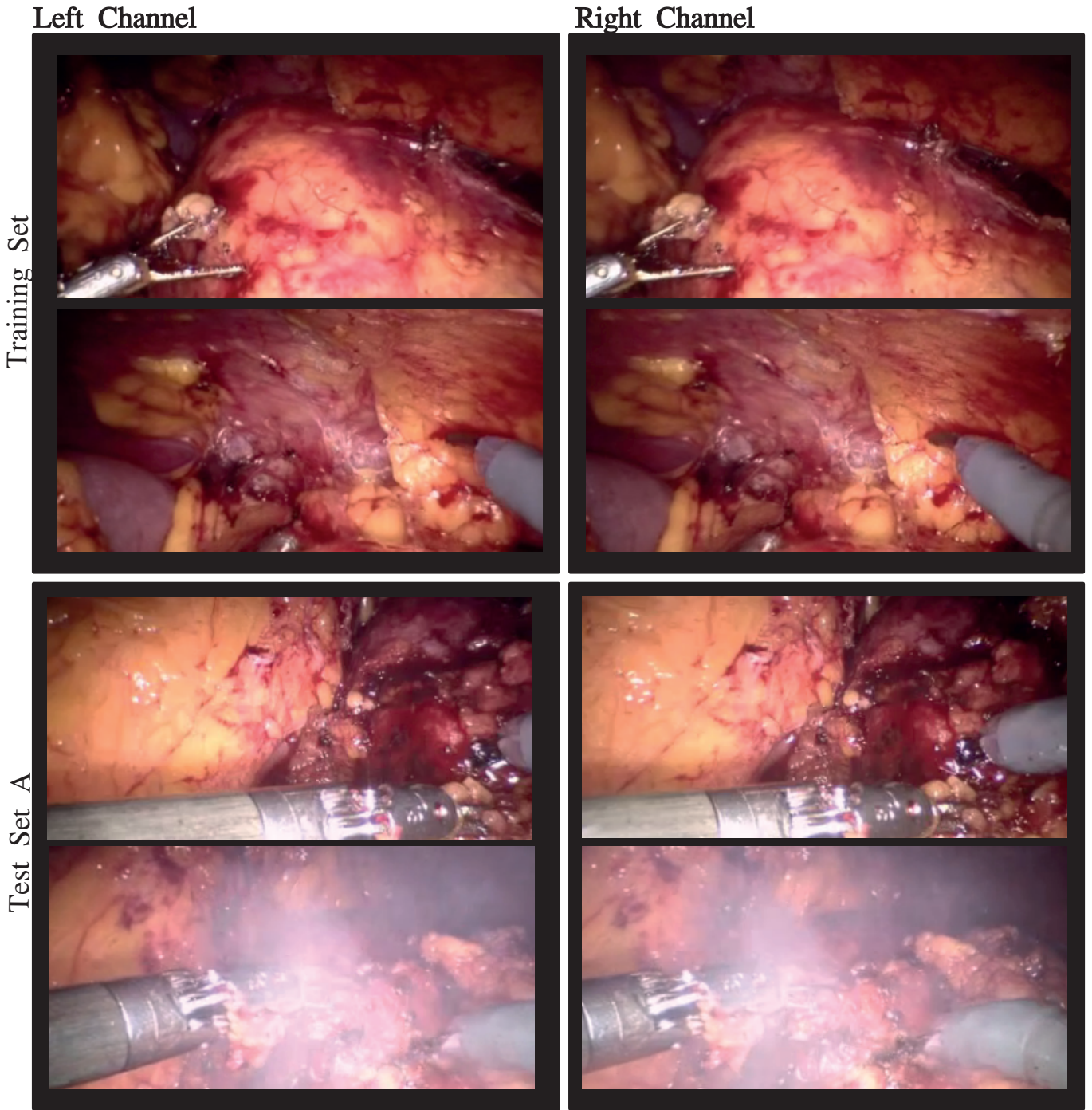


Fig. 3.1 In this Figure left-right channel of *In Vivo* samples for unlabeled training and test set are provided. These scenes are publicly accessible on the Hamlyn Center Vision repository and in particular are provided by [Ye et al. \(2017\)](#)

construction. The 45 stereo images come from three different *in Vivo* porcine endoscopic inspections with a size of 1280×1024 pixels. The ground truths are generated with structured lights with prior knowledge of camera calibration parameters and transforms.

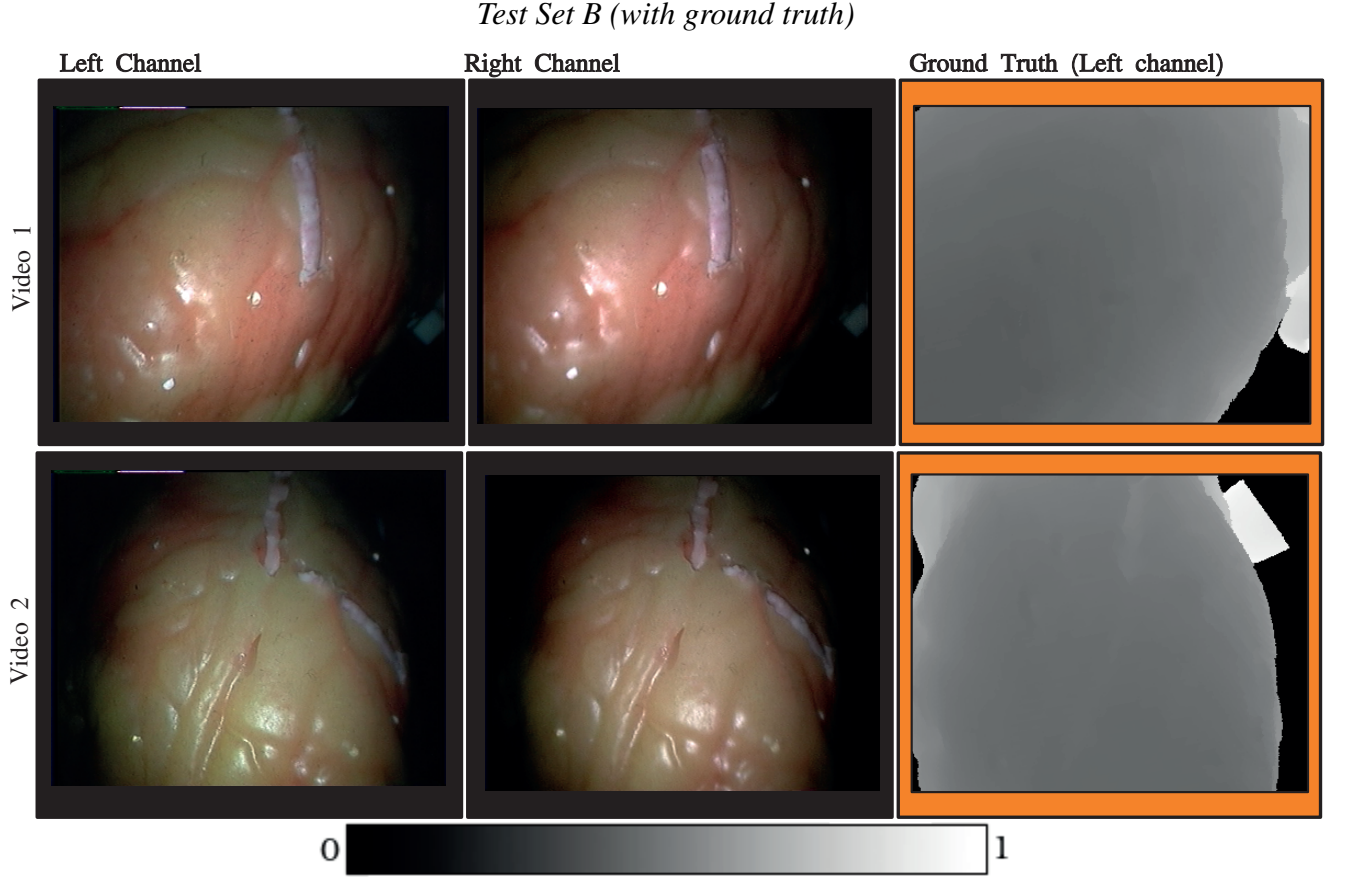


Fig. 3.2 In this Figure left-right channel samples with *In Silico* ground truth for different scenes are provided. These two scenes are publicly accessible on the Hamlyn Center Vision repository and in particular are provided by [Pratt et al. \(2010\)](#).

The stereo images of *Test Set B* have a size of $n = 360 \times m = 288$ pixels, while the stereo images of *Test Set C* have a size of $n = 1280 \times m = 1024$ pixels. For a fair comparison, instead of losing details by center cropping or strong resizing, I produced two consecutive subdivisions and resizing of the stereo images and left channel GT. Therefore, if the original size of the images/left GT is $(n \times m)$, two images with size $(n \times \frac{m}{2})$ are obtained. Subsequently, to fit the model input size $(j \times k)$, these images are scaled with bilinear interpolation to the same model input size: $(n \times \frac{m}{2}) \hookrightarrow (j \times k)$. The reverse procedure is performed at the comparisons phase. Specifically, once my model estimated two depth maps for each original image, these two depth maps are recomposed in one depth map and brought back to the original size to be compared with the ground truth.

There are several problems on these datasets (especially those *in Silico*), specially for the way they were built, because they could invalidate the models on which they are tested (see also [\[Here\]](#)). This is because the *In Silico* phantom models do not fully reflect the observed reality. For example, the recorded *in Silico* organs are suspended in the void, without surrounding elements, thus leading to unrealistic depth estimations.

Test Set C (with ground truth)

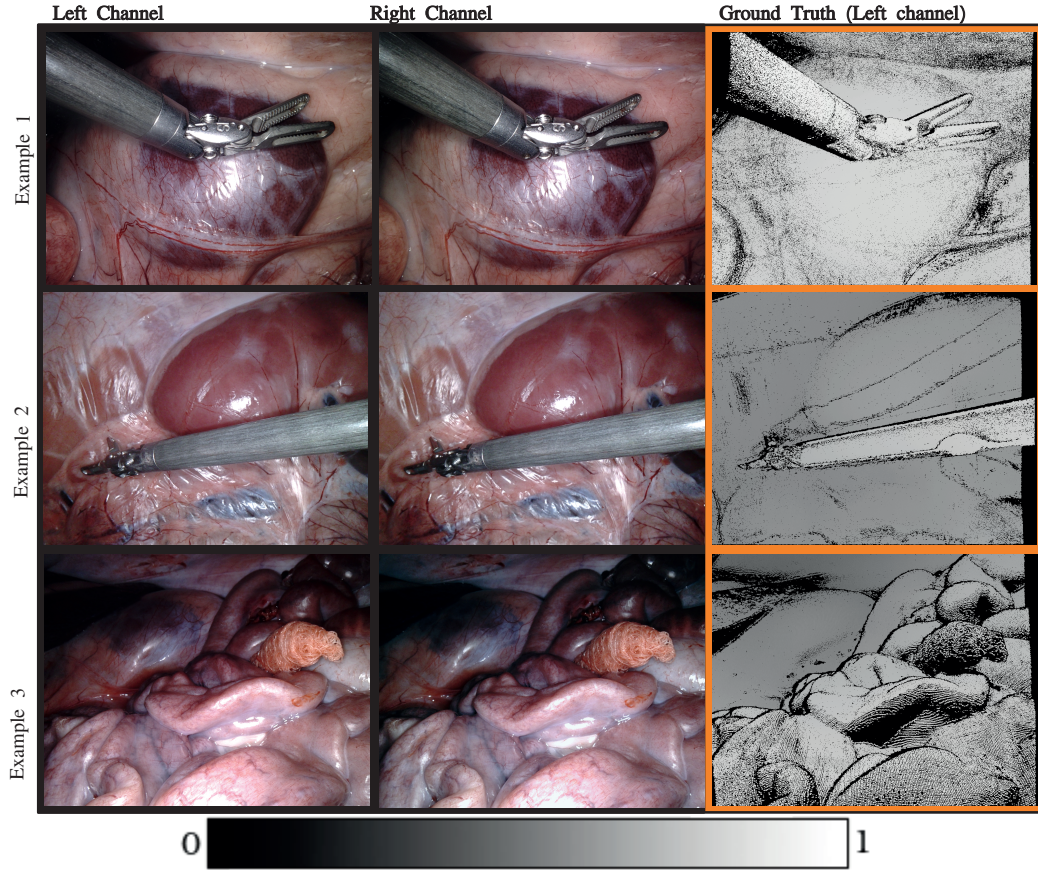


Fig. 3.3 In this Figure three left-right channel *In Vivo* samples with structured-light-based ground truth for different scenes are provided. This dataset is under gentle concession of *MICCAI Endovis Grand Challenge 2019*.

Moreover, *in Silico* surfaces do not fully reproduce a real surface due to the materials used, thus also altering the way the light is reflected in real laparoscopes, etc.. I partially agree with the presence of these problems. I think that the main weaknesses of these strategies are caused by the imperfect ground truths which are produced with several strategies. For example, in Figure 3.2, it is evident that in both the examples of Video 1 and Video 2 of *Test Set B* there are some characteristics in the depth map that are not perfectly corresponding to what a model can obtain from the 2D image. In this case, the ground truth generated from the computer tomography (CT) goes so deep into 3D modeling that, correctly, it also represents some parts that are invisible and remain invisible from an endoscopic point of view. The risk of using deep-learning-model validation with this test set lies in the fact that the neural network could both generate and estimate parts of the images from scratch, thus generating false positives that could be "correctly" aligned with the ground-truth obtained from CTs. These alignments would not be true at all. This error, in a surgical operation, could lead the surgeon to see in augmented reality the depth of an object that is not there and that, if there is, it is at a different depth, even below the surface. In Figure 3.3, three samples of *Test Set C* are

shown. In this case the ground truth depths are completely discretised to a few depth levels. So there is no detail in the differentiation of depths even if it is evident that the surfaces are continuous. Models that estimate the surface in a continuous field if tested over these ground truths could be both overestimated and/or underestimated. Moreover, always in Figure 3.3, these ground truths generated by structured lights are distorted on the borders. In real-time surgery, this could induce a wrong validated model to represent objects in wrong positions and with unreal depths.

3.3 Simulated test set

In this section I introduce *Test Set D*, which is a close-to-real simulated test set with a noiseless ground truth. In Figure 3.4 there are some samples of the simulated dataset with their ground truths¹. The dataset is born in response to the ground truth issues described in Section 3.2.2. In particular, the simulation represents a 3D environment that reproduces 6 possible internal surfaces with different perturbations, reaching 13 simulated video. The goal of the simulations is to design a truly controlled test set with different difficulty levels that gradually increase. In this way, it is possible to challenge the network and clearly understand where the models win or fail. Some control simulations are then created without a huge presence of artifacts, then artifacts are inserted (endoscopic navigation with changes of lights, specular reflections, smoke, blood, etc...).

3.3.1 Initial setup

The simulated virtual environment is built both by 3D merging and remorphing a liver computer tomography scan provided by [Suwelack et al. \(2014\)](#). In this way it is modeled a near-real 3D structure that can represent anatomical details and environments that could be found in laparoscopic surgery. I have proposed the various simulated scenes to the careful analysis of a group of surgeons and have selected only the scenes that surgeons consider familiar, removing those that seemed far from a surgical and operative reality. The 3D scenes are texturized with real laparoscopic images taken from *Test Set C*. The texturing process took many hours, as the 3D modeled parts had to be aligned collinearly with the textured parts. I had to respect a certain geometry of the 3D environment with respect to the applied texture and apply other precautions that I describe below. For example, some deeper areas in the 3D model might correspond to a darker red area where blood can flow (see scene 6 and scene 8 of Figure 3.4). While, the most prominent parts, or the parts where the endoscope could reflect the light most,

¹. For a better display of depth maps and depth ground truths, in some figures along the thesis, I leverage a colour palette used also into similar works which is called *Inferno* ([Hunter 2007](#)). The bar indicating the intensity of the colours between 0 and 1 is found in Figure 5.3.

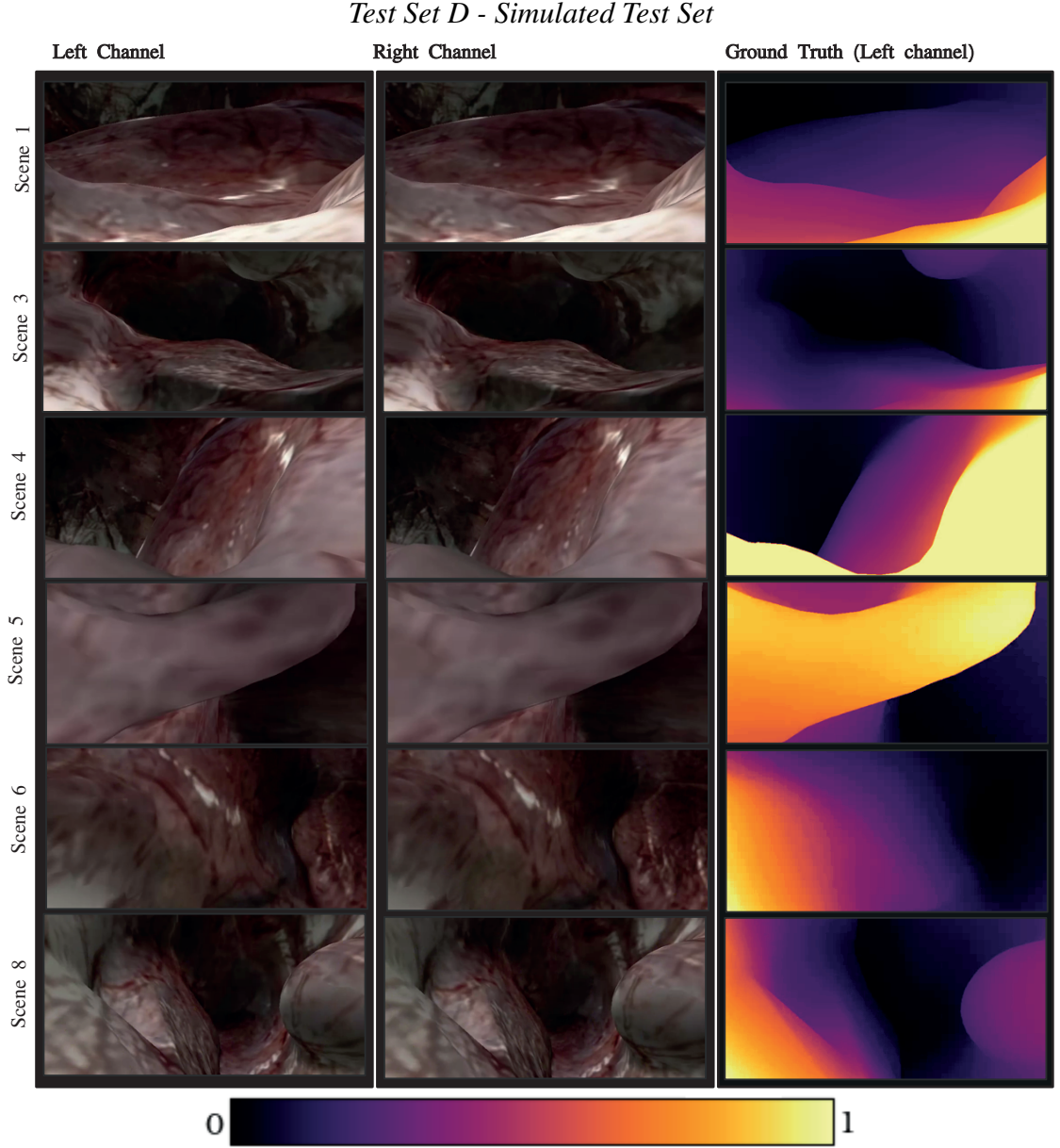


Fig. 3.4 In this Figure left-right channel samples simulated *Test Set D* with related G_1 ground truths ¹.

generally had to be brighter, or less bright if not framed (this is present in all scenes). It all depends on the position of the endoscope, but also on what the scene simulates. To be more clearer, I provide an example of a typical feature of this type of surfaces: being soft tissues, if pressed by a hypothetical surgical instrument, their color becomes lighter because the underlying blood, by pressure, branches off to areas far from the points of pressure. Consequently, more protruding areas may be either lighter or darker due to anatomy, bone tissue and other reasons which are not related only to the visible tridimensional geometry.

3.3.2 Video tipologies and artifacts simulations

All the scenes I produced are texturized to represent lighter, darker, bloody areas and with the presence of different types of tissues, blood vessels of varying sizes and with clearly different geometric shapes. However, in this Section, I classify the scenes by considering the type of challenge that they present. In particular, as explained in the previous paragraph, 13 videos were selected by expert in the field, and these video present different challenges on 6 different scenes which are respectively named: Scene 1, Scene 3, Scene 4, Scene 5, Scene 6, Scene 8 (See also Figure 3.4).

3.3.2.1 NAVI scene challenge

In this case, endoscopic navigation along the vertical and horizontal axis and on its two secant bisectors is simulated. In scene 1, scene 4 and scene 6 the initial position of the stereo camera is central to the coordinates of the left image. In scene 3, the initial position of the stereo camera is slightly off-centre in the upper right corner of the left image. In scene 6, the initial camera position is slightly off-centre in the lower left quadrant. While, in scene 8, the position of the stereo camera is slightly decentered to the lower right of the centre point. In all the quadrats I positioned the camera the shift from the center scenes in coordinates (0,0) is of ± 5 pixels with respect their respective horizontal and vertical axis.

3.3.2.2 LIGHT scene challenge

In all scenes, the light changes are very pronounced. The intensity of the light is organized according to the closest positions of the endoscope. The light diffusion model over the texturized areas is Lambertian (Ikeuchi 2014). The light reflections are made irregular by the presence of specular reflections on the textures. The 3D environment manages the shadows and the brightest area natively by dynamically changing the positions of the light along with the video creation. The movements of the lights are the same as those described for the movements of the stereo camera in the previous subsection. Scene 1, Scene 4 and Scene 6 present specular reflection artifacts. The specular reflections in Scene 3 and Scene 8 are moderate. While in Scene 5 are absent. In particular, several shades have been introduced throughout the scenes which are quite challenging for the models. The most marked is in Scene 3 in Figure 3.4: in the area of the lower right corner the shading has been voluntarily increased with respect to its real depth, which can be ascertained from the ground truth. This is decidedly contrary to a natural situation that can be represented in scene 1 but, in some cases it could occur due to an incorrect position of the endoscope lights and must be taken into consideration. In Figure 3.5 three light position changing samples are shown for two scenes.

3.3.2.3 DEFORM scene challenge

Scenes 4 and Scene 5 fall into category DEFORM. In particular, in scene 4 a deformation similar to that induced by diaphragmatic breathing was generated, therefore very slow and with a very large surface dilation are designed. While, in video 5, a deformation induced by cardiac pulsation and propagated in the circulatory system is generated. Therefore in this case, the type of movement is less slow and more limited both temporally and in the area of action.

3.3.2.4 SMOKE scene challenge

A realistic video of smoke recorded with the alpha channel background for Scene 1 and Scene 4 of Figure 3.4 is superimposed. The smoke is gradually getting stronger until almost the entire scene disappears. It is very similar to a situation that actually exists in the operating environment. Note that the smoke in this area is usually not so dense as to create marked shading on the surface itself, both due to lack of space and the extreme proximity of the endoscope to the electro-cauterized surface. Three examples of gradually superimposed smoke are shown with respect to the original images in Figure 6.3.

3.3.3 Implementation details and configurations

The entire collection of scenes were recorded with vokoscreenNG (link [here](#)) in mkv format at 60 frames per second directly observing the changes induced on MeshLab ([Cignoni et al. 2008](#)) on the simulated environments. A python script has automatically generated parameterizable macros to induce the 3D environment to follow programmed behaviors. Each stereo video was converted to stereo images using ffmpeg (see [here](#)). The induced movements were robotized in order to make them accurately equal between the scenes under analysis. In particular, to robotize the behaviors in the three-dimensional environment of the video, a synchronization system between mouse and keyboard commands on MeshLab was used through the Xmacro library (see [here](#)). The Xmacro C++ library took as input the macros created previously and automatically launched the recording of the scene. The scene recording for each scene occurred 4 times: the first time for the left channel, the second time to record the video of the left channel depthmap, and the same for the right channel. The baseline was varied from 15 to 30 pixels for the different scenes. The ground truth depth maps for the left and right channel were generated by the depthmap MeshLab shader introduced by Valerio Corsini of the team of MeshLab. In particular, the *fragment program* of the shader is configured to represent the 3D environment surface in gray scale. In particular, the colour of each vertex is assigned by weighting the gray scale value in a way directly proportional to the z position of the vertex in a triangle mesh. For all the scenes, the

camera settings and camera intrinsics are provided accompanied with the 3D model on the repository online as MeshLab project files [\[Google Drive\]](#)

Test Set D - three samples with ground truth of LIGHT type scenes

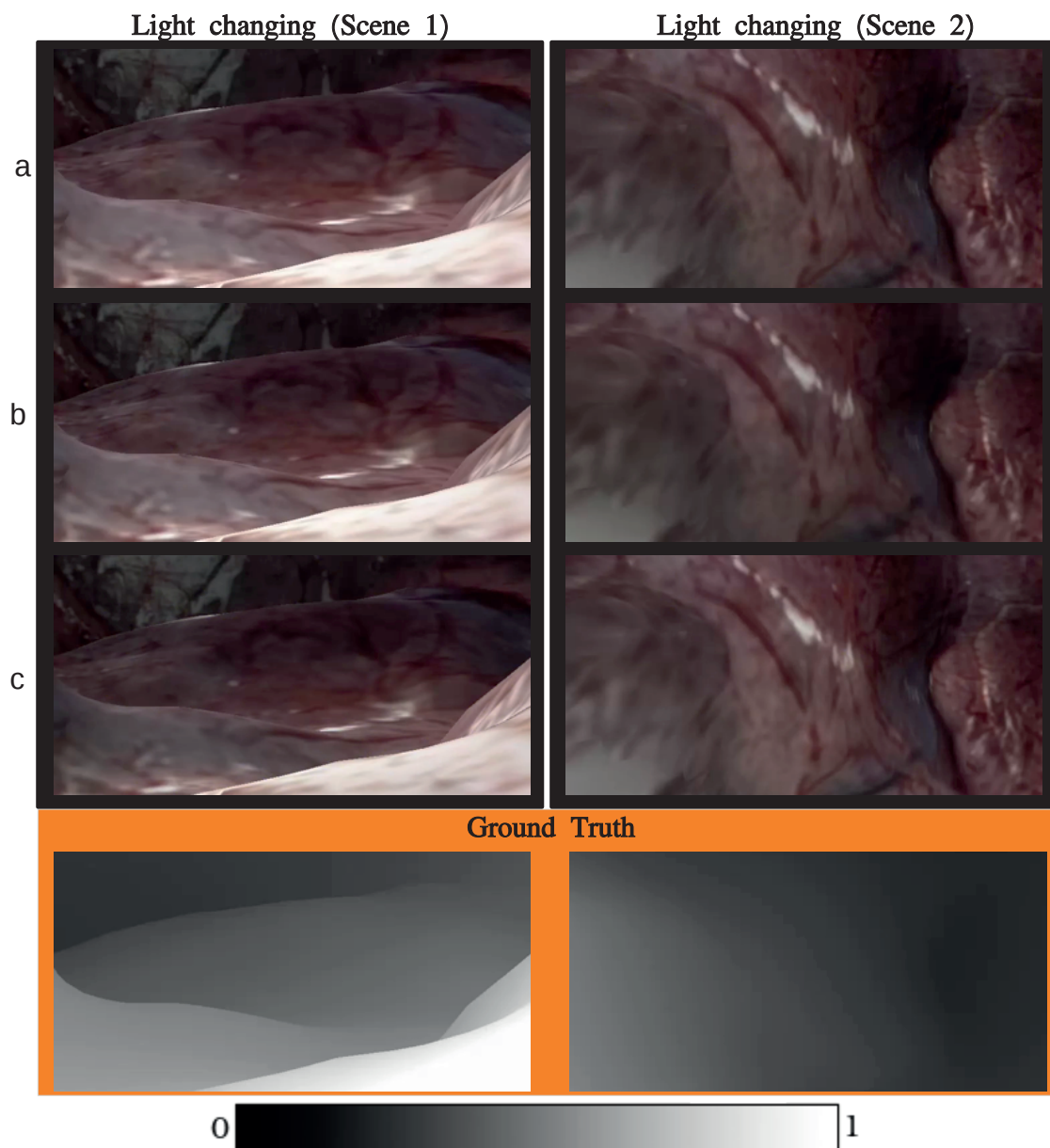


Fig. 3.5 In this Figure three left channel samples with ground truth for LIGHT challenge scenes (at three different endoscopic light positions) of *Test Set D* are provided.

CHAPTER 4

Unsupervised models for Mono and Stereo Depth Estimation

In this chapter, I will introduce two unsupervised deep learning models for depth estimation in laparoscopic mono and stereo configurations. Before proceeding, I would like to spend a few words on what unsupervised learning is: it could be seen as an application of self-taught learning (Raina 2009). In neural networks as in humans, self-taught learning means learning without being guided by additional information or labels connected to what we have to learn. For example, it means learning to play a musical instrument without knowing music theory, the nomenclature of notes or without knowing the score. A single unlabeled example will never get a network or, probably, a human brain to train. The strength of these methods relies on their architectures but also on the number and quality of unlabeled data. In this sense, it can be proved that unsupervised learning, in some situations, leads to better performance than supervised learning which, in turn, is conditioned by massive labeling and data organization (Laga et al. 2020). Given that, I organized the Chapter as follows: The first model I introduce is a mono depth predictor which takes the form of an unsupervised encoder/decoder (*Mono depth CNN* - See Figure 4.1). The model is detailed in Section 4.1. The second model is an unsupervised Siamese encoder/decoder network ending with a stereo matching module (*Stereo depth CNN* - Figure 4.3). The latter is described in Section 4.2. The novelty of the models presented in this thesis is based both in their design but also in their arrangement by stacking encoders/decoders in a pipeline to obtain more accurate predictions (*Stereo depth pipeline* - See Figure 4.2). The pipeline consists of three networks trained separately and only subsequently combined. The first two trained networks are the two *Mono depth CNNs* for left and right mono channel (Block **SN1** in Figure 4.3). The *Stereo depth CNN* is the third trained network (Block **SN2 + SM** in Figure 4.3). The training configurations are described in Section 4.4. The training losses are described in Section 4.3. Some implementation details are provided in Section 4.5. Training time performances and prediction time performances are described in Sections 4.6 and 4.7. Much of the text in this Chapter is taken from Bardozzo et al. (2021). The paper is submitted to *Medical Image Analysis - Elsevier*.

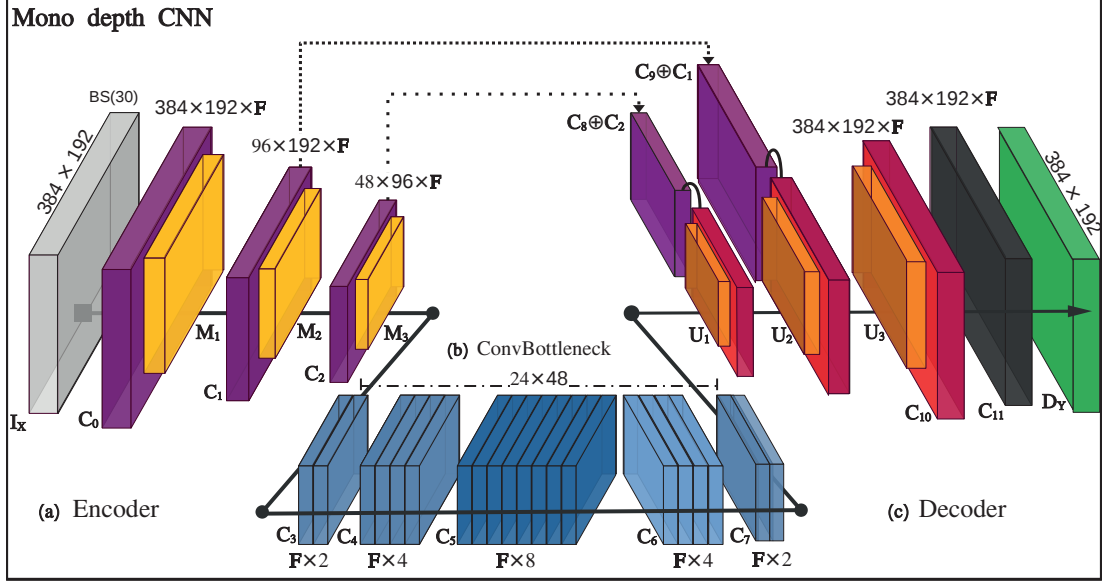


Fig. 4.1 In Figure 4.1 show the *Mono depth CNN* architecture. The input layer (I_X), where X could be the left or right channel image is shown in gray. The *Convolutional* layers C_n are shown in violet for the (a) encoder, in blue for the (b) bottleneck and in red for the (c) decoder. The *Max Pooling* layers (M_1, M_2, M_3) are shown in yellow. The *upsampling* layers (U_1, U_2, U_3) are shown in orange. The variable F corresponds to the number of filter and is equal to 32. The two two skip connections $C_1 \oplus C_9$ and $C_2 \oplus C_8$ are shown with dotted arrows.

4.1 *Mono depth CNN* architecture

The input and target layers of the *Mono depth CNN* architecture has in input grayscale images of 384×192 pixel size. The mono encoder architecture (**Figure 4.3** Box a) is inspired by a simplified version of VGG-16 (Simonyan and Zisserman 2015) where, to preserve the spatial information, fully connected layers are replaced by convolutional layers on the bottleneck. For each convolutional layer in **Figure 4.3** - Box (1) the number of filters F is 32 with a kernel size of 3×3 ($F = 32, k_s = (3 \times 3) \forall C_n$). According to Godard et al. (Godard et al. 2017, 2019), Exponential Linear Unit (ELU) is chosen as the activation function for all the convolutional layers from C_0 to C_{10} . The encoder f_{enc} is structured as 5 convolutional layers with a kernel size of 3×3 and with an increasing filter size F from 32 to 256. The f_{enc} contains 3 down sampling layers (M_1, M_2, M_3) with pool size $ps = (2 \times 2)$, stride 2 and zero padding (see also yellow blocks in **Figure 4.3** - Box (a)). The down sampling strategy adopted is *Max pooling* because it is effective in providing informative down sampled feature maps (Ketkar and Santana 2017). In the mono CNN bottleneck, according to the blue blocks in **Figure** - Box (b), the number of filters is both increased to 64 ($F \times 2$), 128 ($F \times 4$) and 256 ($F \times 8$) and decreased inversely. The decoder f_{dec} mirrors the encoder structure, containing the same number of convolutional layers of f_{enc} and by replacing

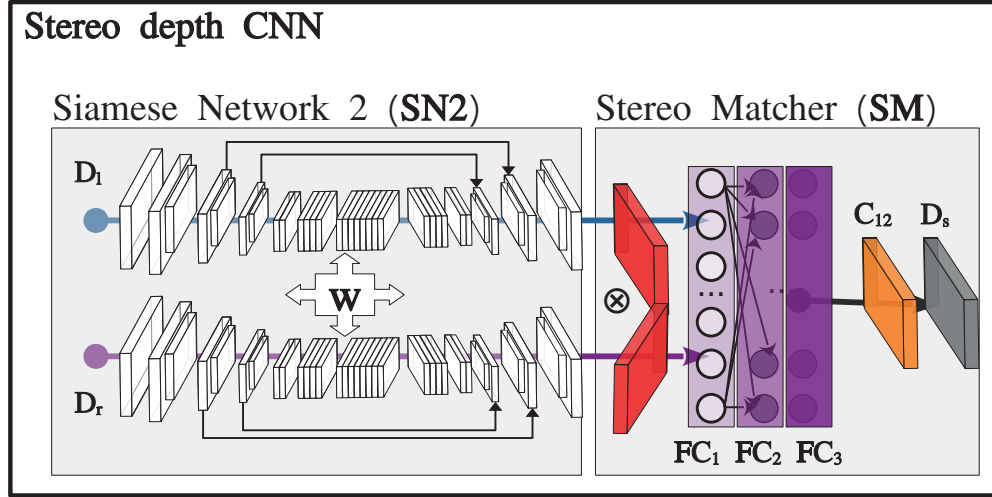


Fig. 4.2 *Stereo depth CNN* is designed with two encoder/decoder networks in a siamese configuration (**SN1**) which are connected with a stereo matching module (**SM**). The encoder/decoder network has the shape of the *Mono depth CNN* described in Figure 4.1. Instead, the **SM** is an additional architectural module composed of fully connected layers and a convolutional layer C_{12} at the end (in orange). The *Stereo depth CNN* takes in input two depth maps (D_l) and (D_r) for left and right channel respectively. The left branch C_{10}^l and right branch C_{10}^r concatenation $\otimes$ is shown in red. The fully connected layers (FC_1, FC_2, FC_3) are indicated in violet. The output depthmap is indicated in gray as D_s .

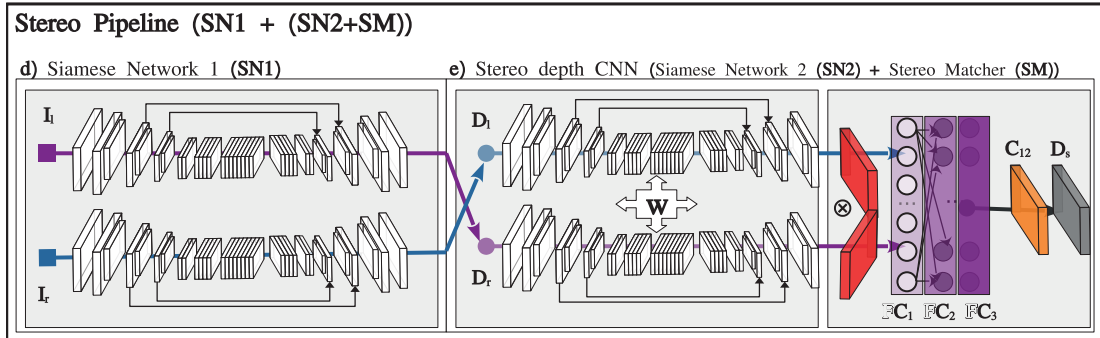


Fig. 4.3 In Figure 4.3 *Stereo depth pipeline* is shown. In particular, two *Mono depth CNNs* are both disposed in a siamese configuration (**SN1**) and stacked on the *Stereo depth CNN* (**SN2 + SM** - see also Figure 4.2). In this way the *Stereo depth pipeline* takes the form of **SN1 + (SN2 + SM)**. At the prediction time the *Stereo depth pipeline* takes in input (I_l) and (I_r) for left and right channel providing in output (D_s).

down sampling layers with up sampling layers. The *upsampling* layers (U_1, U_2, U_3) have an upsampling factor $uf = (2 \times 2)$ and transform spatially the features maps with *proximal* interpolation (see also orange blocks in **Figure 4.3** - Box (a)). At the network tail, f_{dec} outputs a single channel response map which is fed to a Sigmoid activation on C_{11} to provide pixel-wise depth predictions. In forward, *exponential linear unit* (ELU) neurons activation represent the output the features of $C_n \in f_{enc}, f_{dec}$ with n

from 1 to 10. In backward, *ELU* balances the weight updates by *bias shift* fixing neuron activation on their average values (Clevert et al. 2016, Trottier et al. 2017). Moreover, between f_{dec} and f_{enc} , as in ResNet-like networks, vanishing gradient problems are prevented by positioning two skip connections *via addition* (He et al. 2016): $C_1 \oplus C_9$ and $C_2 \oplus C_8$. The position of the skip connections seems to be very relevant for improving or worsening network convergence. The relative layers size and further details are provided in **Figure 4.3 (a-b-c)**. The output layer of this model is equal to input size and provides a mono depth estimation (D_1 or D_r) which depends on the input (further details are shown in the next sections).

4.2 Stereo depth CNN architecture and Stereo depth pipeline configurations

According to similar Siamese networks (Ye et al. 2017, Xu et al. 2019, Luo et al. 2019), the first Box of *Stereo depth pipeline* of Figure 4.3, is composed of two *Mono depth CNNs* (see also Section 4.1) which are trained separately and arranged in parallel (**SN1** in **Figure 4.3-(d)**). The second configured block (**SN2+SM**) is inspired by the *Accurate Architecture* of Žbontar and LeCun (2016) with some additional modifications. In particular it consists of a Siamese network (**SN2**) followed by a stereo matching module (**SM**) with spatial mapping and transformation in the pixel domain of $[0, 1]$ (see **Figure 4.3-(e)**). In this case, as shown in the same Figure, the two encoders/decoders are engineered for weight sharing (**W**). Moreover, their output layers (C_{10}^l and C_{10}^r) are arranged in a concatenation layer $C_{10}^l \otimes C_{10}^r$ (red blocks in **Figure 4.3- Box (e)**). The concatenation layer is flattened by three consecutive fully connected layers **FC**₁, **FC**₂ and **FC**₃ of 200 neurons, respectively (violet blocks in **Figure 4.3- Box (e)**). The stereo matched information is reconstructed from **FC**₃ to a single filter C_{12} of kernel size 3×3 with sigmoid activation. Thus, as it is described in Section 4.4, this model reconstructs depth classified spatial information into a 2D depth map representation by learning combined nonlinear function transformations from both paired stereo features and mono depth estimates.

4.3 Unsupervised loss functions

4.3.1 Mono depth CNN loss

My models are characterized by learning the transformation of the features by representing stereo displacement fields in a mono or stereo configuration. This is induced by considering one of the stereo channels as the input and the other as the target (Additional information in Appendix A.3). In **SN1**, the unsupervised losses of the *Mono depth CNN* is mainly focused on representing the reconstruction error. Instead, with the

Stereo depth CNN, an advanced *depth consistency loss* is designed to learn the stereo matching cost function (between $\mathbf{D}_l$, $\mathbf{D}_r$) and a more detailed edge reconstruction loss. The different losses are parametrized with respect to their predictions $\hat{y}$ and targets y . For example, in **SN1**, the left-branch encoder/decoder $\hat{y}$ is $\mathbf{D}_l$ and its relative target y is I_r . While, on the right-branch, one $\hat{y}$ is D_r and its relative target y is I_l . In detail, $\hat{y}$ and y are collected in tensors per N batches of a specific batch size. According to similar works (Zhao et al. 2016), the *Mono depth CNN* loss (ℓ_{MD}) is defined in Equation 4.1 as a linear combination of ℓ_{dS} (Eq. 4.2) and ℓ_{L1} (Eq. 4.3) with $\alpha = 0.9$ and $\beta = 1.0 - \alpha$ and $\alpha + \beta = 1.0$:

$$\ell_{MD}(y, \hat{y}) = \alpha \times \ell_{dS} + \beta \times \ell_{L1}. \quad (4.1)$$

In the encoders/decoder, the combined loss based on dissimilarity **SSIM** (dS) and **L1** norm is averaged per single batch. In particular, dS is parametrized with $k_1 = 0.01$, $k_2 = 0.03$ and a sliding 3×3 Gaussian kernel of $\sigma = 1.5$. For each i -th batch, the pixel intensities of the gray scale image range is not necessarily fixed at $[0, 1]$ for each image but may change. Thus, accordingly to SSIM definition (Abadi et al. 2016), this dynamic range of intensities is computed for each i -th batch as follows: $dr_i = \max(y_i, \hat{y}_i) - \min(y_i, \hat{y}_i)$. The dissimilarity dS loss (ℓ_{dS}) is defined in Equation 4.2:

$$\ell_{dS}(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^N \frac{1 - SSIM(y_i, \hat{y}_i)}{2}. \quad (4.2)$$

Accordingly, the L1 loss (ℓ_{L1}) is:

$$\ell_{L1}(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^N |y_i - \hat{y}_i|. \quad (4.3)$$

4.3.2 Stereo depth CNN loss

In the *Stereo depth CNN*, a three-losses weighted combination is designed. In addition to ℓ_{dS} and ℓ_{L1} , a third regularization loss is introduced $\ell_{L1\nabla}$ that penalizes the mean absolute first-order gradient. This regularization finds recent applications both for depth estimation (Alhashim and Wonka 2018, Laga et al. 2020) and for super-resolution problems (Lu and Chen 2019) with its adaptation to *Sobel edge detectors* (Kanopoulos et al. 1988). In particular, it provides more attention on edge reconstruction errors (Alhashim and Wonka 2018). In addition, this type of gradient-based loss is less sensitive to variations in lighting and acquisition conditions than the photometric losses (Laga et al. 2020).

The *Stereo depth CNN* loss (ℓ_{SD}) is given in Equation 4.4:

$$\ell_{SD}(y, \hat{y}) = w_\alpha \times \ell_{dS} + w_\beta \times \ell_{L1} + w_\gamma \times \ell_{L1\nabla}. \quad (4.4)$$

Since this is an unsupervised training model, the best *Stereo depth CNN* configuration of Equation 4.4 parameters is obtained by evaluating dozens different weights combinations on the training set. The best four models were compared on the test set, which is completely independent from training set. The best accuracy is obtained by setting $w_\alpha = 0.1$, $w_\beta = 0.2$ and $w_\gamma = 0.6$. In *Stereo depth CNN*, $\hat{y}$ is the couple (D_s^l, D_s^r) and its relative reference y is D_l or D_r . According to similar works (Godard et al. 2017, Luo et al. 2019, Xu et al. 2019), ℓ_{SD} could be intended as a sort of improved *depth consistency loss*.

Given the vertical and horizontal gradients of y and $\hat{y}$ (respectively (δ_v^y, δ_h^y) and $(\delta_v^{\hat{y}}, \delta_h^{\hat{y}})$) the loss $\ell_{L1\nabla}$ is defined in Equation 4.5:

$$\ell_{L1\nabla}(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^N |\delta_v^{y^i} - \delta_v^{\hat{y}^i}| + |\delta_h^{y^i} - \delta_h^{\hat{y}^i}|. \quad (4.5)$$

4.4 Training configurations

In the first training step of the *Stereo depth pipeline*, the *Mono depth CNN* model of **SN1** is trained separately by switching alternatively the left and right channel of stereo rectified images (I_l, I_r) as inputs or targets. For example, if the *input layer* is I_l then *target layer* is I_r or vice versa. Accordingly, the Mono CNN model generates a mono depth estimation D_l or D_r . In the second step, for **SN2 + SM**, the inputs are the stereo depths (D_l, D_r) previously computed and the target is the left reference image (D_l). Eventually, it is possible to train the network with the right channel as target (see also violet and blue arrows in **Figure 4.3 - Box (b)**). The weights of convolutional and fully connected layers are initialized with standard *Glorot uniform* distribution (Glorot and Bengio 2010). This training set is very redundant due to the high frame rate in which the stereo images are sampled over time. To improve model training, the batches are composed of a random image selection with a fixed specific seed. In addition, it is necessary to choose an optimizer able both to provide a balanced importance to infrequent features and, accordingly, to manage their learning rate. *AdaDelta* is able to adapt learning rates on more recent gradient updates instead of accumulating all past gradients. In other words, *AdaDelta* set up its parameters (like learning rate) with a slower convergence on frequent features by observing also infrequent ones. In my case a version of *AdaDelta* is used with an initial learning rate for the *Mono depth CNN* equal to $1 * 10^{-2}$, while for the *Stereo depth CNN* it is equal to $1 * 10^{-3}$. The decay rate for the *Mono depth CNN* and for the *Stereo depth CNN* is $\rho = 0.95$. The two *Mono depth CNNs* are trained separately for 150 epochs with 812, 417 trainable parameters (see also

Section 4.5). The two *Mono depth CNN* differently saturates. Thus, network overfitting is avoided by regularizing the training phase with early stop at the 136th epoch for left-to-right **SN1** branch and at the 138th epoch for the right-to-left **SN1** branch (see also Figure ??). The batch size is equal to 30. Furthermore, the *Stereo depth CNN* is trained for 60 epochs, with early stopping at the 30-th epoch (see also Figure ??). The *Stereo depth CNN* has 814,914 trainable parameters with a batch size of 30 ($BS=30$) (see also Section 4.5). During loss optimization, different types of regularizers can be placed at different architecture levels both to avoid bias and to regularize weights. In my case, for the *Stereo depth CNN*, the loss is regularized at the level of the fully connected layers FC_1, FC_2, FC_3 . In particular, the regularizer is a function LI with the hyperparameter $\lambda = 0.001$.

4.5 *Mono depth CNN* and *Stereo depth CNN* implementation details

In Table 4.1, some model specifications for *Mono depth CNN* are described. While, model specifications *Stereo depth CNN* are described in Table 4.2. Note that, in Table 4.2, the two mono encoder/decoders BED_1 and BED_2 share some weights. For this reason, according to the architecture in Section 4.2, in $*$ ¹ - Table 4.2 the number of parameters is not doubled. The Output Shape represents four parameters in this specific positional order [BS, H, W, F]. BS is the batch size and for both models it is 30. Tensors height and width are indicated respectively with H and W. While with F is indicated the number of filters.

For completeness, *Mono depth CNN* and *Stereo depth CNN* implementations are provided in the following code snippets. The models are written in *Keras backend TensorFlow 2.2.0 on Python 3.8.6*.

```
''' mono depth cnn - encoder/decoder '''
def create_mono_encoder/decoder(input_shape, k1):
    input = Input(shape=input_shape)
    encoded.t = Conv2D(k1, (3, 3), activation='elu', padding='same')(input)
    encoded = MaxPooling2D((2, 2), padding='same')(encoded.t)
    encoded.t = Conv2D(k1, (3, 3), activation='elu', padding='same')(encoded)
    encoded = MaxPooling2D((2, 2), padding='same')(encoded.t)
    encoded.s = Conv2D(k1, (3, 3), activation='elu', padding='same')(encoded)
    encoded = MaxPooling2D((2, 2), padding='same')(encoded.s)
    encoded_skip = Conv2D(k1*2, (3, 3), activation='elu', padding='same')(encoded)
    encoded = Conv2D(k1*4, (3, 3), activation='elu', padding='same')(encoded_skip)
    decoded = Conv2D(k1*8, (3, 3), activation='elu', padding='same')(encoded)
    decoded = Conv2D(k1*4, (3, 3), activation='elu', padding='same')(decoded)
    decoded = Conv2D(k1*2, (3, 3), activation='elu', padding='same')(decoded)
    decoded = UpSampling2D((2, 2), interpolation='nearest')(decoded)
    decoded = Conv2D(k1, (3, 3), activation='elu', padding='same')(decoded)
    decoded = tf.keras.layers.add([decoded, encoded.s])
    decoded = UpSampling2D((2, 2), interpolation='nearest')(decoded)
    decoded = Conv2D(k1, (3, 3), activation='elu', padding='same')(decoded)
    decoded = tf.keras.layers.add([decoded, encoded.t])
    decoded = UpSampling2D((2, 2), interpolation='nearest')(decoded)
    decoded = Conv2D(k1, (3, 3), activation='elu', padding='same')(decoded)
    decoded = Conv2D(1, (3, 3), activation='sigmoid', padding='same')(decoded)
    return Model(input, decoded)

''' Stereo depth CNN - Siamese Network 2 + Stereo Matching '''
def create_siamese_stereo_matcher(input_shape, k1=32, d1=200):
    input_a = Input(shape=input_shape)
    input_b = Input(shape=input_shape)
```

```

processed_a = create_mono_encoder/decoder(input_shape , k1)
processed_b = create_mono_encoder/decoder(input_shape , k1)
concat_proc = Concatenate(axis=-1)([processed_a , processed_b])
concat_proc = Dense(d1, activation='elu' ,
kernel_regularizer=tf.keras.regularizers.l1(0.01),
bias_regularizer=tf.keras.regularizers.l1(0.01))(concat_proc)
concat_proc = Dense(d1, activation='elu' ,
kernel_regularizer=tf.keras.regularizers.l1(0.01),
bias_regularizer=tf.keras.regularizers.l1(0.01))(concat_proc)
concat_proc = Dense(d1, activation='elu' , kernel_regularizer=tf.keras.regularizers.l1(0.01),
bias_regularizer=tf.keras.regularizers.l1(0.01))(concat_proc)
concat_proc = Conv2D(1, (3, 3), activation='sigmoid' , padding='same')(concat_proc)
return Model([input_a , input_b] , [concat_proc])

```

4.6 Training time performances

For each *Mono depth CNN*, the training time is of ≈ 33 hours on NVIDIA GeForce GTX 1080. While, *Stereo depth CNN* is trained with a cluster based computation on MARCONI-100 suited for distributed strategies on 4 x NVIDIA Volta V100 GPUs. In

Table 4.1 . In Table 4.1, *Mono depth CNN* parameters, connections and layer shapes are shown.

Layer (type)	Output Shape	Param	Connected to
input_1 (InputLayer)	[(BS, 192, 384, 1)	0	
conv2d (Conv2D)	(BS, 192, 384, 32)	320	input_1
max_pooling2d (MaxPooling2D)	(BS, 96, 192, 32)	0	conv2d
conv2d_1 (Conv2D)	(BS, 96, 192, 32)	9248	max_pooling2d
max_pooling2d_1 (MaxPooling2D)	(BS, 48, 96, 32)	0	conv2d_1
conv2d_2 (Conv2D)	(BS, 48, 96, 32)	9248	max_pooling2d_1
max_pooling2d_2 (MaxPooling2D)	(BS, 24, 48, 32)	0	conv2d_2
conv2d_3 (Conv2D)	(BS, 24, 48, 64)	18496	max_pooling2d_2
conv2d_4 (Conv2D)	(BS, 24, 48, 128)	73856	conv2d_3
conv2d_5 (Conv2D)	(BS, 24, 48, 256)	295168	conv2d_4
conv2d_6 (Conv2D)	(BS, 24, 48, 128)	295040	conv2d_5
conv2d_7 (Conv2D)	(BS, 24, 48, 64)	73792	conv2d_6
up_sampling2d (UpSampling2D)	(BS, 48, 96, 64)	0	conv2d_7
conv2d_8 (Conv2D)	(BS, 48, 96, 32)	18464	up_sampling2d
add (Add)	(BS, 48, 96, 32)	0	conv2d_8 + conv2d_2
up_sampling2d_1 (UpSampling2D)	(BS, 96, 192, 32)	0	add
conv2d_9 (Conv2D)	(BS, 96, 192, 32)	9248	up_sampling2d_1
add_1 (Add)	(BS, 96, 192, 32)	0	conv2d_9 + conv2d_1
up_sampling2d_2 (UpSampling2D)	(BS, 192, 384, 32)	0	add_1
conv2d_10 (Conv2D)	(BS, 192, 384, 32)	9248	up_sampling2d_2
conv2d_11 (Conv2D)	(BS, 192, 384, 1)	289	conv2d_10
Total params: 812,417			
Trainable params: 812,417			
Non-trainable params: 0			

Table 4.2 . In Table 4.2, *Stereo depth CNN* parameters, connections and layer shapes are shown.

Layer (type)	Output Shape	Param	Connected to
input_1 (InputLayer BED_1)	[(BS, 192, 384, 1)	0	
input_2 (InputLayer BED_2)	[(BS, 192, 384, 1)	0	
model (Model)	(BS, 192, 384, 1)	$812417 *^1$	input_1 AND input_2
concatenate (Concatenate)	(BS, 192, 384, 2)	0	BED_1 AND BED_2
dense (Dense)	(BS, 192, 384, 32)	96	concatenate
dense_1 (Dense)	(BS, 192, 384, 32)	1056	dense
dense_2 (Dense)	(BS, 192, 384, 32)	1056	dense_1
conv2d_12 (Conv2D)	(BS, 192, 384, 1)	289	dense_2
Total params: 814,914			
Trainable params: 814,914			
Non-trainable params: 0			

this case the training time reaches a total of ≈ 4 hours per model.

4.7 Prediction time performances

On a NVIDIA GeForce RTX 2060, the prediction time of *Mono depth CNN* is in average of $20ms$ per frame ($50fps$). *Stereo depth CNN* prediction speed is of $\approx 30ms$ per frame ($\approx 33fps$). *Stereo depth pipeline* requires, in average, $55ms$ per frame ($\approx 18fps$). Finally, the computation time of fuzzy processing is $\approx 100ms$ per frame ($\approx 10fps$) on 384×192 image size. My system could be considered on mid-budget GPUs as close-to-real-time.

CHAPTER 5

Depth map corrections with fuzzy post processing.

The CF_{12} fuzzy post-processing algorithm (Bardozzo et al. 2020) is applied to the outputs of *Stereo depth pipeline* (D_s) to get in output its binary mask ($M_k(D_s)$). In particular, it is used both to foreground/background pixel classification (proximal/distal features) and to improve depth predictions by recognizing specular reflections and other artifacts. CF_{12} -algorithm, based on the homonymous aggregation function, is the most accurate between the four fuzzy binarization methods introduced in Bardozzo et al. (2020). Even if I will not enter in deep theoretical details, according to Dimuro et al. (2020), CF_{12} stands for a group fuzzy integral generalizations based on two functions $F1$ and $F2$ which respect some important fuzzy constraints (see also Appendix B). As we have already described in the Introduction 1, with the advent of neural networks and for the application domain described, it is not advisable to introduce in a surgical workflow additional methods that are computationally slow (even if they are working well from the theoretical point of view). In effect, if on one hand, fuzzy integrals are used to avoid uncertainties in binarization often better than other methods (Bardozzo et al. 2020, Barreto and Coelho 2018, El Baf et al. 2008), on the other hand, the main disadvantage of fuzzy methods relies in the allocation of significant computational resources with respect to some fuzzy constraints. For example, it is required an additional allocation of computational resources specific for problem sorting to respect the monotonicity property (see Appendix B.1). Moreover, the most part of fuzzy algorithms are suited to work on CPUs, while in this thesis I will show a practical way to speed-up these applications on GPUs.

The main intuition that led me to develop a GPU-oriented fuzzy algorithm comes from thinking about some theoretical fuzzy rules and looking closely at the cascade construction of a non-fuzzy integral image through an approach based on *dynamic programming algorithms*. In fact, (i) it is possible to observe that a non-fuzzy integral image is constructed in a constant sorting time by exploiting the well-known summed-area table algorithm (SAT). Therefore, the second intuition was (ii) to find a method for calculating a fuzzy integral image by leveraging the power of *dynamic programming approaches* and of the summed-area table algorithm. Once this strategy was set up, (iii) the third intuition was to exploit the fuzzy integral image in the same way the well-

known algorithm of [Bradley and Roth \(2007\)](#) exploits the non-fuzzy integral image for adaptive image binarization. In Section 5.1, these initial intuitions will be deepened and developed for a modern and fast applicative strategy to compute a fuzzy integral image. The latter is used to binarize the depth map with an adaptive algorithm and is described in Section 5.2¹. The GPU optimization of CF_{12} -algorithm is described in Section 5.3. Once the depth map binarization is computed, the algorithm for CNN estimated depth map post processing is described in Section 5.4.

5.1 Computation of the fuzzy integral image CF_{12}

According to the pipeline described in Figure 5.1, the computation of non-fuzzy integral image S and of the fuzzy integral image $F_{CF_{12}}$ are described in Sections 5.1.1 and 5.1.2.

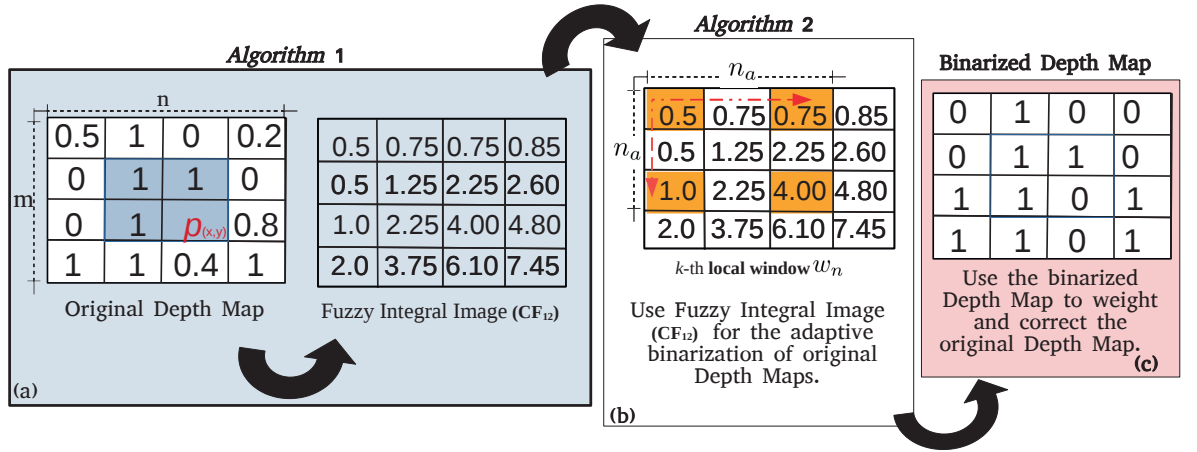


Fig. 5.1 Overview of the complete pipeline from the original depth map to the binarized depth map by going through the Algorithm 1 (Box (a)) and then Algorithm 2 (Box (b)). In Figure 5.1 Box (b), the k -th local search window w_n used for the locally adaptive thresholding is shown. It is important to underline that, as it is described in Algorithm 2, only the 4 values in the orange rectangles are used for binarization. These 4 values are not necessarily adjacent like in the j -th operative window of Figure 5.2. The dashed red arrows show the local window sliding directions, from up to down, from left to right. The local window has a fixed size of $n_a * n_a$. The binarized depth map in output will be used for original depth map post processing as it is described in Section 5.4.

5.1.1 Computation of non-fuzzy integral image S with SAT

Let $n, m \in \mathbb{N}$, $[n] = \{1, \dots, n\}$, $[m] = \{1, \dots, m\}$. An depth map D_S consisting of $n \times m$ pixels arranged in n rows and m columns is associated to the matrix

¹Note that the CF_{12} -algorithm is composed of two smaller algorithms: Algorithm 1 which is described in Section 5.1 and Algorithm 2 which is described in Section 5.2

$[p(x, y)]_{(x,y) \in [n] \times [m]}$ assigning the intensity $p_{\mathbf{D}_S}(x, y)$ to each pixel position $(x, y) \in [n] \times [m]$.

The *non-fuzzy integral image* or simply the *integral image* S is the matrix $[S(x, y)]_{(x,y) \in [n] \times [m]}$, defined for any pixel $p_{\mathbf{D}_S}(x, y) \in [n] \times [m]$ by the following formula 5.1:

$$S(x, y) = \sum_{i \leq x} \sum_{j \leq y} p_{\mathbf{D}_S}(i, j). \quad (5.1)$$

The computation of S has time complexity of $O(l * (n * m))$, which is derived by the number l of times that the Equation 5.1 is applied. However, leveraging the summed-area table algorithm (SAT) (Crow 1984), the computation of $S(x, y)$ can be maintained constant and the SAT time complexity remains fixed to $O(n * m)$. The SAT can be developed efficiently computing for each pixel $(x, y) \in [n] \times [m]$ the column-wise prefix-sums and the row-wise prefix-sums (Kasagi et al. 2014), as it is shown in Equation 5.2:

$$\begin{aligned} S(x, y) &= p(x, y) + S(x, y - 1) + \\ &S(x - 1, y) - S(x - 1, y - 1), \end{aligned} \quad (5.2)$$

with convention $S(0, k) = 0$, for each $k = 0, \dots, m$ and $S(l, 0) = 0$, for each $l = 0, \dots, n$.

5.1.2 Algorithm 1: Fuzzy integral image computation with Fuzzy-SAT

Algorithm 1 is used to compute the fuzzy integral images based on CF_{12} Sugeno generalization. There are two main elements to understand in order to follow the development of the Algorithm 1:

- **(I)** The first element to consider is the identification of the j -th operative window used for the computation of the integral image S (see also Section 5.1.1).
- **(II)** The second element to consider is the theoretical definition of the aggregation function CF_{12} used to handle the elements of the j -th operative window.

(I) The j -th operative window is well defined in Figure 5.2 - Box **(a-b)** and its movement along the image matrix is well explained by the Equation 5.2. **(II)** The theoretical aspects of CF_{12} are defined in Appendix B.1.5.

Once the integral image S is computed, for each pixel (x, y) of the j -th operative window it is possible to compute the *fuzzy integral image* $F_{CF_{12}}$, as follows:

$$\begin{aligned} F_{CF_{12}}(x, y) &= \\ CF_{12}(S(x, y), S(x, y - 1), S(x - 1, y), S(x - 1, y - 1)), \end{aligned} \quad (5.3)$$

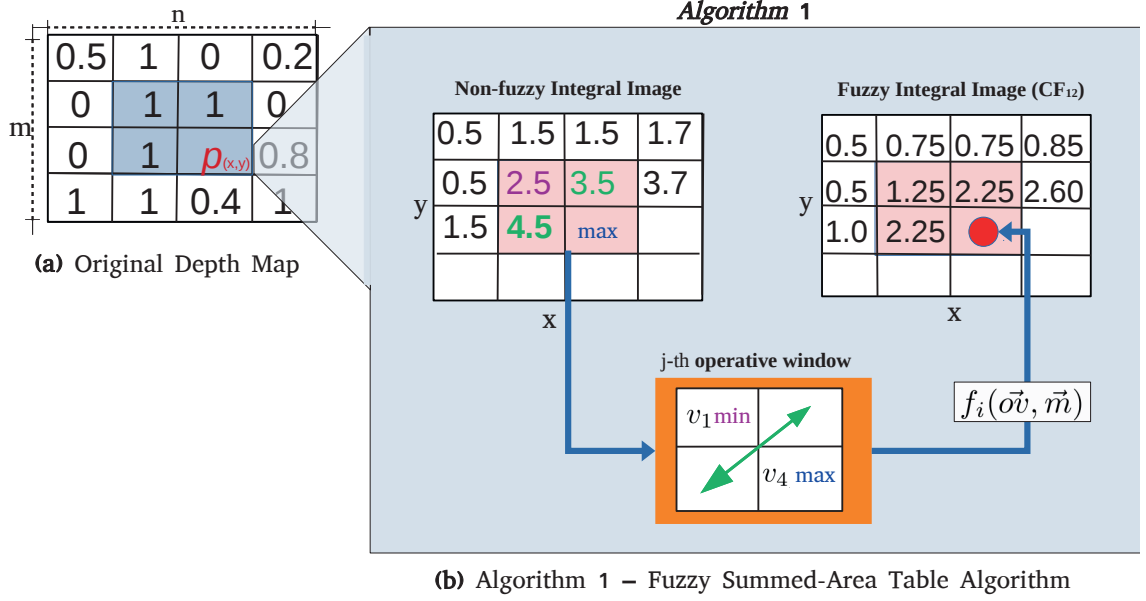


Fig. 5.2 Overview of Algorithm 1. From the original Depth Map in input (Box (a)) the algorithm computes both the non-fuzzy integral image S and dynamically transforms it into a fuzzy integral image ($F_{CF_{12}}$) (see Section 5.1.1 and 5.1). The blue square in Box (a) represents current value of $p(x, y) \in I$ (in red) and its neighboring pixels defined in j -th operative window. For each pixel $p(x, y)$ and for every j -th operative window (yellow box), the 4 values in S are mapped with the associated fuzzy measures through $F_{CF_{12}}(x, y) = f_i(\vec{ov}, \vec{m})$. The output of fuzzy-based integral functional computation is saved in the fuzzy integral image $F_{CF_{12}}$. This computation is described in formula 5.3. In the j -th operative window, the fixed values of min v_1 and max v_4 are represented in violet and blue. For the decision of v_2 and v_3 , the green arrow indicates the P_{swap} action as described in Procedure 5.5.

First observation: *The symmetric property of the fuzzy measure ensures that the number of elements in the calculation is fixed to the maximum 4 elements of j -th operative window.*

In my case, it is possible to setup a fuzzy set E as the set of 4 elements of the j -th operative window computed on S and used in Algorithm 1:

$$E := \{S(x, y), S(x, y - 1), S(x - 1, y), S(x - 1, y - 1)\} \quad (5.4)$$

According, to the definition of the *uniform fuzzy measure* provided in Appendix B.1.2, μ_{uni} is the ratio between one of the possible subsets of $|E|$ and the number of elements of E ($|E| = 4$).

Accordingly to the definition in Appendix B.1, the set function $\mu: 2^{[n]} \rightarrow [0, 1]$ is a function whose domain consists of all possible subsets of N (Beliakov et al. 2020). In my case, the reader could be induced to think that the maximum four elements of the j -th operative window become $2^4 = 16$ elements; $\mu(\cdot), \mu(\{1\}), \mu(\{2\}), \mu(\{3\}), \dots, \mu(\{1, 2\}), \mu(\{1, 3\}), \mu(\{2, 3\}) \dots \mu(\{1, 2, 3\}) \dots \mu(\{1, 2, 3, 4\})$.

However, the first advantage of using a fuzzy measure like μ_{uni} is that it is symmetric, thus for any $A, B \in [n]$ then $|A| = |B|$ in cardinality. This implies that the measure $\mu(A)$ is equal to $\mu(B)$. In other words, for example, it is possible that $\mu(\{S(x, y), S(x, y - 1)\}) = \mu(\{S(x, y), S(x - 1, y)\})$ because in both the cases, the sets cardinality is 2 and their fuzzy measure value is necessarily 0.5.

Second observation: *In this problem, the natural ordering of the elements generated by the summed-area table algorithm allows to respect the fuzzy monotonicity constraint without additional computational sorting times. At most, there is only one more swap operation.*

Observing the j -th operative window, the maximum value v_4 is always the element $S(x, y)$ present on the right lower corner, while, the minimum value v_1 is $S(x - 1, y - 1)$ element ((see Figure 5.2 - Box (b)) - *min* in violet, *max* in blue). This derives from the *dynamic programming* "cascade" of operations that are carried out in the construction of S with the summed-area table algorithm. Therefore an element deriving from the previous elements on the matrix (always greater than 0) will always be larger than the previous ones. Thus, in order to complete the sorting, it is eventually necessary only to swap $s_1 = S(x, y - 1)$ and $s_2 = S(x - 1, y)$ by the following swap operation P_{swap} ((see Figure 5.2 - Box (b)) - double green arrow):

$$P_{swap}(s_1, s_2) = \begin{cases} v_2 = s_1, v_3 = s_2 & \text{if } s_1 < s_2 \\ v_2 = s_2, v_3 = s_1, & \text{otherwise} \end{cases} \quad (5.5)$$

In such a way it is obtained a one dimensional array of sorted values at a cost of only one comparison: $\vec{ov} = [v_0, v_1, v_2, v_3, v_4]$, where by convention $v_0 = 0$.

Moreover, since $E_{(i)} = \{(i), \dots (4)\}$ is the subset of indices of the $4 - i + 1$ greatest component of $\vec{ov}$, for the uniform fuzzy measure μ_{uni} (see formula B.1) then $\mu_{uni}(E_{(i)}) = \frac{4 - i + 1}{4}$. This means that dealing with always a maximum of 4 values, the vector of fuzzy measures will be always of this form:

$$\vec{m} = [\mu_{uni}(E_{(1)}), \mu_{uni}(E_{(2)}), \mu_{uni}(E_{(3)}), \mu_{uni}(E_{(4)})] = [1, 0.75, 0.50, 0.25].$$

In conclusion, as shown in Algorithm 1, the procedure takes advantage of the natural ordering of the four elements aggregated in (5.3), obtaining a vector of ordered values $\vec{ov}$ and an associated costless vector of fuzzy measures $\vec{m}$.

Given that, only the 4 operative window corners are considered at a time ($OCF_{12}(4)$) by leaving the computational complexity polynomial in time (as it is for the original SAT): $O(n \times m) + O_{f_i}(4) + \dots = O(n \times m)$.

Note that, in the pseudocode of Algorithm 1, a bridge function $f_i(\vec{ov}, \vec{m})$ for each pixel $p(x, y)$ is defined, in order to map each fuzzy integral-based functional computation of the aggregation function CF_{12} on the two vectors: $\vec{ov}$ and $\vec{m}$ for the j -th operative

window.

Algorithm 1 : Computation of $F_{CF_{12}}$.

Require: Gray-scale depth map image I with intensities in $[0, 1]$.

function COMPUTEFUZZYSUMMEDAREATABLE(I)

$n, m \leftarrow \dim(I)$

▷ Dimension of I

$S \leftarrow$ allocate a zero-matrix with size (n, m)

$F_{CF_{12}} \leftarrow$ allocate a zero-matrix with size (n, m)

$\vec{m}_1 \leftarrow [1, 0.75, 0.50, 0.25]$; $\vec{m}_2 \leftarrow [1, 0.50]$

for $r \leftarrow 1$ to n **do**

for $c \leftarrow 1$ to m **do**

if $r \neq 1 \wedge c \neq 1$ **then**

$v_1 \leftarrow S[r-1, c-1]$; $s_1 \leftarrow S[r, c-1]$; $s_2 \leftarrow S[r-1, c]$

$S[r, c] \leftarrow I[r, c] + s_1 + s_2 - v_1$; $v_4 \leftarrow S[r, c]$

$v_2, v_3 \leftarrow P_{swap}(s_1, s_2)$ - Procedure 5.5

$v_0 \leftarrow 0$; $\vec{ov} \leftarrow [v_0, v_1, v_2, v_3, v_4]$

$F_{CF_{12}}[r, c] \leftarrow f_i(\vec{ov}, m_1)$

else if $r \neq 1$ **then**

$v_1 \leftarrow S[r-1, c]$; $S[r, c] \leftarrow I[r, c] + v_1$

$v_4 \leftarrow S[r, c]$; $\vec{ov} \leftarrow [v_0, v_1, v_4]$

$F_{CF_{12}}[r, c] \leftarrow f_i(\vec{ov}, m_2)$

else if $c \neq 1$ **then**

$v_1 \leftarrow S[r, c-1]$; $S[r, c] \leftarrow I[r, c] + v_1$

$v_4 \leftarrow S[r, c]$; $\vec{ov} \leftarrow [v_0, v_1, v_4]$

$F_{CF_{12}}[r, c] \leftarrow f_i(\vec{ov}, m_2)$

else

$S[r, c] \leftarrow I[r, c]$; $F_{CF_{12}}[r, c] \leftarrow S[r, c]$

end if

end for

end for

return $F_{CF_{12}}$

end function

5.2 Adaptive binarization based on fuzzy integral images

As it is shown in Figure 5.2 and in the pipeline of Figure 5.2, Algorithm 1 outputs the fuzzy integral image $F_{CF_{12}}$. Then, the latter is given in input to Algorithm 2 for binarization. For what is concerning the binarization, $F_{CF_{12}}$ will be leveraged as the non-fuzzy integral image S is exploited in *Bradley* algorithm. However, in Algorithm 2 a modified version of the *Bradley* algorithm is presented according to these constraints: **(1)** As it is described in Chapter 4, the depth map D_s is defined in a binary space and represented in this Thesis with values between $[0, 1]$. **(2)** The choice of sliding windows is not static but engineered on dimensions that are not necessarily odd or even. This is the second step in the pipeline of Figure 5.2.

In Algorithm 2, the square determined by the upper left corner (x_1, y_1) and the lower right corner (x_2, y_2) is $[(x_1, y_1), (x_2, y_2)]$. Once $F_{CF_{12}}$ is obtained, the sum of the pixel intensities in $[(x_1, y_1), (x_2, y_2)]$ are denoted by $p_s(x_1, y_1, x_2, y_2)$ and it is given by Equation 5.6:

$$p_s(x_1, y_1, x_2, y_2) = F_{CF_{12}}(x_2, y_2) - F_{CF_{12}}(x_2, y_1) - F_{CF_{12}}(x_1, y_2) + F_{CF_{12}}(x_1, y_1), \quad (5.6)$$

where $1 \leq x_1 \leq x_2 \leq m, 1 \leq y_1 \leq y_2 \leq n$. Thus, the average value of pixel intensities in $[(x_1, y_1), (x_2, y_2)]$, is given by Equation 5.7:

$$p_a(x_1, y_1, x_2, y_2) = \frac{p_s(x_1, y_1, x_2, y_2)}{(x_2 - x_1) \times (y_2 - y_1)} \quad (5.7)$$

with $1 \leq x_1 \leq x_2 \leq m, 1 \leq y_1 \leq y_2 \leq n$. In the second part of the Algorithm 2, given the current pixel $p(x, y)$, every pixel intensity of the original depth map $\mathbf{D}_s$ is compared pixel-by-pixel to the average value of the pixel intensities in the k -th local window around $p(x, y)$ (See also Figure 5.1 - Box **(b)**). For a given size $2s \times 2s$ of the local window, the Algorithm 2 iteratively binarizes original depth map and provides the binary values $M_k(x, y)$ for each pixel $p(x, y)$, as described in the following Formula:

$$M_k(x, y) = \begin{cases} 1 & \text{if } p(x, y) \leq p_a(x_1, y_1, x_2, y_2) \times t, \\ 0 & \text{otherwise,} \end{cases} \quad (5.8)$$

where $(x_1, y_1, x_2, y_2) = (x - s, y - s, x + s, y + s)$. Note that the k -th local window needs to be changed, if it is not within the borders of original image, and it is accounted in the Algorithm implementation I provided in Appendix B.2. Note also that it is considered only a percentage of p_a controlled by sensitivity value t , which is defined in interval $[0, 1]$. Furthermore, the size and the coordinates of the sliding nearest neighbor's pixel local window w_n is set with the dimensional parameter n_a . The latter is computed through 2 empirical parameters: a_1 and a_2 . As it is described above, the w_n is used to locally binarize the central pixels. Thus, w_n is sized and positioned following the procedure described in Algorithm 2. The kernel area of w_n is equal to n_a^2 . The dimensional parameter n_a is computed as follows:

$$n_a = \left\lfloor \frac{\min(n, m)}{a_1 \times a_2} \right\rfloor. \quad (5.9)$$

and it is based on the (n, m) dimensions of I .

Algorithm 2 : Depth-map binarization based on $F_{CF_{12}}$

Require: Gray-scale depth map image I with intensities in $[0, 1]$.

Require: The fuzzy integral image $F_{CF_{12}}$.

Require: The parameters a_1 and a_2 .

Require: The sensitivity parameter θ

function BINARIZEDDEPTHMAP($I, F_{CF_{12}}, a_1, a_2, t$)

$n, m \leftarrow \dim(I)$

 ▷ Dimension of I

$I_b \leftarrow$ allocate a zero-matrix with size (n, m)

$n_a \leftarrow$ Defined in Formula 5.9 with a_1 and a_2

for $r \leftarrow 1$ to n **do**

for $c \leftarrow 1$ to m **do**

$y_0 \leftarrow \max(r - n_a, 0)$

 ▷ Set the w_n

$y_1 \leftarrow \min(r + n_a, r)$

$x_0 \leftarrow \max(c - n_a, 0)$

$x_1 \leftarrow \max(c + n_a, c)$

$p_{area} \leftarrow (y_1 - y_0) * (x_1 - x_0)$

$p_s \leftarrow F_{CF_{12}}[y_1, x_1] - F_{CF_{12}}[y_0, x_1] - F_{CF_{12}}[y_1, x_0] + F_{CF_{12}}[y_0, x_0]$

$p_a \leftarrow \frac{p_s}{p_{area}}$

if $I[r, c] \leq p_a \times (1 - \theta)$ **then**

$I_b[r, c] \leftarrow 1$

else

$I_b[r, c] \leftarrow 0$

end if

end for

end for

return I_b

end function

Table 5.1 I provide for each test set with ground truth described in Chapter 3 - Section 3.2.2, the CF_{12} -hyper-parameters table used for *Stereo depth CNN* post processing ($f(\mathbf{D}_S)$). The fuzzy kernel size is determined by Equation 5.9 on parameters $\mathbf{a}_1$ and $\mathbf{a}_2$ (see Section 5.2), in this case $\mathbf{a}_1 = \mathbf{a}_2$. The sensitivity parameter is t . Further, π is the percentage of depth intensity reduction and ρ is the Gaussian kernel filter size.

	π	ρ	$\mathbf{a}_1 = \mathbf{a}_2$	t
<i>Test Set B - Video 1</i>	0.2	3	4	0.70
<i>Test Set B - Video 2</i>	0.4	5	6	0.58
<i>Test Set C - Video 1</i>	0.2	2	5	0.60
<i>Test Set D - Video 1</i>	0.1	2	4	0.74

5.3 GPU optimization of CF_{12} -algorithm (Algorithm 1 and Algorithm 2)

It has recently been possible to harness the power of the GPU to optimize this type of algorithms which are generally structured for CPU calculations. By the way, without going down to a CUDA programming level. For example, regarding Algorithm 1 and Algorithm 2 and on this specific input size for CNNs, I was able to fine-tune the calculation going from the order of seconds to the order of milliseconds. In particular I used Python's numba package. I provide the code snippet in Appendix B.2 with the data structures pre-allocated on the GPU so that the tensor calculations can be applied.

5.4 Fuzzy post-processing of depth map estimations

The depth image $\mathbf{D}_S$ (in gray scale) inputted to CF_{12} -algorithm is binarized, classifying the proximal features (and also their artifacts as specular reflections) with the proximal values equal to 0 and the distal features as 1. A visual pipeline is shown in Figure 5.3. In detail, the *inverse binary* mask M_k is the output of CF_{12} binarization (see also Figure 5.3 (d)). In the same **Figure**, the black dotted square underlines the area of action where the estimated $\mathbf{D}_S$ (in (c)) is transformed in $f(\mathbf{D}_S)$ (as shown in Figure 5.3 (e)). In detail, according to last step in the pipeline of Figure 5.2, on the original depth map $\mathbf{D}_S$, the following transformation is applied:

$$f(I_{[x,y]}) = \begin{cases} \mathbf{D}_{S[x,y]} = \mathbf{D}_{S[x,y]} - \mathbf{D}_{S[x,y]} \times \pi & \text{if } M_{k[x,y]} == 0 \\ \mathbf{D}_{S[x,y]} = \mathbf{D}_{S[x,y]} & \text{otherwise} \end{cases} \quad (5.10)$$

The effect of Equation 5.10 is to reduce the value of I by a factor π for pixels classified as proximal features (those in foreground, when $M_{k[x,y]} = 0$). Otherwise, the procedure preserves the original pixel values which are classified by CF_{12} as background features. Then, a Gaussian filter with ρ value is applied to reduce derivative artifacts. The parameters π and ρ are reported in Table 5.1. As it is possible to observe

in Algorithm 2 the binarization procedure is parametric. In detecting specular reflections, a deeper analysis reveals that, on each test set and at different processing levels, different parameters can produce better results. The optimal parameters search is described in (Bardozzo et al. 2020) even if it is inapplicable for a real-time application. For this reason, all hyper-parameters are fixed for each test set according to the Table 5.1. The 3D visual result of fuzzy post processing application is shown in Figure 5.3 - Box (f).

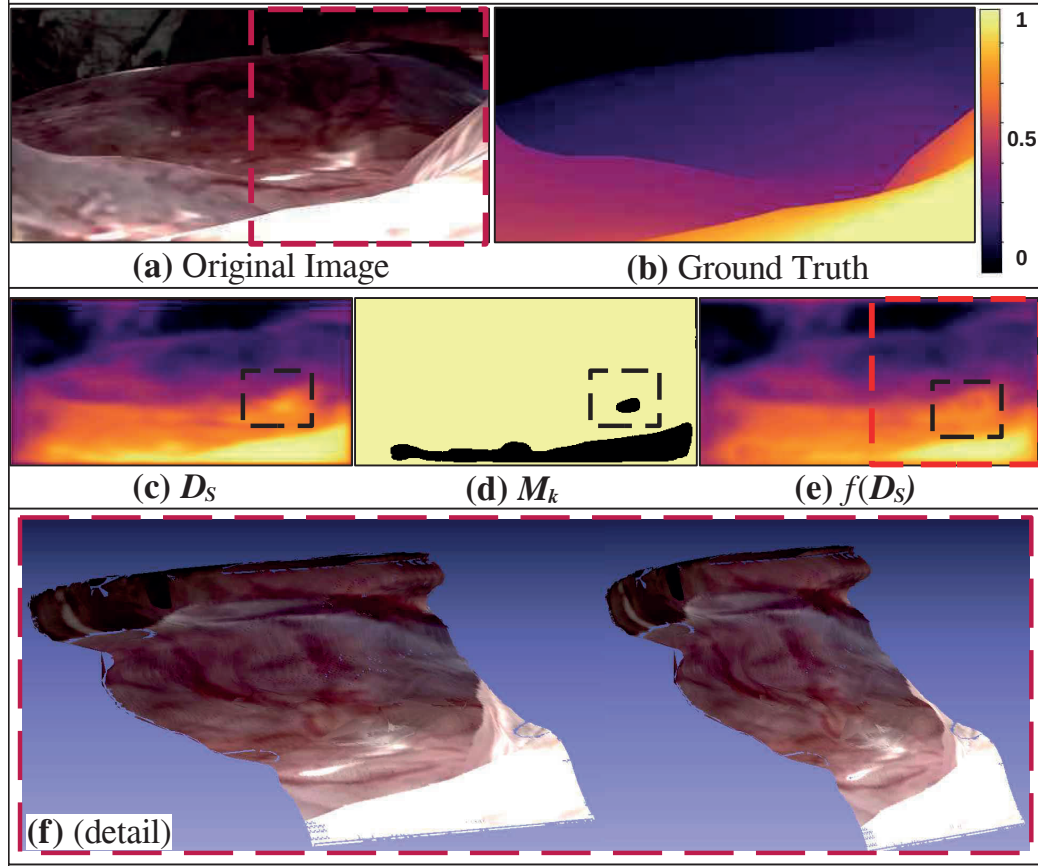


Fig. 5.3 In Figure 5.3 - Box (a), the left reference image I^l is shown. In Box (b), its ground truth is reported. The *Stereo depth pipeline* depth estimation D_S is shown in Box (c). In Figure (d), the dashed black square shows a set of features which are classified with value 0 on the binary mask M_k . In Figure (c), the same set of features is indicated with the dashed black square. While, in $f(D_S)$, according to the procedure described in Section 5.4, these features are lowered in intensity (Dashed black square of Box e). In Box (f), the 3D Mesh of the window depicted in the red dashed square of Box (a)(e) is shown with two different rotation degrees. In detail, this Mesh is obtained from the depth estimation of Box (e). While, the pixel colours of Box (a) are remapped into the Mesh in Box (f). In this case fuzzy-post processing is able to completely remove specular reflections.

CHAPTER 6

Results and Applications

In this Chapter, I describe all the tests performed on the datasets described in Chapter 3 with the models developed in Chapters 4 and 5. Section 6.1 describes the experimental setup and the types of measures used to evaluate depth estimations. In Section 6.2, comparisons on test set *Test Set A* are performed. In Section 6.3, my network mono and stereo predictions are compared with other traditional and deep learning models on *in Silico* and *in Vivo* test sets (*in Silico* with GT and *in Vivo* with GT, *Test Set B* and *Test Set C* respectively). Moreover, a within and between experiment comparisons on the simulated test set *Test Set D* for several conditions are performed (see also Table 6.4 and Table 6.5). Finally, in Section 6.5 visual evaluations are provided through 3D depth representations. Some important considerations and conclusions are in Chapter 7. Most of the results obtained in my research can be found in the submitted paper (Bardozzo et al. 2021).

6.1 Experimental setup

6.1.0.1 Evaluations on datasets without ground truth

During the training, my models alternately cross the right channel from the left and vice versa to exploit, in the predictive phase, more information acquired during training. Then, in the test steps, when dealing with stereo datasets, the *Mono depth CNN* takes as input the original right channel image (I_r) and outputs an estimates of the left image (D_l). For this reason, the depth map reconstruction quality is measured by considering the difference between D_l and the original left image I_l . In the stereo pipeline, *Steredo depth pipeline* takes as input a rectified image pair (I_l, I_r) and returns D_s . Since in these test sets there is not a ground truth, as in other works (Ye et al. 2017, Xu et al. 2019, Luo et al. 2019), the depth map accuracy is evaluated as the difference between D_s and the original left image I_l . The reconstruction quality is evaluated with *Structural similarity index (SSIM)* and *Peak signal-to-noise ratio (PSNR)*. These are different and complementary image reconstruction metrics (Ye et al. 2017, Xu et al. 2019, Luo et al. 2019). There are several free parameters for *SSIM*. In these comparisons, *SSIM*

is used with a Gaussian kernel (with $\sigma = 1.5$), $k_1 = 0.01$ and $k_2 = 0.03$ by default. The Gaussian kernel size k_w is a free parameter that it is used to asses image similarity at different scales (Wang et al. 2004). For what is concerning *Test Set A*, the value of k_w is not specified in previous works (Ye et al. 2017, Xu et al. 2019, Luo et al. 2019), thus my comparisons are performed on a range of values with $k_w := 3 \times 3$, 2×2 and 1×1 . On the other hand, *PSNR* has no hyper-parameters and we use the implementation from (MATLAB 2010).

6.1.1 Evaluations on datasets with ground truth

On the test sets with ground truth (*Test Set B*, *Test Set C* and *Test Set D*), the estimated left channel depth maps are compared to the left channel ground truth depth maps. I measured reconstruction error directly with Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). RMSE is more sensitive to outliers than MAE because it penalizes the square error (Allan et al. 2021). Therefore, both metrics are instructive for comparisons. For completeness, image reconstruction error is measured with a non-Gaussian SSIM using a 7×7 block kernel size (Van der Walt et al. 2014).

6.1.2 Results reported previously

Some results, which are collected from their respective papers, as indicated by asterisks, are shown in Tables 6.1 and 6.2. In the case of Ye et al. (2017) where public code is available, we tested the trained models they provide on the online repository ¹ (date of download *May, 2020*). These trained networks show improved results with respect to their previously published ones because they add skip connections with a combined $L1 - SSIM$ loss (*Basic M₁* and *Siamese M₂* for Basic and Siamese) and with a multi-scale training approach (*Basic M₃* and *Siamese M₄* for Mono and Stereo, respectively).

6.1.2.1 Comparison

My depth predictions and GTs are defined in the range $[0, 1]$. M_1 , M_2 , M_3 , M_4 are given in output into the pixel range $[0, 255]$. To make it comparable the latters are linearly rescaled to $[0, 1]$. According to what described in Chapter 5.4, fuzzy post-processing is applied to D_S and indicated for stereo depths with $f(D_S)$. All the deep learning models, which are compared (including mine), are trained with unsupervised/semi-supervised deep learning approaches using the same training set (see also Chapter 3.1). Finally, both traditional and learning models are tested on the same test set (*Test Set A*).

¹Ye et al. (2017): https://github.com/magnumye/disp_est_net

Table 6.1 Table 6.1 shows comparisons via *SSIM* and *PSNR* on *Test Set A* for different traditional and deep learning models (see Chapter 3 - Section 3.2.2).

	SSIM	
Traditional models		
Geiger et al. (2010)	0.473 ± 0.079 ²	
Yamaguchi et al. (2014)	0.547 ± 0.092 ²	
Deep learning models	Mono	Stereo
Ye et al. (2017) ⁴ M_1 - M_2	0.584 ± 0.030	0.673 ± 0.040
Ye et al. (2017) ⁴ M_3 - M_4	0.672 ± 0.030	0.673 ± 0.040
Xu et al. (2019)	0.541 ± 0.071	0.835 ± 0.052
Mine ⁵ with $k_w = 3 \times 3$	0.770 ± 0.035	0.825 ± 0.031
Mine ⁵ with $k_w = 2 \times 2$	0.823 ± 0.030	0.837 ± 0.037
Mine ⁵ with $k_w = 1 \times 1$	0.929 ± 0.018	0.938 ± 0.024

	PSNR	
Deep learning models	Mono	Stereo
Garg et al. (2016)	11.003 ± 7.860 ³	
Ye et al. (2017) ⁴ M_1 - M_2	8.216 ± 1.042	12.058 ± 1.779
Ye et al. (2017) ⁴ M_3 - M_4	11.887 ± 1.062	12.474 ± 1.778
Xu et al. (2019) Xu et al. (2019)	7.765 ± 1.369	14.496 ± 1.968
Mine ⁵	12.335 ± 1.469	13.357 ± 1.247

6.2 Evaluation without Ground Truth

My models are compared to traditional models of Geiger et al. (2010), Yamaguchi et al. (2014) and unsupervised deep learning models of Garg et al. (2016), Ye et al. (2017) and Xu et al. (2019) by means of *SSIM* and *PSNR*. The results on *Test Set A* are shown in Table 6.1 ^{2 3 4 5}. For *SSIM*, my *Mono depth CNN* achieves the best results mainly considering the Gaussian kernel size equal to $k_w = (w \times w)$, $w = [1, 2, 3]$ with a very small standard deviation. Similarly, *PSNR* is also better for our mono model. For *SSIM*, the *Stereo depth pipeline* achieves the best results with a $k_w = (w \times w)$, $w = [1, 2]$. In mono and stereo configurations, I have chosen several k_w values both to better evaluate the robustness and sensitivity of our results and because this parameter is not indicated in the other works (as explained in section 6.1.0.1). Xu et al. (2019) show better values only for *PSNR* on the stereo/Siamese model, even if the standard

² These results are reported in Ye et al. (2017)

³ These results are reported in Luo et al. (2019)

⁴ Note that, as described in Section 6.2, M_1 , M_2 , M_3 and M_4 are four improved models (Basic and Siamese) which are provided by Ye et al. (2017) on its GitHub repository.

⁵ My **Mono** is referred to the prediction D_1 of *Mono depth CNN* (right-to-left branch of **SN1**), while my **Stereo** is the prediction D_S of *Stereo depth pipeline* (**SN1**+(**SN2**+**SM**))

deviation is greater than mine.

6.3 Evaluation with Ground Truths

6.3.1 Comparisons on Test Set B

In this Section, my models are compared to traditional models of Wang et al. (2018), Stoyanov et al. (2010) and with the unsupervised deep learning models of Godard et al. (2017), Ye et al. (2017) (which are described in Section 6.1.2) and with Luo et al. (2019) semi-supervised model. In this first comparison, two *in Silico* heart-beating videos (see *Test Set B* in Chapter 3 - Section 3.2.2) depth estimations are measured with respect to ground truths with MAE and $RMSE$. The results of the comparison are shown in Table 6.2. On this *Test Set B* - Video 1, my *Mono depth CNN* model perform significantly better than many traditional models, the semi-supervised model of Luo et al. (2019) and the mono Siamese single and multi-scale models of Ye et al. (2017). In Stereo comparisons the single-scale model of Ye et al. (2017) seems to perform slightly better even if with a higher standard deviation. Similarly, stereo single-scale model and Mono multi-scale models in *Phantom heart Video 2* seem to perform better than mine. However, they lack in robustness showing a higher standard deviation. This evidence is also shown in previous comparisons on Table 6.1. A visual comparison is provided in Figure 6.1: although the predictions of Ye et al. (2017) appear to be more performant, with an accurate visual analysis is possible to observe that they are very noisy and incorrect. Moreover, the fuzzy mask, obtained in post processing, appear to be highly representative with respect to specular reflections and the veins in foreground. In conclusion, even if this dataset is widely used in Literature, as it is also discussed in Chapter 3, the comparisons between the models are slightly weak for depth estimations cause of the inaccurate and incomplete ground truths. This is also the point of view of Allan et al. (2021) and Antal (2016, 2017), which achieved a $RMSE$ of ≈ 4.84 with a supervised deep neural network model trained on 3D coordinates for depth estimation.

6.3.2 Comparisons on Test Set C

Table 6.3 shows my performances compared to Ye et al. (2017) models when using the *Test Set C* (see also Section 3.2.2). In this case, the fuzzy processing of the depth map appears to have slightly lower reconstruction errors, while the $SSIM$ is kept optimal both in the configuration with and without fuzzy processing. This is also in accord to the visual comparison provided in Figure 6.2. A robotic arm is overlapped to the internal surface disturbing in a certain sense the scene. Always in Figure 6.2, it appears that the models of Ye et al. (2017) better recognize the proximity of the robotic arm even if their depth maps delocalize and deform the arm not respecting the positioning of the

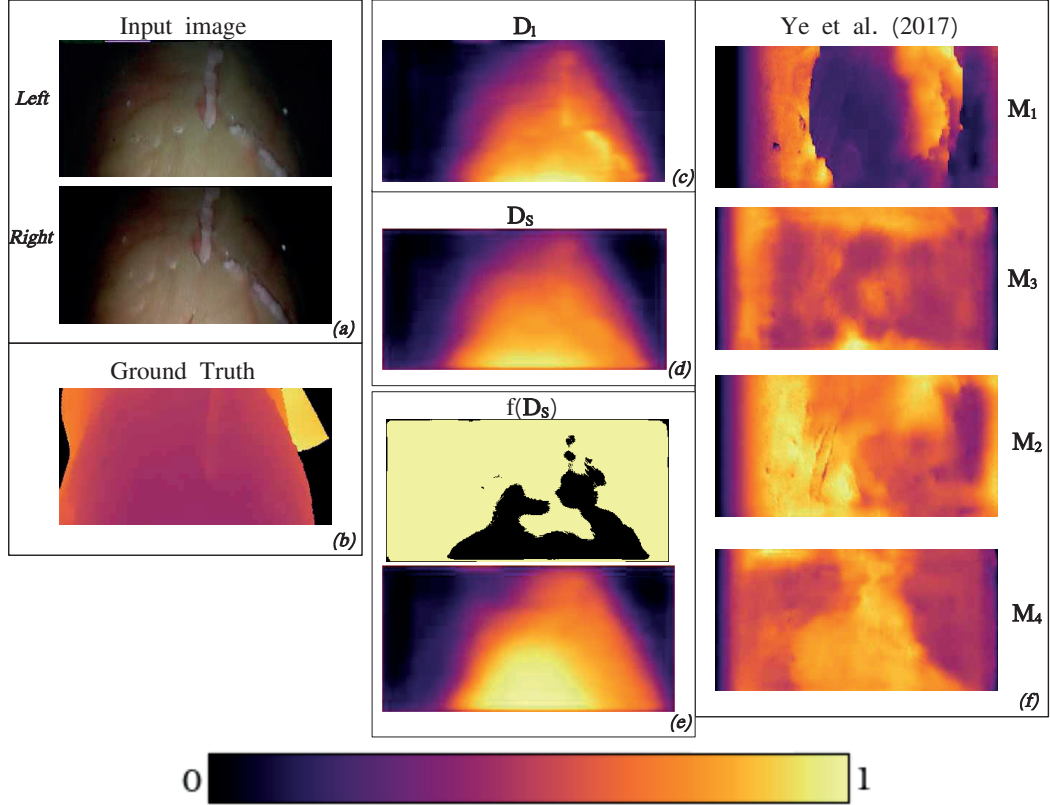


Fig. 6.1 Visual comparisons on a sample of *Test set B*: In this Figure a sample of depth estimation for my models are in comparison with Ye et al. (2017)’s M_1 , M_2 , M_3 and M_4 . Box (a): *Test Set B* left/right input sample. Box (b): the ground truth of the input sample. Box (c): the depth estimation in output from *Mono depth CNN*. Box (d): the depth map in output from *Stereo depth pipeline*. Box (e): the depth map estimated by the *Stereo depth pipeline* is post-processed. Box (f): Mono and stereo depth estimation in output from the models of Ye et al. (2017).

Table 6.2 Comparisons on *Test Set B* (See also Chapter 3 - Section 3.2.2). The best results both for mono and stereo models are in bold.

TestSet2 - Phantom heart Video 1		
Model	MAE	RMSE
Godard et al. ³	2.39 ± 0.62	2.99 ± 0.61
Wang et al. ³	2.16 ± 0.65	—
Stoyanov et al. ³	2.36 ± 0.92	3.88 ± 0.87
Luo et al. Luo et al. (2019)	1.84 ± 0.40	2.69 ± 0.58
M ₁ (Mono) Ye et al. (2017)	0.57 ± 0.14	0.67 ± 0.17
M ₃ (Stereo) Ye et al. (2017)	0.64 ± 0.21	0.74 ± 0.20
M ₂ (Mono) Ye et al. (2017)	0.65 ± 0.27	0.78 ± 0.25
M ₄ (Stereo) Ye et al. (2017)	0.72 ± 0.25	0.80 ± 0.25
D ₁	0.55 ± 0.08	0.67 ± 0.11
D _S	0.66 ± 0.12	0.78 ± 0.14
$f(D_S)$	0.67 ± 0.12	0.78 ± 0.14

TestSet2 - Phantom heart Video 2		
Model	MAE	RMSE
Godard et al. ³	1.79 ± 0.40	2.65 ± 0.28
Wang et al. ³	2.14 ± 0.83	—
Stoyanov et al. ³	3.20 ± 1.15	4.85 ± 1.82
Luo et al. Luo et al. (2019)	1.49 ± 0.41	1.90 ± 0.38
M ₁ (Mono) Ye et al. (2017)	0.61 ± 0.09	0.71 ± 0.08
M ₃ (Stereo) Ye et al. (2017)	0.33 ± 0.06	0.43 ± 0.06
M ₂ (Mono) Ye et al. (2017)	0.48 ± 0.10	0.62 ± 0.08
M ₄ (Stereo) Ye et al. (2017)	0.39 ± 0.13	0.49 ± 0.11
D ₁	0.54 ± 0.04	0.69 ± 0.04
D _S	0.60 ± 0.01	0.71 ± 0.02
$f(D_S)$	0.58 ± 0.03	0.67 ± 0.05

elements. A possible solution to test tool positioning on depth maps is by using *Mask R-CNN* based surgical tool detection systems Ciaparrone et al. (2020) on the original images and depth maps. In addition, in Ye et al. (2017) stereo models, the dark side bars are not removed / reconstructed thus narrowing the area actually estimated and increasing the reconstruction error. In this dataset, *Stereo depth pipeline* appears to predict more accurate depth maps than my *Mono depth CNN* model. As it is discussed in Chapter 3 - Section 3.2.2, these ground truths are not perfectly accurate because only a few depth levels are represented. For example, in Figure 6.2, the robotic arm appears to be positioned in a separate horizontal plane with respect to the animal surface.

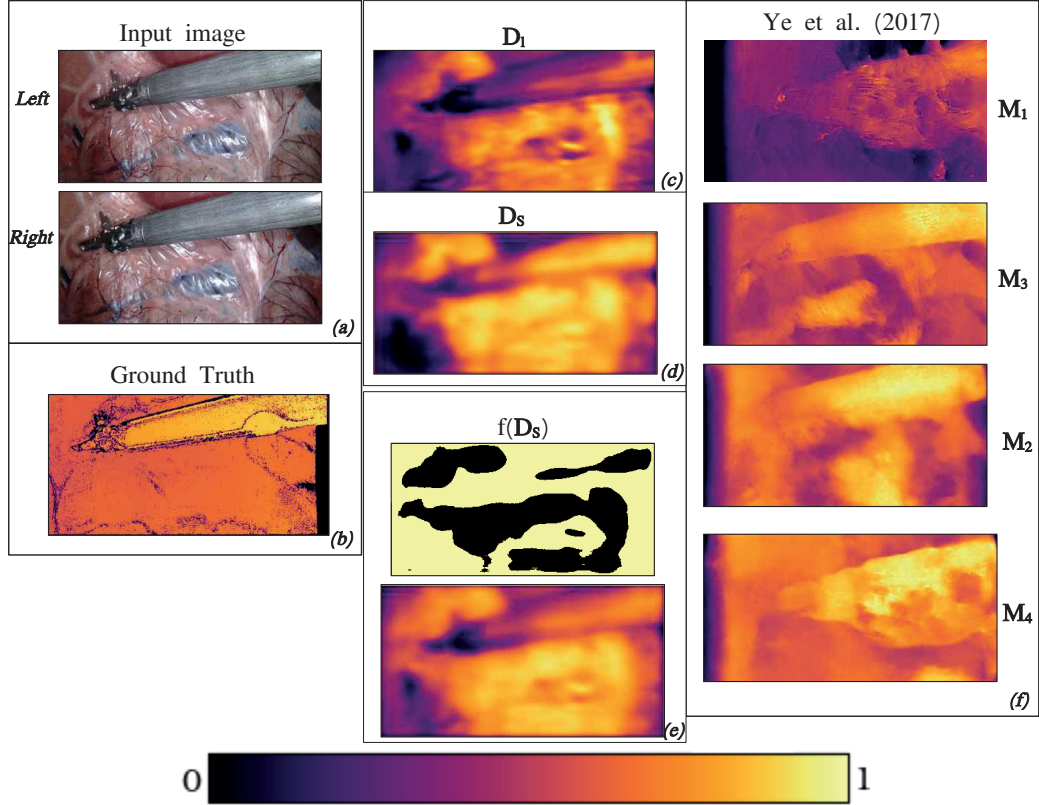


Fig. 6.2 Visual comparisons on a sample of *Test set C*: In this Figure a sample of depth estimation for my models are in comparison with [Ye et al. \(2017\)](#)'s M_1 , M_2 , M_3 and M_4 models. Box (a): *Test Set B* left/right input sample. Box (b): the ground truth of the input sample. Box (c): the depth estimation in output from *Mono depth CNN*. Box (d): the depth map in output from *Stereo depth pipeline*. Box (e): the depth map estimated by the *Stereo depth pipeline* is post-processed. Box (f): Mono and stereo depth estimation in output from the models of [Ye et al. \(2017\)](#).

6.4 Evaluations on the simulated Test Set D

6.4.1 Experiment 1: Change-light-fix-position vs fix-position-change-light

In Experiment 1 exactly 10,000 stereo images of 4 different scenes synchronized on the same position are compared in two different conditions: (i-1) positional variation of endoscope camera with fixed lights (NAVI) and (i-2) positional variations of lights with a fixed camera position (LIGHT). In detail, in experiment (i-1), changing the positioning of the camera while keeping the light fixed results in better performance on *Mono depth CNN* than on Stereo one. While, in experiment (i-2) my models are poorly conditioned by the shadows generated by moving the lights, achieving very robust performances similar to the experiment (i-1). My models outperform the others of [Ye et al. \(2017\)](#). Again the *Mono depth CNN* predictions appear to be slightly more accurate. The results of this experiment are shown in Table 6.4.

6.4.2 Experiment 2: Scene ensemble vs Scene ensemble with smoke

In the second experiment **(ii)**, an ensemble of ≈ 7000 stereo images from 3 different scene typologies is configured with and without smoke perturbations. In particular scene 1 - *NAVI challenge*, the same scene 1 but with lights variations for scene 1 - *LIGHT challenge* and scene 4 of *DEFORM challenge* are selected. The ensemble with and without smoke perturbations is chosen to represent a mix of scenes with different challenging features: presence of blood, specular reflections, smoke generate from electro-cautery, etc. For this reason, the smokeless scene ensemble (SE) can be considered an almost real laparoscopic exploration video. While the ensemble scene with smoke (SE + SMOKE) is meant to be a simulation of a laparoscopic surgery video where electro cauterization is applied. The results obtained by my models and those of [Ye et al. \(2017\)](#) are shown in Table 6.5. My models perform better in terms of SSIM and reconstruction errors than the models without and with smoke perturbations (SE

Table 6.3 Table of comparisons on *Test Set C - MICCAI 2019 EndoVis Grand Challenge* dataset with **SfM**-based GT. The best results both for mono and stereo models are in bold

My models	MAE	RMSE	SSIM
D_1 (Mono)	0.30 ± 0.06	0.36 ± 0.06	0.60 ± 0.06
D_S (Stereo)	0.22 ± 0.05	0.29 ± 0.05	0.61 ± 0.06
$f(D_S)$ (Stereo)	0.21 ± 0.04	0.28 ± 0.05	0.61 ± 0.06
Ye et al. (2017)	MAE	RMSE	SSIM
M_1 (Mono)	0.48 ± 0.09	0.53 ± 0.09	0.55 ± 0.03
M_3 (Stereo)	0.26 ± 0.05	0.31 ± 0.05	0.60 ± 0.05
M_2 (Mono)	0.34 ± 0.07	0.38 ± 0.06	0.59 ± 0.05
M_4 (Stereo)	0.25 ± 0.04	0.30 ± 0.05	0.60 ± 0.05

Table 6.4 Comparisons on *Test Set C* - NAVI and LIGHT challenge between my models and those of Ye et al. (2017). The best results both for mono and stereo models are in bold.

My models		NAVI	LIGHT
D_1 (Mono)	<i>MAE</i>	0.09±0.01	0.09±0.01
	<i>RMSE</i>	0.12±0.01	0.13±0.01
	<i>SSIM</i>	0.84±0.01	0.82±0.02
D_S (Stereo)	<i>MAE</i>	0.16±0.01	0.16±0.02
	<i>RMSE</i>	0.21±0.01	0.21±0.03
	<i>SSIM</i>	0.82±0.01	0.80±0.02
$f(D_S)$ (Stereo)	<i>MAE</i>	0.16±0.01	0.16±0.02
	<i>RMSE</i>	0.20±0.02	0.20±0.03
	<i>SSIM</i>	0.83±0.01	0.81±0.02
<i>Ye et al. (2017)</i>		NAVI	LIGHT
M_1 (Mono)	<i>MAE</i>	0.29±0.01	0.26±0.05
	<i>RMSE</i>	0.37±0.01	0.33±0.05
	<i>SSIM</i>	0.68±0.02	0.67±0.02
M_3 (Stereo)	<i>MAE</i>	0.44±0.02	0.43±0.03
	<i>RMSE</i>	0.47±0.01	0.47±0.03
	<i>SSIM</i>	0.68±0.02	0.67±0.01
M_2 (Mono)	<i>MAE</i>	0.31±0.01	0.33±0.02
	<i>RMSE</i>	0.36±0.01	0.37±0.02
	<i>SSIM</i>	0.72±0.01	0.69±0.01
M_4 (Stereo)	<i>MAE</i>	0.37±0.01	0.36±0.03
	<i>RMSE</i>	0.40±0.01	0.47±0.03
	<i>SSIM</i>	0.70±0.01	0.69±0.01

and **SE+SMOKE** respectively). Moreover, my models remain stable to smoke perturbations while maintaining higher accuracy levels in depth predictions. The desmoking property is reflected also in similar models Vishal et al. (2020), Bolkar et al. (2018), Chen et al. (2019). *StaSiS-Net* Stereo predictions show high accuracy when post processed with the fuzzy strategy described in Section 5.4 with respect *Mono depth CNN*. Furthermore, a single frame example of smoke addition with relative depth estimation is provided in Figure 6.4 with respect to Figure 5.3 - Box (a). An analysis with several levels of smoke is provided in

6.4.3 Experiment 3: Normal condition vs three levels of smoke

In Experiment 2, I already proved the goodness of my models facing smoke perturbations. In this experiment, I changed the tail of the *Stereo depth pipeline* to evaluate if the *Stereo depth CNN* trained until the 30-th epoch is effectively more accurate than the network trained until the 60-th epoch. Thus three different levels of smoke are added to three samples of *Test Set D* - scene 4 DEFORM challenge and their *Stereo depth pipeline* predictions are computed. Depth predictions at epoch 30 are indicated with

Table 6.5 My models are compared to those one of [Ye et al. \(2017\)](#) on an ensemble of challenging scenes from *Test Set D* (NAVI,LIGHT,DEFORM - see also Section ??). The scene ensemble (**SE**) is analysed with and without SMOKE perturbations. For mono and stereo analyses the best results are in bold.

My models		SE	SE+SMOKE
D_1 (Mono)	<i>MAE</i>	0.09 ± 0.01	0.12 ± 0.01
	<i>RMSE</i>	0.12 ± 0.01	0.16 ± 0.01
	<i>SSIM</i>	0.84 ± 0.01	0.82 ± 0.01
D_S (Stereo)	<i>MAE</i>	0.16 ± 0.01	0.12 ± 0.01
	<i>RMSE</i>	0.21 ± 0.01	0.16 ± 0.01
	<i>SSIM</i>	0.82 ± 0.01	0.81 ± 0.02
$f(D_S)$ (Stereo)	<i>MAE</i>	0.16 ± 0.01	0.11 ± 0.01
	<i>RMSE</i>	0.20 ± 0.01	0.16 ± 0.01
	<i>SSIM</i>	0.83 ± 0.01	0.84 ± 0.01
<i>Ye et al. (2017)</i>		SE	SE+SMOKE
M_1 (Mono)	<i>MAE</i>	0.31 ± 0.01	0.32 ± 0.03
	<i>RMSE</i>	0.39 ± 0.01	0.39 ± 0.02
	<i>SSIM</i>	0.69 ± 0.01	0.69 ± 0.01
M_3 (Stereo)	<i>MAE</i>	0.37 ± 0.01	0.35 ± 0.01
	<i>RMSE</i>	0.42 ± 0.02	0.40 ± 0.02
	<i>SSIM</i>	0.72 ± 0.01	0.73 ± 0.01
M_2 (Mono)	<i>MAE</i>	0.28 ± 0.01	0.29 ± 0.02
	<i>RMSE</i>	0.33 ± 0.01	0.35 ± 0.02
	<i>SSIM</i>	0.75 ± 0.01	0.73 ± 0.01
M_4 (Stereo)	<i>MAE</i>	0.28 ± 0.01	0.32 ± 0.02
	<i>RMSE</i>	0.32 ± 0.01	0.37 ± 0.02
	<i>SSIM</i>	0.76 ± 0.01	0.74 ± 0.01

D_s^{30} , while depth predictions at epoch 60 are indicated with D_s^{60} . The visual results on Figure 6.3 show that the depth estimations at D_s^{30} provide quite stable depth estimations to small and medium amounts of smoke, while the predictions at D_s^{60} seem to better performing in preserving only the background information, also with high levels of smoke. However, as it is discussed on Chapter 3, my analyses lead to prefer D_s^{30} because on average values turns out to be more reliable. Effectively, observing with attention Figure 6.3, D_s^{30} maintains some details that in D_s^{60} are missed (such for example the dark orange particles in bottom left side). Despite the simple visual analysis, in Table 6.6 the analytical analysis on reconstruction errors and other scores provides a clear explanation of the obtained results with and without fuzzy post processing. It is possible to observe that stereo depth estimation D_s^{60} is worse than the mono one which is the same between the two tables (*Mono depth CNN* is not changed). Moreover, in this case, D_s^{30} is clearly better than $f(D_1)$ and D_s^{60} . In Table 6.6, FD_{score} and ME_{score} are two scores used in *MICCAI EndoVis Grand Challenge 2019* (the lower the best): they are similar to failure detection methods and more stable to outliers.

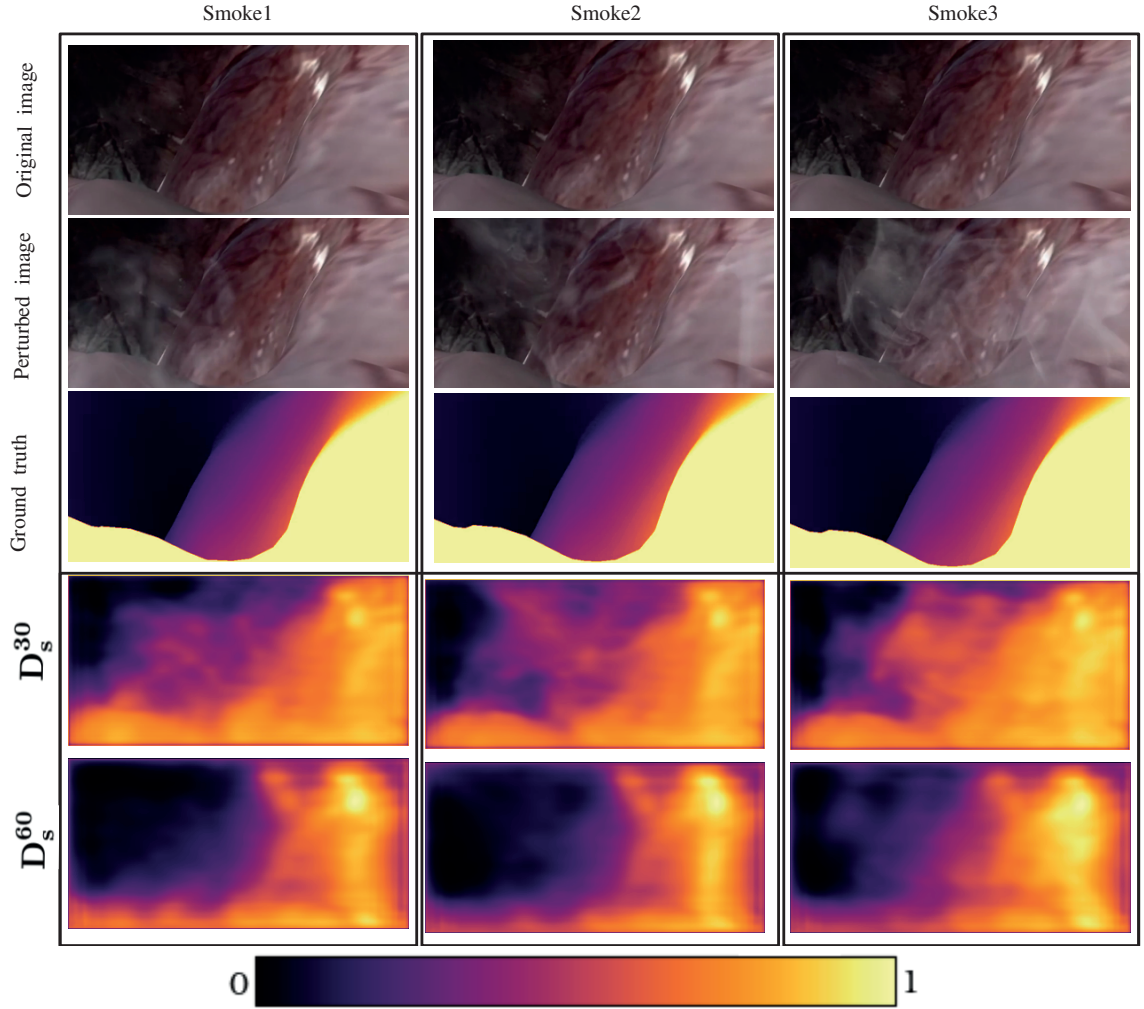


Fig. 6.3 In this Figure three different levels of noise (smoke) are added to three samples of *Test Set D*. *Stereo depth pipeline* predictions are computed with two different trained *Stereo depth CNNs*. In particular, depth predictions are provided with *Stereo depth CNN* trained until epoch 30 (its depths predictions are indicated with D_s^{30}) and another one it is trained until epoch 60 (its depth predictions are indicated with D_s^{60}). The depths are without fuzzy post processing. In Table 6.6 an analytical comparison between depth predictions D_s^{30} and D_s^{60} is provided.

Table 6.6 Table of comparisons between depth estimations D_s^{30} ad D_s^{60} on samples with smoke of Figure 6.3.

Stereo depth pipeline with Stereo depth CNN trained at Epoch 30 (in the tail)

	MAE	FD-score	SSIM	MSE	ME-score	RMSE	PSNR
D_I	0.215	0.0724	0.8439	0.0738	0.2759	0.0738	11.3178
$f(D_I)$	0.1681	0.0144	0.8485	0.0473	0.1983	0.0473	13.2515
D_s^{30}	0.1485	0.0063	0.8649	0.0378	0.1656	0.0378	14.2212
$f(D_s^{30})$	0.1328	0.0013	0.8532	0.0303	0.1321	0.0303	15.1878

Stereo depth pipeline with Stereo depth CNN trained at Epoch 60 (in the tail)

	MAE	FD-score	SSIM	MSE	ME-score	RMSE	PSNR
D_I	0.215	0.0724	0.8439	0.0738	0.2759	0.0738	11.3178
$f(D_I)$	0.1681	0.0144	0.8485	0.0473	0.1983	0.0473	13.2515
D_s^{60}	0.1637	0.0573	0.8507	0.0557	0.2175	0.0557	12.5428
$f(D_s^{60})$	0.1651	0.0668	0.8601	0.0559	0.2205	0.0559	12.5236

6.4.4 Additional samples and comparisons also with fuzzy pre processing.

In this Section, I provide a link to a google drive to download some samples from my dataset [\[here\]](#). In Figure 3.4 left-right channel samples for different scenes of *Test Set D* are provided. In particular 3 key-frames for each challenge typology are provided (see also Chapter 3). For each key-frame, I provided depth estimations, GTs and tables with the associated reconstruction errors and similarity indexes (note that the tables are provided without headers - please refer to the Table 6.6 headers). In addition, depths are provided without fuzzy processing and with two different types of fuzzy processing: fuzzy pre-processing and fuzzy post-processing. In fuzzy pre-processing the images in input to **SN1**-left-to-right *Mono depth CNN* and **SN1**-right-to-left *Mono depth CNN* are pre-processed, while in post-processing the output of *Stereo depth pipeline* is fuzzy processed. Thus, it means that $f(I_l)$ and $f(I_r)$ are fuzzy pre processed, instead, in fuzzy post-processing, $f(D_s)$ is post-processed. Fuzzy pre-processing on original images is less accurate than post processing on depths (See also Chapter 5).

6.5 Visual results

In Figure 5.3 and 6.5, the depth maps are first transformed into point clouds (*pcd*) and then their surfaces are reconstructed. The colors maps are remapped according to the z-position obtained from the depths. According to *Open3D* tools (Zhou et al. 2018), to speed up the 3D surface reconstruction and keep a discrete reconstruction accuracy, the *pcd* are disposed in a 3D voxel grid. Then in each voxel, all the points present are downsampled with their centroid. Next, their normals are estimated. Finally, the ball-pivoting algorithm for surface reconstruction is applied (Bernardini et al. 1999). In detail, Figure 5.3 - Box (e) shows a 3D reconstruction of a depth map detail (red dashed

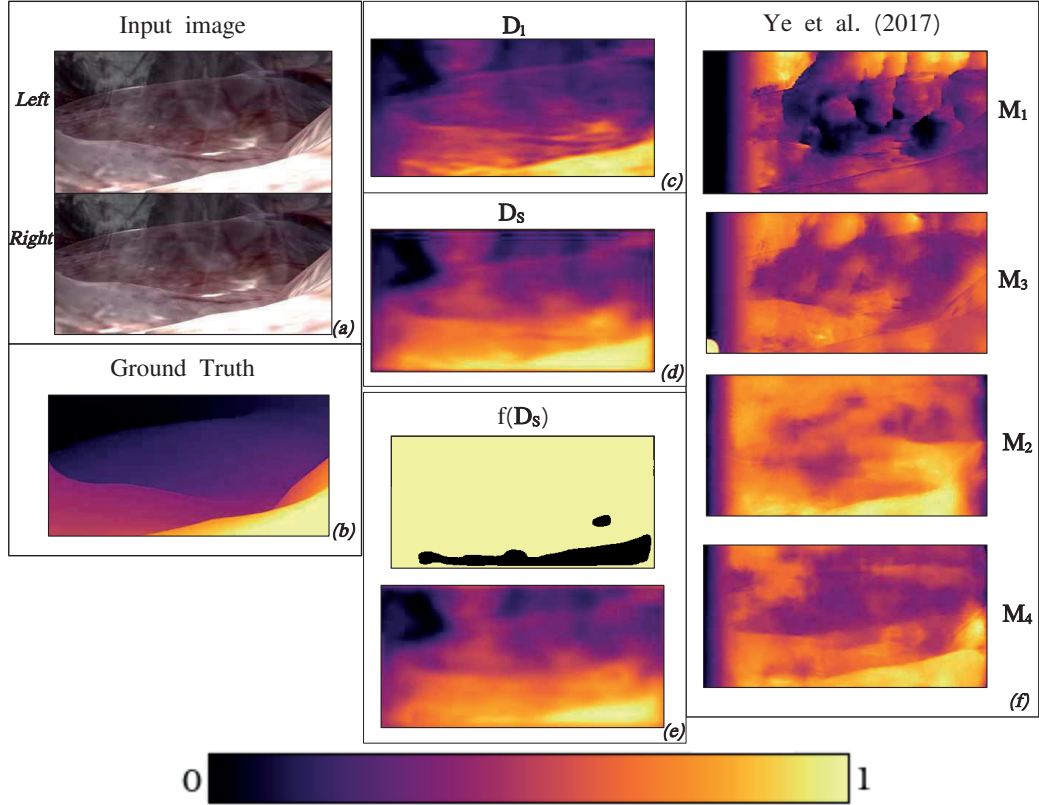


Fig. 6.4 Visual comparisons on a sample of *Test set D* with *SMOKE challenge* perturbations and evident specular reflections: In this Figure a sample of depth estimation for my models are in comparison with [Ye et al. \(2017\)](#)’s M_1 , M_2 , M_3 and M_4 models. Box (a): *Test Set B* left/right input sample. Box (b): the ground truth of the input sample . Box (c): the depth estimation in output from *Mono depth CNN*. Box (d): the depth map in output from *Stereo depth pipeline*. Box (e): the depth map estimated by the *Stereo depth pipeline* is post-processed. Box (f): Mono and stereo depth estimation in output from the models of [Ye et al. \(2017\)](#).

square) improved by the fuzzy processing on D_S . This particular detail shows that the specular reflections are completely removed. On the hand in Figure 6.5 the specular reflections are managed directly by the *Stereo depth pipeline* without the need of post-processing strategies. Moreover, a video showing the potential of 3D reconstruction based on my neural network models can be reached at this link: [YouTube Video](#) (Additional information in Appendix A.2). In this video, *Scene 4* of *Test Set D* and a part of the video of *Test Set A* are shown using point clouds and 3D meshes. In particular, it is possible to visually evaluate sensitivity and robustness of depth estimations to specular reflections, smoke and other types of perturbations, in real-time, both on a simulated video and on a real video without fuzzy post-processing. It is beyond the scope of this thesis to discuss the implementation of the tool shown in the video, but what I would like to emphasize is the functionality in using neural depth estimation models to be concretely applied within the surgical workflow by allowing the surgeon to move the dynamic scenes with a user-friendly interface.

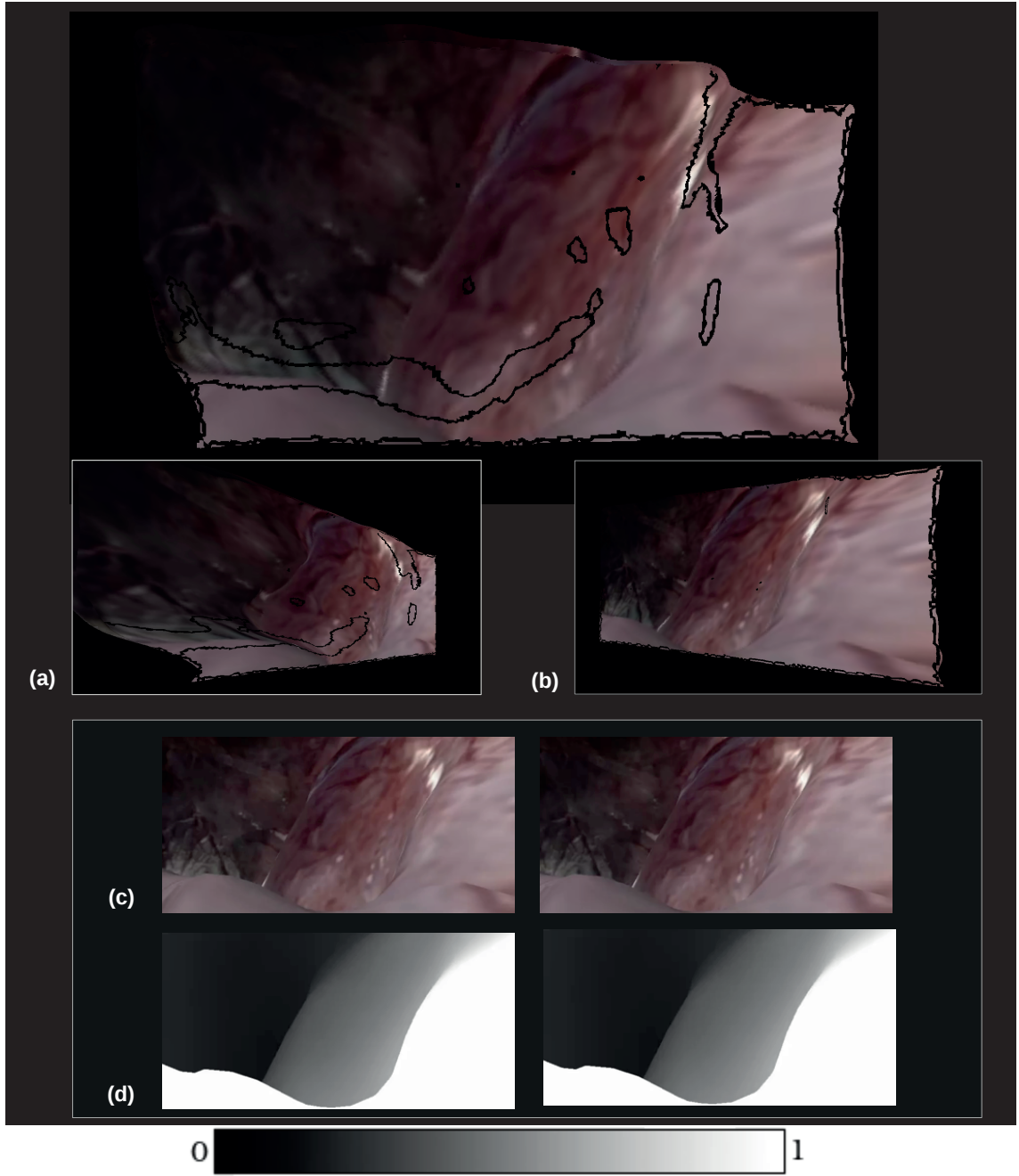


Fig. 6.5 In Figure 6.5, the estimated 3D surface reconstruction of a frame from DEFORM challenge set is provided. The estimated 3D reconstruction is shown with different 45 rotation degrees from right-to-left (a) and left-to-right (b). The original left-right image and their ground truths are in box (c) and (d) respectively. The 3D reconstructed surfaces on a black background were taken directly from the tool I developed. Their brightness and shading are dependent on the 3D Mesh reprojection, so there are no further modifications to the 3D model in post-processing or color adjustments. Moreover, it is not applied fuzzy post-processing on their estimated depths. The pitted holes on the borders and on 3D models are the missing or incorrect triple of points that the tool was unable to reconstruct from the depth map. A finer 3D reconstruction could address these holes by paying for additional computation time.

CHAPTER 7

Conclusions

In this thesis, I designed and implemented stacked and siamese encoders/decoders combined with stereo matching deep learning models for stereo and mono depth estimations in laparoscopic scenes. For the reasons I explained in Chapter 1, this is a challenging problem due to both the scarcity of concrete data on which benchmarking the models, and to the intrinsic nature of the observed scenes. As far as is my knowledge, as well as I described in Chapter 2, there are few state-of-the-art models that show solutions that can be concretely applied in a surgical workflow. Therefore, on one hand, I had to reconstruct a challenging-enough data set to test my models and those of others (see also Chapter 3), on the other hand, I had to rethink the way of designing models based on neural networks to obtain reliable depth estimates that would improve the state-of-the-art and that could be applied in a concrete surgical workflow (see also Chapters 4 and 5). In Chapter 6, the depth estimations were tested in terms of robustness and sensitivity on a large collection of images with ground truths. In the same Chapter, I provided additional comparisons on reconstruction errors considering state-of-the-art deep learning and traditional models.

The depth predictions generated by my models are stable with respect to perturbations induced by change of lights and shadows, specular reflections, moderate presence of smoke, blood, partial occlusions, consistent deformations and spasms on *In Vivo* test sets. Nonetheless, *Mono depth CNN* model and its configuration on the *Stereo depth pipeline* show very high accuracy also on *Simulated* test sets with a very accurate ground truth (*Test set D*). Always on *Test set D*, the additional information derived from stereo pair does not appear to significantly improve depth estimations. On the other hand, stereo information appears to be very important on the two *in Vivo* test sets (*Test set C* in Section 6.3.2 and *Test set B* in Section 6.2). This can be derived from the fact that my models are able to generalize better on *in Vivo* stereo features than on simulated ones, meanwhile on *in Silico* and on *Simulated* tests it seems that my depth predictions are more accurate. This could mean that, for the proposed models, the stereo cameras are fundamental in the unsupervised training phase. In fact, in the prediction phase, the adoption of a stereo or mono predictive model directly depends on the type of considered images. It is true even if the differences between mono and stereo models, in terms

of similarities and reconstruction errors, are very small. As I have described along the Thesis, for some datasets with ground truths, some reconstruction measures are sometimes unreliable due to actually incorrect or ill-defined ground truths. However, the various tests clearly demonstrate the goodness of the models presented and their effectiveness within the surgical workflow and despite the various perturbations. Moreover, I provided real-time and close-to-real-time results on a YouTube video, showing the empirical effectiveness and feasibility of developing tools that can actually be used within a surgical workflow and are essentially based on estimates deriving from neural networks. My PhD Thesis ends with a conclusion that is not a conclusion. Despite all my efforts, I am not convinced that this work is an end point and that there is nothing after it. I think that this work is a valid starting point to redesign and look with different approaches and methods to some aspects that have penalized research in this field so far. For example, I think, that in future works, instead of using fuzzy post-processing, it could be interesting to study its effect in the training phase, also changing the methods for processing. Always, in future works, it should be possible to delve into what concerns the estimation of depth in critical conditions, as well as reshaping the CNN models in a supervised or semi-supervised fashion by using additional 3D information in the training phase. As for the registration, for example, it will be possible to apply a neural network that performs the segmentation and which is aligned with a neural network that performs the depth estimation in order to obtain segmented 3D objects that can be easily aligned to the surgical images (like CT).

REFERENCES

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. et al. (2016), Tensorflow: A system for large-scale machine learning, *in* ‘12th USENIXsymposium on operating systems design and implementation (OSDI 16)’, pp. 265–283.
- Alhashim, I. and Wonka, P. (2018), ‘High quality monocular depth estimation via transfer learning’, *arXiv preprint arXiv:1812.11941* .
- Ali, S., Zhou, F., Bailey, A., Braden, B., East, J. E., Lu, X. and Rittscher, J. (2021), ‘A deep learning framework for quality assessment and restoration in video endoscopy’, *Medical Image Analysis* **68**, 101900.
URL: <http://www.sciencedirect.com/science/article/pii/S1361841520302644>
- Allan, M., Mcleod, J., Wang, C. C., Rosenthal, J. C., Fu, K. X., Zeffiro, T., Xia, W., Zhanshi, Z., Luo, H., Zhang, X. et al. (2021), ‘Stereo correspondence and reconstruction of endoscopic data challenge’, *arXiv preprint arXiv:2101.01133* .
- Antal, B. (2016), Automatic 3d point set reconstruction from stereo laparoscopic images using deep neural networks, PECCS 2016, Science and Technology Publications, p. 116–121.
- Antal, B. (2017), ‘Disparity map prediction from stereo laparoscopic images using a parallel deep convolutional neural network’.
- Asif, M., Song, H., Chen, L., Yang, J. and Frangi, A. F. (2020), ‘Intrinsic layer based automatic specular reflection detection in endoscopic images’, *Computers in Biology and Medicine* **128**, 104106.
- Bahraini, M. S., Rad, A. B. and Bozorg, M. (2019), ‘Slam in dynamic environments: A deep learning approach for moving object tracking using ml-ransac algorithm’, *Sensors* **19**(17), 3699.
- Barbalata, C. and Mattos, L. S. (2014), ‘Laryngeal tumor detection and classification in endoscopic video’, *IEEE journal of biomedical and health informatics* **20**(1), 322–332.

- Bardozzo, F., Coollins, T., Forgione, A., Hostettler, A. and Roberto, T. (2021), ‘Stasis-net: a stacked and siamese stereo network for depth estimation in modern 3d laparoscopy.’, *Paper submitted to Medical Image Analysis* **xx**, xx–xx.
- Bardozzo, F., De La Osa, B., Horanská, L., Fumanal-Idocin, J., delli Priscoli, M., Troiano, L., Tagliaferri, R., Fernandez, J. and Bustince, H. (2020), ‘Sugeno integral generalization applied to improve adaptive image binarization’, *Information Fusion* **68**, 37–45.
- Barreto, G. A. and Coelho, R. (2018), *Fuzzy Information Processing: 37th Conference of the North American Fuzzy Information Processing Society, NAFIPS 2018, Fortaleza, Brazil, July 4-6, 2018, Proceedings*, Vol. 831, Springer.
- Beliakov, G., James, S. and Wu, J.-Z. (2020), *Discrete fuzzy measures*, Springer.
- Bengio, Y., Goodfellow, I. and Courville, A. (2017), *Deep learning*, Vol. 1, MIT press Massachusetts, USA:.
- Bensignor, T., Morel, G., Reversat, D., Fuks, D. and Gayet, B. (2016), ‘Evaluation of the effect of a laparoscopic robotized needle holder on ergonomics and skills’, *Surgical endoscopy* **30**(2), 446–454.
- Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C. and Taubin, G. (1999), ‘The ball-pivoting algorithm for surface reconstruction’, *IEEE transactions on visualization and computer graphics* **5**(4), 349–359.
- Bernhardt, S., Nicolau, S. A., Soler, L. and Doignon, C. (2017), ‘The status of augmented reality in laparoscopic surgery as of 2016’, *Medical image analysis* **37**, 66–90.
- Bogdanova, R., Boulanger, P. and Zheng, B. (2016), ‘Depth perception of surgeons in minimally invasive surgery’, *Surgical innovation* **23**(5), 515–524.
- Bolkar, S., Wang, C., Cheikh, F. A. and Yildirim, S. (2018), Deep smoke removal from minimally invasive surgery videos, in ‘2018 25th IEEE International Conference on Image Processing (ICIP)’, IEEE, pp. 3403–3407.
- Bradley, D. and Roth, G. (2007), ‘Adaptive thresholding using the integral image’, *Journal of graphics tools* **12**(2), 13–21.
- Brandao, P., Mazomenos, E. and Stoyanov, D. (2019), ‘Widening siamese architectures for stereo matching’, *Pattern recognition letters* **120**, 75–81.
- Chadebecq, F., Vasconcelos, F., Mazomenos, E. and Stoyanov, D. (2020), ‘Computer vision in the surgical operating room’, *Visceral Medicine* pp. 1–7.

- Chen, L., Tang, W., John, N. W., Wan, T. R. and Zhang, J. J. (2019), ‘De-smokegcn: generative cooperative networks for joint surgical smoke detection and removal’, *IEEE transactions on medical imaging* **39**(5), 1615–1625.
- Chhatkuli, A., Bartoli, A., Malti, A. and Collins, T. (2014), Live image parsing in uterine laparoscopy, in ‘2014 IEEE 11th international symposium on biomedical imaging (ISBI)’, IEEE, pp. 1263–1266.
- Chien, H.-J., Chuang, C.-C., Chen, C.-Y. and Klette, R. (2016), When to use what feature? sift, surf, orb, or a-kaze features for monocular visual odometry, in ‘2016 International Conference on Image and Vision Computing New Zealand (IVCNZ)’, IEEE, pp. 1–6.
- Ciaparrone, G., Bardozzo, F., Priscoli, M. D., Kallewaard, J. L., Zuluaga, M. R. and Tagliaferri, R. (2020), A comparative analysis of multi-backbone mask r-cnn for surgical tools detection, in ‘2020 International Joint Conference on Neural Networks (IJCNN)’, IEEE, pp. 1–8.
- Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F. and Ranzuglia, G. (2008), Meshlab: an open-source mesh processing tool., in ‘Eurographics Italian chapter conference’, Vol. 2008, Salerno, pp. 129–136.
- Clevert, D., Unterthiner, T. and Hochreiter, S. (2016), Fast and accurate deep network learning by exponential linear units (elus), in Y. Bengio and Y. LeCun, eds, ‘4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings’.
- Collins, T. and Bartoli, A. (2012), Towards live monocular 3d laparoscopy using shading and specularity information, in ‘International Conference on Information Processing in Computer-Assisted Interventions’, Springer, pp. 11–21.
- Collins, T., Bartoli, A., Bourdel, N. and Canis, M. (2016), Robust, real-time, dense and deformable 3d organ tracking in laparoscopic videos, in ‘International Conference on Medical Image Computing and Computer-Assisted Intervention’, Springer, pp. 404–412.
- Crow, F. C. (1984), Summed-area tables for texture mapping, in ‘ACM SIGGRAPH computer graphics’, Vol. 18, ACM, pp. 207–212.
- Dimuro, G. P., Lucca, G., Bedregal, B., Mesiar, R., Sanz, J. A., Lin, C.-T. and Bustince, H. (2020), ‘Generalized cf1f2-integrals: From choquet-like aggregation to ordered directionally monotone functions’, *Fuzzy Sets and Systems* **378**, 44–67.

- Duan, C., Junginger, S., Huang, J., Jin, K. and Thurow, K. (2019), ‘Deep learning for visual slam in transportation robotics: A review’, *Transportation Safety and Environment* **1**(3), 177–184.
- El Baf, F., Bouwmans, T. and Vachon, B. (2008), Foreground detection using the Choquet integral, *in* ‘2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services’, IEEE, pp. 187–190.
- Foti, S., Koo, B., Dowrick, T., Ramalhinho, J., Allam, M., Davidson, B., Stoyanov, D. and Clarkson, M. J. (2020), ‘Intraoperative liver surface completion with graph convolutional vae’, *arXiv preprint arXiv:2009.03871*.
- Fritsch, J., Kuehnl, T. and Geiger, A. (2013), A new performance measure and evaluation benchmark for road detection algorithms, *in* ‘International Conference on Intelligent Transportation Systems (ITSC)’.
- Garg, R., Bg, V. K., Carneiro, G. and Reid, I. (2016), Unsupervised cnn for single view depth estimation: Geometry to the rescue, *in* ‘European conference on computer vision’, Springer, pp. 740–756.
- Geiger, A., Lenz, P., Stiller, C. and Urtasun, R. (2013), ‘Vision meets robotics: The kitti dataset’, *International Journal of Robotics Research (IJRR)*.
- Geiger, A., Lenz, P. and Urtasun, R. (2012), Are we ready for autonomous driving? the kitti vision benchmark suite, *in* ‘Conference on Computer Vision and Pattern Recognition (CVPR)’.
- Geiger, A., Roser, M. and Urtasun, R. (2010), Efficient large-scale stereo matching, *in* ‘Asian conference on computer vision’, Springer, pp. 25–38.
- Glorot, X. and Bengio, Y. (2010), Understanding the difficulty of training deep feed-forward neural networks, *in* ‘Proceedings of the thirteenth international conference on artificial intelligence and statistics’, pp. 249–256.
- Godard, C., Mac Aodha, O. and Brostow, G. J. (2017), Unsupervised monocular depth estimation with left-right consistency, *in* ‘Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition’, pp. 270–279.
- Godard, C., Mac Aodha, O., Firman, M. and Brostow, G. J. (2019), Digging into self-supervised monocular depth estimation, *in* ‘Proceedings of the IEEE international conference on computer vision’, pp. 3828–3838.
- Goldman, M., Hassner, T. and Avidan, S. (2019), Learn stereo, infer mono: Siamese networks for self-supervised, monocular, depth estimation, *in* ‘Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops’, pp. 0–0.

Grabisch, M., Marichal, J.-L., Mesiar, R. and Pap, E. (2009), *Aggregation functions*, Vol. 127, Cambridge University Press.

Hager, G. D., Taylor, R. H. and Reiter, A. (2018), Self-supervised learning for dense depth estimation in monocular endoscopy, in 'OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis. CARE 2018, 7th International Workshop, CLIP 2018, Third International Workshop, ISIC 2018, MICCAI 2018, Granada, Spain', Vol. 11041, Springer, p. 128.

Hamzah, R. A. and Ibrahim, H. (2016), 'Literature survey on stereo vision disparity map algorithms', *Journal of Sensors* **2016**.

He, K., Zhang, X., Ren, S. and Sun, J. (2016), Deep residual learning for image recognition, in 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 770–778.

Heiselman, J. S., Clements, L. W., Collins, J. A., Weis, J. A., Simpson, A. L., Geevarghese, S. K., Kingham, T. P., Jarnagin, W. R. and Miga, M. I. (2017), 'Characterization and correction of intraoperative soft tissue deformation in image-guided laparoscopic liver surgery', *Journal of Medical Imaging* **5**(2), 021203.

Horanská, L. and Šipošová, A. (2018), 'A generalization of the discrete Choquet and Sugeno integrals based on a fusion function', *Information Sciences* **451**, 83–99.

Hunter, J. D. (2007), 'Matplotlib: A 2d graphics environment', *Computing in Science and Engineering* **9**(3), 90–95.

Ikeuchi, K. (2014), *Computer vision: A reference guide*, Springer Publishing Company, Incorporated.

Jo, K., Choi, Y., Choi, J. and Chung, J. W. (2019), 'Robust real-time detection of laparoscopic instruments in robot surgery using convolutional neural networks with motion vector prediction', *Applied Sciences* **9**(14), 2865.

Kajihara, T., Funatomi, T., Makishima, H., Aoto, T., Kubo, H., Yamada, S. and Mukaigawa, Y. (2019), 'Non-rigid registration of serial section images by blending transforms for 3d reconstruction', *Pattern Recognition* **96**, 106956.

Kanopoulos, N., Vasanthavada, N. and Baker, R. L. (1988), 'Design of an image edge detection filter using the sobel operator', *IEEE Journal of solid-state circuits* **23**(2), 358–367.

- Kasagi, A., Nakano, K. and Ito, Y. (2014), Parallel algorithms for the summed area table on the asynchronous hierarchical memory machine, with gpu implementations, in ‘2014 43rd International Conference on Parallel Processing’, IEEE, pp. 251–260.
- Ketkar, N. and Santana, E. (2017), *Deep Learning with Python*, Vol. 1, Springer.
- Kobler, E., Klatzer, T., Hammernik, K. and Pock, T. (2017), Variational networks: connecting variational methods and deep learning, in ‘German conference on pattern recognition’, Springer, pp. 281–293.
- Laga, H., Jospin, L. V., Boussaid, F. and Bennamoun, M. (2020), ‘A survey on deep learning techniques for stereo-based depth estimation’, *arXiv preprint arXiv:2006.02535*.
- Leonard, S., Sinha, A., Reiter, A., Ishii, M., Gallia, G. L., Taylor, R. H. and Hager, G. D. (2018), ‘Evaluation and stability analysis of video-based navigation system for functional endoscopic sinus surgery on in vivo clinical data’, *IEEE transactions on medical imaging* **37**(10), 2185–2195.
- Li, R., Pan, J., Si, Y., Yan, B., Hu, Y. and Qin, H. (2019), ‘Specular reflections removal for endoscopic image sequences with adaptive-rpca decomposition’, *IEEE transactions on medical imaging* **39**(2), 328–340.
- Lin, B., Sun, Y., Qian, X., Goldgof, D., Gitlin, R. and You, Y. (2016), ‘Video-based 3d reconstruction, laparoscope localization and deformation recovery for abdominal minimally invasive surgery: a survey’, *The International Journal of Medical Robotics and Computer Assisted Surgery* **12**(2), 158–178.
- Lin, J., Clancy, N. T. and Elson, D. S. (2015), ‘An endoscopic structured light system using multispectral detection’, *International journal of computer assisted radiology and surgery* **10**(12), 1941–1950.
- Liu, X., Abdolmalaki, R. Y., Mancini, G. J. and Tan, J. (2017), ‘Optical design of an in vivo laparoscopic lighting system’, *Journal of biomedical optics* **22**(12), 125003.
- Lo, B., Scarzanella, M. V., Stoyanov, D. and Yang, G.-Z. (2008), Belief propagation for depth cue fusion in minimally invasive surgery, in ‘International Conference on Medical Image Computing and Computer-Assisted Intervention’, Springer, pp. 104–112.
- Lu, B., Chu, H. K., Huang, K. and Lai, J. (2019), ‘Surgical suture thread detection and 3-d reconstruction using a model-free approach in a calibrated stereo visual system’, *IEEE/ASME Transactions on Mechatronics* **25**(2), 792–803.

- Lu, Z. and Chen, Y. (2019), ‘Single image super resolution based on a modified u-net with mixed gradient loss’, *arXiv preprint arXiv:1911.09428*.
- Luo, H., Hu, Q. and Jia, F. (2019), ‘Details preserved unsupervised depth estimation by fusing traditional stereo knowledge from laparoscopic images’, *Healthcare Technology Letters* **6**(6), 154–158.
- Luo, H., Yin, D., Zhang, S., Xiao, D., He, B., Meng, F., Zhang, Y., Cai, W., He, S., Zhang, W. et al. (2020), ‘Augmented reality navigation for liver resection with a stereoscopic laparoscope’, *Computer Methods and Programs in Biomedicine* **187**, 105099.
- Mahmood, F. and Durr, N. J. (2018), ‘Deep learning and conditional random fields-based depth estimation and topographical reconstruction from conventional endoscopy’, *Medical image analysis* **48**, 230–243.
- Mahmoud, N., Cirauqui, I., Hostettler, A., Doignon, C., Soler, L., Marescaux, J. and Montiel, J. (2016), Orbslam-based endoscope tracking and 3d reconstruction, in ‘International workshop on computer-assisted and robotic endoscopy’, Springer, pp. 72–83.
- Maier-Hein, L., Groch, A., Bartoli, A., Bodenstedt, S., Boissonnat, G., Chang, P.-L., Clancy, N., Elson, D. S., Haase, S., Heim, E. et al. (2014), ‘Comparative validation of single-shot optical techniques for laparoscopic 3-d surface reconstruction’, *IEEE transactions on medical imaging* **33**(10), 1913–1930.
- Maier-Hein, L., Mountney, P., Bartoli, A., Elhawary, H., Elson, D., Groch, A., Kolb, A., Rodrigues, M., Sorger, J., Speidel, S. et al. (2013), ‘Optical techniques for 3d surface reconstruction in computer-assisted laparoscopic surgery’, *Medical image analysis* **17**(8), 974–996.
- Maimaitimin, M., Watanabe, K. and Maeyama, S. (2017), ‘Stacked convolutional auto-encoders for surface recognition based on 3d point cloud data’, *Artificial Life and Robotics* **22**(2), 259–264.
- Malti, A., Bartoli, A. and Collins, T. (2011), Template-based conformal shape-from-motion from registered laparoscopic images., in ‘MIUA’, Vol. 1, p. 6.
- MATLAB (2010), *version 7.10.0 (R2010a)*, The MathWorks Inc., Natick, Massachusetts.
- Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A. and Brox, T. (2016), A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation, in ‘Proceedings of the IEEE conference on computer vision and pattern recognition’, pp. 4040–4048.

- Menze, M. and Geiger, A. (2015), Object scene flow for autonomous vehicles, *in* ‘Conference on Computer Vision and Pattern Recognition (CVPR)’.
- Modrzejewski, R., Collins, T., Seeliger, B., Bartoli, A., Hostettler, A. and Marescaux, J. (2019), ‘An in vivo porcine dataset and evaluation methodology to measure soft-body laparoscopic liver registration accuracy with an extended algorithm that handles collisions’, *International journal of computer assisted radiology and surgery* **14**(7), 1237–1245.
- Park, Y. S., Oo, A. M., Son, S.-Y., Shin, D. J., Ahn, S.-H., Park, D. J., Kim, H.-H. et al. (2016), ‘Is a robotic system really better than the three-dimensional laparoscopic system in terms of suturing performance?: comparison among operators with different levels of experience’, *Surgical endoscopy* **30**(4), 1485–1490.
- Penza, V., Ciullo, A. S., Moccia, S., Mattos, L. S. and De Momi, E. (2018), ‘Endoabs dataset: Endoscopic abdominal stereo image dataset for benchmarking 3d stereo reconstruction algorithms’, *The International Journal of Medical Robotics and Computer Assisted Surgery* **14**(5), e1926.
- Pratt, P., Stoyanov, D., Visentini-Scarzanella, M. and Yang, G.-Z. (2010), Dynamic guidance for robotic surgery using image-constrained biomechanical models, *in* ‘International Conference on Medical Image Computing and Computer-Assisted Intervention’, Springer, pp. 77–85.
- Prevost, G. A., Eigl, B., Paolucci, I., Rudolph, T., Peterhans, M., Weber, S., Beldi, G., Candinas, D. and Lachenmayer, A. (2019), ‘Efficiency, accuracy and clinical applicability of a new image-guided surgery system in 3d laparoscopic liver surgery’, *Journal of gastrointestinal surgery* pp. 1–8.
- Qin, Y. and Hua, H. (2016), ‘Optical design and system engineering of a multiresolution foveated laparoscope’, *Applied optics* **55**(11), 3058–3068.
- Raina, R. (2009), Self-Taught Learning, PhD thesis, Stanford, CA, USA. AAI3382948.
- Sabater, N., Morel, J.-M. and Almansa, A. (2011), ‘How accurate can block matches be in stereo vision?’, *SIAM Journal on Imaging Sciences* **4**(1), 472–500.
- Schwarz, L. A. (2007), ‘Non-rigid registration using free-form deformations’, *Technische Universität München*.
- Simonyan, K. and Zisserman, A. (2015), Very deep convolutional networks for large-scale image recognition, *in* ‘International Conference on Learning Representations’.

- Sinha, R. Y., Raje, S. R. and Rao, G. A. (2017), ‘Three-dimensional laparoscopy: principles and practice’, *Journal of Minimal Access Surgery* **13**(3), 165.
- Stoyanov, D., Scarzanella, M. V., Pratt, P. and Yang, G.-Z. (2010), Real-time stereo reconstruction in robotically assisted minimally invasive surgery, in ‘International Conference on Medical Image Computing and Computer-Assisted Intervention’, Springer, pp. 275–282.
- Suwelack, S., Röhl, S., Bodenstedt, S., Reichard, D., Dillmann, R., dos Santos, T., Maier-Hein, L., Wagner, M., Wünscher, J., Kenngott, H. et al. (2014), ‘Physics-based shape matching for intraoperative image guidance’, *Medical physics* **41**(11), 111901.
- Tan, K., Wang, B. and Gao, Y. (2017), Automatic specular reflections removal for endoscopic images, in ‘Ninth International Conference on Digital Image Processing (ICDIP 2017)’, Vol. 10420, International Society for Optics and Photonics, p. 104203Y.
- Tateno, K., Tombari, F., Laina, I. and Navab, N. (2017), Cnn-slam: Real-time dense monocular slam with learned depth prediction, in ‘Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition’, pp. 6243–6252.
- Tosi, F., Aleotti, F., Poggi, M. and Mattoccia, S. (2019), Learning monocular depth estimation infusing traditional stereo knowledge, in ‘Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition’, pp. 9799–9809.
- Trottier, L., Giguere, P. and Chaib-draa, B. (2017), Parametric exponential linear unit for deep convolutional neural networks, in ‘2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)’, pp. 207–214.
- Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E. and Yu, T. (2014), ‘scikit-image: image processing in python’, *PeerJ* **2**, e453.
- Veličković, P., Wang, D., Lane, N. D. and Liò, P. (2016), X-cnn: Cross-modal convolutional neural networks for sparse datasets, in ‘2016 IEEE symposium series on computational intelligence (SSCI)’, IEEE, pp. 1–8.
- Vishal, V., Venkatesh, V., Lochan, K., Sharma, N. and Singh, M. (2020), Unsupervised desmoking of laparoscopy images using multi-scale desmokenet, in ‘International Conference on Advanced Concepts for Intelligent Vision Systems’, Springer, pp. 421–432.
- Wagner, M., Mayer, B. F. B., Bodenstedt, S., Stemmer, K., Fereydooni, A., Speidel, S., Dillmann, R., Nickel, F., Fischer, L. and Kenngott, H. G. (2018), ‘Computer-assisted 3d bowel length measurement for quantitative laparoscopy’, *Surgical endoscopy* **32**(9), 4052–4061.

- Wang, C., Cheikh, F. A., Kaaniche, M. and Elle, O. J. (2018), Liver surface reconstruction for image guided surgery, *in* ‘Medical Imaging 2018: Image-Guided Procedures, Robotic Interventions, and Modeling’, Vol. 10576, International Society for Optics and Photonics, p. 105762H.
- Wang, T.-C., Efros, A. A. and Ramamoorthi, R. (2016), ‘Depth estimation with occlusion modeling using light-field cameras’, *IEEE transactions on pattern analysis and machine intelligence* **38**(11), 2170–2181.
- Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P. (2004), ‘Image quality assessment: from error visibility to structural similarity’, *IEEE transactions on image processing* **13**(4), 600–612.
- Westebring-van der Putten, E. P., Goossens, R. H., Jakimowicz, J. J. and Dankelman, J. (2008), ‘Haptics in minimally invasive surgery—a review’, *Minimally Invasive Therapy and Allied Technologies* **17**(1), 3–16.
- Xu, K., Chen, Z. and Jia, F. (2019), ‘Unsupervised binocular depth prediction network for laparoscopic surgery’, *Computer Assisted Surgery* **24**(sup1), 30–35.
- Yamaguchi, K., McAllester, D. and Urtasun, R. (2014), Efficient joint segmentation, occlusion labeling, stereo and flow estimation, *in* ‘European Conference on Computer Vision’, Springer, pp. 756–771.
- Ye, M., Johns, E., Handa, A., Zhang, L., Pratt, P. and Yang, G.-Z. (2017), ‘Self-supervised siamese learning on stereo image pairs for depth estimation in robotic surgery’, *Hamlyn Symposium on Medical Robotics*. .
- Yusiong, J. P. and Naval, P. (2019), Asianet: Autoencoders in autoencoder for unsupervised monocular depth estimation, *in* ‘2019 IEEE Winter Conference on Applications of Computer Vision (WACV)’, IEEE, pp. 443–451.
- Zabih, R. and Woodfill, J. (1994), Non-parametric local transforms for computing visual correspondence, *in* ‘European conference on computer vision’, Springer, pp. 151–158.
- Žbontar, J. and LeCun, Y. (2016), ‘Stereo matching by training a convolutional neural network to compare image patches’, *The journal of machine learning research* **17**(1), 2287–2318.
- Zhao, H., Gallo, O., Frosio, I. and Kautz, J. (2016), ‘Loss functions for image restoration with neural networks’, *IEEE Transactions on computational imaging* **3**(1), 47–57.
- Zhou, K., Meng, X. and Cheng, B. (2020), ‘Review of stereo matching algorithms based on deep learning’, *Computational Intelligence and Neuroscience* **2020**.

Zhou, Q.-Y., Park, J. and Koltun, V. (2018), ‘Open3D: A modern library for 3D data processing’, *arXiv:1801.09847* .

LIST OF PUBLICATIONS

Journal

1. **Bardozzo F.**, Collins T., Forgione A. , Hostettler A. and Tagliaferri R. (2021) - StaSiS-Net: a stacked and siamese stereo network for depth estimation in modern 3D laparoscopy. (Submitted to Medical Image Analysis - Elsevier)
2. De la Osa B., Wieczynski, J., Pereira, G., Asmus, T., Emmendorfer, L., Fernandez, J., Fumanal-Idocin, J., **Bardozzo, F.**, Sanz, J. A., Bustince, H. (2021) - Measures for comparing fuzzy rules and their application for the selection of the best method for the Generalized Modus Ponens. (Submitted to IEEE Transactions on Fuzzy Systems)
3. Delli Priscoli M., Memmolo P., Ciaparrone G., Bianco V., Merola F., Miccio L., **Bardozzo F.**, Pirone D., Mugnano M., Cimmino F., Capasso M., Iolascon A., Ferraro P., Tagliaferri R. (2021) - Neuroblastoma cells classification through learning approaches by direct analysis of digital holograms. (Open Access) IEEE Journal of Selected Topics in Quantum Electronics, 1-1.
4. **Bardozzo F.**, De La Osa B., Horanská Ľ., Fumanal-Idocin J., delli Priscoli M., Troiano L., Tagliaferri R., Bustince H. (2020) - Sugeno integral generalization applied to improve adaptive image binarization. Information Fusion, 68, 37-45.
5. **Bardozzo F.**, Lió P., Tagliaferri R. - (2020) - Signal metrics analysis of oscillatory patterns in bacterial multi-omic networks. Bioinformatics, 1367-4803.
6. **Bardozzo F.**, Lió P., Tagliaferri R. (2018) - A study on multi-omic oscillations in Escherichia coli metabolic networks. BMC bioinformatics, 19(7), 194.

Conference

1. Hernán Gil Zuluaga, **F., Bardozzo**, F., Iván Ríos Patiño, J., Tagliaferri, R. (2021). Blind microscopy image denoising with a deep residual and multiscale encoder decoder network. Submitted to 2021 Engineering in Medicine and Biology Conference (EMBC) (pp. 1-4). IEEE.
2. **Bardozzo F.**, Uribe S. C., Russo A. G., Castaño M. J., Priscoli M. D., Esposito F., Tagliaferri R. (2020, July) - Motor strength classification with machine learning approaches applied to anatomical neuroimages. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
3. Ciaparrone G., **Bardozzo F.**, Priscoli M. D., Kallewaard J. L., Zuluaga M. R., Tagliaferri R. (2020, July) - A comparative analysis of multi-backbone Mask R-CNN for surgical tools detection. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
4. **Bardozzo F.**, Lió P., Tagliaferri R. (2015, July) - Multi omic oscillations in bacterial pathways. In 2015 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

Appendices

Appendix A

Insights on Chapter 4

A.1 Additional collection of visual and numerical results on Test Set D.

Additional collection of visual and numerical results on Test Set D and provided in the following [\[Google Drive\]](#)

A.2 Visual results: additional information.

Real-time depth estimations are reprojected both into real-time 3D point clouds and into close-to-real-time 3D Meshes on a mid-budget GPU. In our case, these experiments are performed on an *Nvidia RTX 2060*. For a visual quality assessment, on this [\[YouTube Video\]](#), it is provided the right channel video with ground truth of an internal deformed by pulsations surface (Test Set *D* - Scene 4 *DEFORM* challenge). In addition a real surgical video 3D reconstruction (with the presence of smoke) is provided (The latter is a part of the collection of videos provided by the Vision department of *Hamlyn centre*). Furthermore, *Mono depth CNN* and *Stereo depth pipeline* outputs are shown in 3D point clouds and 3D meshes. It is important to underline that the point cloud changes in close-to-real-time when the depth estimates are produced in close-to-real-time. On the contrary, the representation of the 3D Mesh requires several steps and it is necessary to find a compromise between the quality of reconstruction and performance over time.

A.3 Learning of the stereo displacement function

Mono depth CNN and *Stereo depth CNN* reconstructions are based on learning from stereo rectified images the non-linear relations of the displacement function f_δ . The f_δ should minimize the vertical displacement between monocular camera views in a stereo configuration. As it is described in Chapter 4, in the training phase the alternate system of targeting the right/left images with respect to left/right ones generates the depth information needed for depth estimation. In Figure A.1 - Box (a), the stereo images (left and right channel) are shown for the blue point displaced. In Figure A.1 - Box (b), the intuitive process of learning the displacement function is shown. In

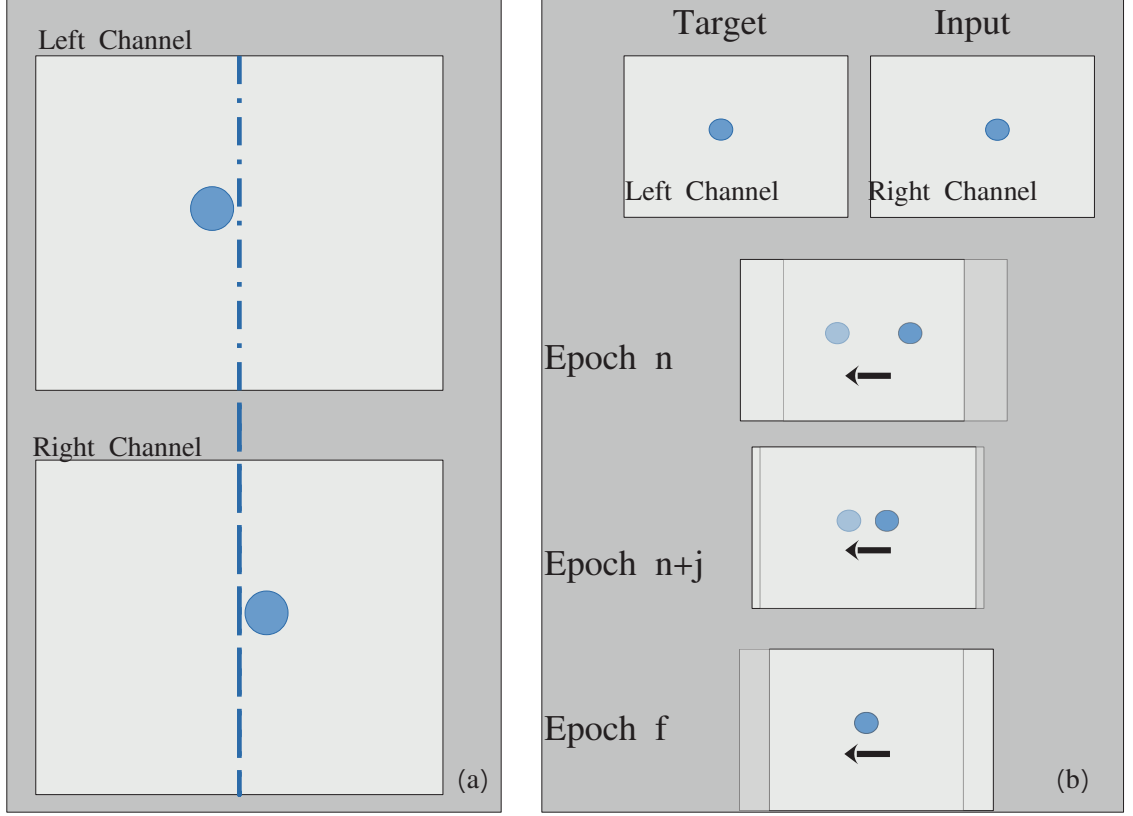


Fig. A.1 In Figure A.1 - Box (a), the stereo images (left and right channel) are shown for the blue point displaced. In Figure A.1 - Box (b), the intuitive process of learning the displacement function is shown. The displacement function f_δ is shown with the black arrow across the learning epochs.

this process, if the encoder/decoder is very 'accurate' with the 'wrong' losses and/or with a 'wrong' architecture design, the risk is that the transformation learned by the f_δ is useless because it reconstructs exactly one the two channel images. The same risk is also described for autoencoders in [Bengio et al. \(2017\)](#). While there are several strategies to make sure this doesn't happen, there is no real theoretical rule that describes what is 'wrong'. For this reason hundreds of experiments have been carried out and dozens of different architectures have been tried before arriving at the deep learning model described in this thesis.

Appendix B

Insights on Chapter 5

The Sections described in Appendix B are mainly extracted from the paper [Bardozzo et al. \(2020\)](#). Even if they are adapted to this Thesis with additional comments and observations.

B.1 Fuzzy Theory

B.1.1 Fuzzy measure

Let $n \in \mathbb{N}$, $[n] = \{1, \dots, n\}$. A set function $\mu: 2^{[n]} \rightarrow [0, 1]$ is a *discrete fuzzy measure* ([Beliakov et al. 2020](#)), if the following conditions are satisfied:

- $\mu(A) \leq \mu(B)$ whenever $A \subseteq B$,
- $\mu(\emptyset) = 0$, $\mu([n]) = 1$.

A fuzzy measure μ is *symmetric*, if for any $A, B \subseteq [n]$, $|A| = |B|$ implies $\mu(A) = \mu(B)$.

B.1.2 Uniform fuzzy measure

Given $|E|$ the cardinality of E and n a natural number which represents the set size, the uniform fuzzy measure μ_{uni} defined by formula (B.1) is applied on E as follows:

$$\mu_{uni}(E) = \frac{|E|}{n}, \quad (\text{B.1})$$

for $E \subseteq [n]$, is symmetric.

B.1.3 Aggregation function

A function $A: [0, \infty[^n \rightarrow [0, \infty[$ is an *aggregation function*, if A is nondecreasing and

$$\inf_{\mathbf{x} \in [0, \infty[^n} A(\mathbf{x}) = 0, \quad \sup_{\mathbf{x} \in [0, \infty[^n} A(\mathbf{x}) = \infty.$$

An aggregation function $A: [0, \infty[^n \rightarrow [0, \infty[$ is

- *internal*, if $\bigwedge_{i=1}^n x_i \leq A(x_1, \dots, x_n) \leq \bigvee_{i=1}^n x_i$, for each $(x_1, \dots, x_n) \in [0, \infty[^n$.
- *translation invariant*, if $A(x_1 + c, \dots, x_n + c) = A(x_1, \dots, x_n) + c$, for all $c \in]0, \infty[$ and $(x_1, \dots, x_n) \in [0, \infty[^n$.
- *idempotent*, if $A(x, \dots, x) = x$, for each $x \in [0, \infty[$.
- *positively homogeneous*, if $A(cx) = cA(x)$, for each $x \in [0, \infty[^n$ and $c > 0$.
- *comonotone additive*, if $A(x + \bar{x}) = A(x) + A(\bar{x})$, for all comonotone vectors $x, \bar{x} \in [0, \infty[^n$ (vectors $x = (x_1, \dots, x_n)$, $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n)$ are comonotone, if $(x_i - x_j)(\bar{x}_i - \bar{x}_j) \geq 0$ for all $i, j \in \{1, \dots, n\}$).
- *comonotone maxitive (comonotone minitive)*, if $A(x \vee \bar{x}) = A(x) \vee A(\bar{x})$ ($A(x \wedge \bar{x}) = A(x) \wedge A(\bar{x})$), for all comonotone vectors $x, \bar{x} \in [0, \infty[^n$.

Let $\mu: 2^{[n]} \rightarrow [0, 1]$ be a fuzzy measure.

B.1.4 Sugeno integral definition

The *Sugeno integral* with respect to fuzzy measure μ is given by

$$Su_\mu(x) = \bigvee_{i=1}^n (x_{(i)} \wedge \mu(E_{(i)})), \quad (\text{B.2})$$

for $x = (x_1, \dots, x_n) \in [0, \infty[^n$, with the same meaning of $x_{(i)}$ and $E_{(i)}$, $i = 1, \dots, n$, as above.

The Sugeno integral is not bounded by the minimum from below, but it is bounded by the maximum from above. It is neither idempotent nor positively homogeneous (however, the Sugeno integral is an idempotent, internal, positively homogeneous function on the interval $[0, 1]$). It gives back the considered fuzzy measure, i.e., $Su_\mu(\mathbf{1}_E) = \mu(E)$, for each $E \subseteq [n]$.

Moreover, the Sugeno integral is comonotone maxitive and comonotone minitive (for more details see, e.g., [Grabisch et al. \(2009\)](#)).

B.1.5 Generalized Sugeno integral

It is proposed a generalization of the Sugeno integral presented in Eq. (B.2). To do so, maximum and minimum operators are replaced by some more general functions.

Definition B.1.1. Let $\mu: 2^{[n]} \rightarrow [0, 1]$ be a symmetric fuzzy measure, $F: [0, \infty[\times [0, 1] \rightarrow [0, \infty[$ be a binary function, $G: [0, \infty[^n \rightarrow [0, \infty[$ be an n -ary function. A Sugeno-like

FG-functional is a function $A: [0, \infty[^n \rightarrow [0, \infty[$ given by

$$A(x_1, \dots, x_n) = G \left(F(x_{(1)}, \mu(E_{(1)})), \dots, F(x_{(n)}, \mu(E_{(n)})) \right), \quad (\text{B.3})$$

for $\mathbf{x} = (x_1, \dots, x_n) \in [0, \infty[^n$, with the same meaning of $x_{(i)}$ and $E_{(i)}$, $i = 1, \dots, n$, as above.

The definition correctness depends on whether the functional A given by formula (B.3) gives back the same value if some ties occur in a vector $\mathbf{x}$ and there is more than one permutation ordering this vector non-decreasingly. The symmetry of the fuzzy measure μ considered in Definition B.1.1 ensures that functional A is well-defined. In fact, for particular cases of G , assumptions under which A is well-defined can be weakened. For example, the case of G being the maximum operator and F an arbitrary fusion function was deeply studied in [Horanská and Šipošová \(2018\)](#), wherein assumptions under which A is well-defined for an arbitrary fuzzy measure μ and a complete characterization of functional A and its properties can be found.

Let $G(x_1, \dots, x_n) = \sum_{i=1}^n x_i$ and $F(x, y) = x \cdot y$. Then we obtain

$$CF_{12} = A(\mathbf{x}) = \sum_{i=1}^n (x_{(i)} \cdot \mu(E_{(i)})). \quad (\text{B.4})$$

CF_{12} is bounded by the minimum from below, but not bounded by the maximum from above. CF_{12} is positively homogeneous and it is idempotent, giving back the capacity, comonotone additive, comonotone maxitive and translation invariant.

B.2 GPU-oriented implementation in python/numba of CF_{12} -optimized-algorithm

```
import warnings
import sys
from numba import jitclass, int32, float32
import numpy as np

if not sys.warnoptions:
    warnings.simplefilter("ignore")

## GPU allocations
spec = [
    ('image', float32[:, :]),
    ('gt', float32[:, :]),
    ('height', int32),
    ('width', int32),
    ('S', float32[:, :]),
    ('S_c', float32[:, :]),
    ('out_img', float32[:, :]),
    ('th_mat', float32[:, :]),
]

@jitclass(spec)
```

```

class fuzzy_sat:
    def __init__(self, image):
        shape = image.shape
        self.image = np.asarray(image, dtype=float32)
        self.height = shape[0]
        self.width = shape[1]
        self.S = np.zeros(shape, dtype=float32) # Create an empty summed area table
        self.S_c = np.zeros(shape, dtype=float32) # Create an empty summed area table
        self.out_img = np.zeros(shape, dtype=float32)
        self.th_mat = np.zeros(shape, dtype=float32)

    def compute_sat(self): #Non-fuzzy Integral image
        for row in range(0, self.height):
            for col in range(0, self.width):
                if (row > 0) and (col > 0):
                    self.S[row][col] = self.image[row][col] + self.S[row][col - 1] + self.S[row - 1][col] - \
                        self.S[row - 1][col - 1]
                elif row > 0:
                    self.S[row][col] = self.image[row][col] + self.S[row - 1][col]
                elif col > 0:
                    self.S[row][col] = self.image[row][col] + self.S[row][col - 1]
                else:
                    self.S[row][col] = self.image[row][col]

    def compute_sat_cf12(self): #Fuzzy CFval 1,2 integral image
        for row in range(0, self.height):
            for col in range(0, self.width):
                if (row > 0) and (col > 0):
                    self.S[row][col] = self.image[row][col] + self.S[row][col - 1] + self.S[row - 1][col] - \
                        self.S[row - 1][col - 1]
                    if (self.S[row][col - 1] > self.S[row - 1][col]):
                        ov = np.asarray(
                            [0, self.S[row - 1][col - 1], self.S[row - 1][col], self.S[row][col - 1], self.S[row][col]])
                    else:
                        ov = np.asarray(
                            [0, self.S[row - 1][col - 1], self.S[row][col - 1], self.S[row - 1][col], self.S[row][col]])
                    self.S_c[row][col] = ov[1] + ov[2] * 0.75 + ov[3] * 0.5 + ov[4] * 0.25
                elif row > 0:
                    self.S[row][col] = self.image[row][col] + self.S[row - 1][col]
                    ov = np.asarray([0, self.S[row - 1][col], self.S[row][col]])
                    self.S_c[row][col] = ov[1] + ov[2] * 0.50
                elif col > 0:
                    self.S[row][col] = self.image[row][col] + self.S[row][col - 1]
                    ov = np.asarray([0, self.S[row - 1][col], self.S[row][col]])
                    self.S_c[row][col] = ov[1] + ov[2] * 0.50
                else:
                    self.S[row][col] = self.image[row][col]
                    self.S_c[row][col] = self.image[row][col]

    def adaptive_thresh_fuzzy(self, al, T): #fuzzy adaptive binarization
        w_n = min(self.height, self.width) / al
        for col in range(self.width):
            for row in range(self.height):
                # SxS region
                y0 = int(max(row - w_n, 0))
                y1 = int(min(row + w_n, self.height - 1))
                x0 = int(max(col - w_n, 0))
                x1 = int(min(col + w_n, self.width - 1))

                count = (y1 - y0) * (x1 - x0)
                sum_ = self.S_c[y1, x1] - self.S_c[y0, x1] - self.S_c[y1, x0] + self.S_c[y0, x0]

                self.th_mat[row, col] = sum_ / count

                if self.image[row, col] * count < sum_ * (1. - T): #it is for values in [0,1]
                    self.out_img[row, col] = 0
                else:
                    self.out_img[row, col] = 1

    def get_S(self):
        return (self.S)
    def get_S_c(self):
        return (self.S_c)
    def get_FTh(self):
        return (self.out_img)

#usage
#int_img = fuzzy_sat(np.asarray(depth_map_image)) # Concretize the class instance
#int_img.compute_sat_cf12() # Compute fuzzy integral depth map
#out_img = int_img.get_FTh() # Obtain the binarized depth

```