

University of Modena and Reggio Emilia
XXXIV Cycle of the International Doctorate School in
Information and Communication Technologies

Doctor of Philosophy Dissertation in
Computer Engineering and Science

Vision-based Human-Vehicle Interaction

Stefano Pini

Supervisor: Prof. Rita Cucchiara
Co-Supervisor: Prof. Roberto Vezzani
PhD Course Coordinator: Prof. Sonia Bergamaschi
Modena, 2022

Review committee:

Robert B. Fisher
University of Edinburgh

Matteo Poggi
University of Bologna

There are frontiers where we are learning, and our desire for knowledge burns. They are in the most minute reaches of the fabric of space, at the origins of the cosmos, in the nature of time, in the phenomenon of black holes, and in the workings of our own thought processes. Here, on the edge of what we know, in contact with the ocean of the unknown, shines the mystery and the beauty of the world. And it's breathtaking.

Seven Brief Lessons on Physics
Carlo Rovelli

Abstract

In recent years, the widespread adoption of digital devices in all aspects of everyday life has led to new research opportunities in the field of Human-Computer Interaction. In the automotive field, where infotainment systems are becoming more and more important to the final user, the availability of inexpensive miniaturized cameras has enabled the development of vision-based Natural User Interfaces, paving the way for novel approaches to the Human-Vehicle Interaction.

In this thesis, we investigate computer vision techniques, based on both visible light and non-visible spectrum, that can replace existing means of human-computer interaction and form the foundation of the next generation of in-vehicle infotainment systems. As sensing technology, we focus on infrared-based devices, such as depth and thermal cameras. They provide reliable data under different illumination conditions, making them a good fit for the mutable automotive environment. Using these acquisition devices, we collect four novel datasets: a facial dataset, to investigate the impact of sensor resolution and quality in changing acquisition settings, a dataset of dynamic hand gestures, collected with several synchronized sensors within a car simulator, a refined set of annotated human poses and a dataset for the estimation of anthropometric measurements from visual data. As vision approaches, we adopt state-of-the-art deep learning techniques, focusing on efficient neural networks that can be easily deployed on computing devices on the edge. In this context, we study several computer vision tasks to cover the majority of human-car interactions. First, we investigate the usage of depth cameras for the face recognition task, focusing on how depth map representations and deep neural models affect the recognition performance. Secondly, we address the problem of in-car dynamic hand gesture recognition in real-time, using depth and infrared sensors. Then, we focus on the analysis of the human body, in terms of both the 3D human pose estimation and the contact-free estimation of anthropometric measurements. Finally, focusing on the area surrounding the vehicle, we explore the 3D reconstruction of objects from 2D images, as a first step towards the 3D visualization of the external environment from controllable viewpoints.

Abstract in lingua italiana

Negli ultimi anni, la diffusione di dispositivi digitali in ogni aspetto della vita quotidiana ha portato a nuove opportunità nel campo dell'Interazione Uomo-Macchina. Nel campo automobilistico, dove i sistemi di infotainment sono sempre più importanti per gli utenti finali, la disponibilità di telecamere economiche e miniaturizzate ha permesso lo sviluppo di interfacce utente naturali basate sulla visione artificiale, aprendo a nuove opportunità nell'Interazione Uomo-Veicolo.

In questa tesi, si propone uno studio di tecniche di visione artificiale, basate sia su luce visibile che sullo spettro non visibile, che possano sostituire gli attuali mezzi di interazione uomo-macchina e formare la base per la prossima generazione di sistemi di infotainment. Come tecnologie di acquisizione, il focus è posto su dispositivi basati su luce infrarossa, come camere termiche e di profondità. Queste tipologie di sensori forniscono dati affidabili in numerose condizioni di illuminazione pertanto sono particolarmente adatte al dinamico ambiente automobilistico. Con questi dispositivi, sono acquisiti quattro dataset: un dataset di volti, per valutare l'impatto di qualità e risoluzione dei sensori in configurazioni di acquisizione variabile, un dataset di gesti dinamici della mano, acquisito in un simulatore di auto con molteplici sensori sincronizzati, un set rifinito di posture umane e un dataset per la stima di misure antropometriche da dati visuali. Come approcci di visione, si sceglie di utilizzare tecniche di deep learning stato dell'arte, focalizzandosi su reti neurali efficienti che possano essere utilizzate su dispositivi integrati a basso consumo. In questo contesto, sono esaminati diversi problemi di visione artificiale, con l'obiettivo di coprire la maggior parte delle interazioni uomo-macchina. Innanzitutto, si analizza l'utilizzo di camere di profondità per il riconoscimento facciale, focalizzandosi sull'impatto che la rappresentazione dei dati di profondità e il tipo di architettura neurale utilizzata hanno sulle capacità di riconoscimento. Inoltre, si studia il riconoscimento di gesti dinamici della mano in tempo reale, utilizzando sensori infrarosso e di profondità. Si analizza anche l'intero corpo umano, in termini di riconoscimento della postura 3D e di stima senza contatto di misure antropometriche. Infine, focalizzandosi sull'area circostante il veicolo, si affronta la ricostruzione 3D di oggetti da immagini 2D, come primo passo verso una visualizzazione 3D navigabile dell'ambiente esterno.

Contents

Abstract	iii
Abstract in lingua italiana	iv
Contents	v
List of Figures	viii
List of Tables	x
1 Introduction	1
2 Background	6
2.1 Vision Beyond the Visible Spectrum	7
2.1.1 Depth Sensors	8
2.1.2 Depth Map Representations	11
2.1.3 Near-Infrared and Thermal Cameras	14
2.2 Literature Survey	17
2.2.1 Face Recognition	17
2.2.2 Hand Gesture Recognition	21
2.2.3 Human Pose Estimation and Refinement	25
2.2.4 Anthropometric Measurements	27
2.2.5 3D Object Reconstruction	28
3 Datasets	31
3.1 Collected Datasets	32
3.1.1 Faces - MultiSFace	32

3.1.2	Dynamic Hand Gestures - Briareo	35
3.1.3	Human Poses - Watch-R(efined)-Patch	37
3.1.4	Anthropometry - Baracca	39
3.2	Existing Datasets	43
3.2.1	Faces	43
3.2.2	Dynamic Hand Gestures	45
3.2.3	Human Poses	46
3.2.4	3D Objects	47
3.3	Connection to the proposed methods	48
4	Proposed Methods and Experimental Results	49
4.1	Depth Map Representations for Face Recognition	53
4.1.1	Methodology	56
4.1.2	Experimental Evaluation	61
4.1.3	Discussion	68
4.2	Dynamic Hand Gesture Recognition	70
4.2.1	Proposed Method	71
4.2.2	Experimental Evaluation	76
4.2.3	Discussion	84
4.3	3D Human Pose Estimation and Refinement	85
4.3.1	2D Human Pose Estimation from Depth Maps	85
4.3.2	3D Human Pose Refinement from Depth Maps	92
4.3.3	Discussion	104
4.4	Estimation of Anthropometric Measurements	106
4.4.1	Proposed Baselines	107
4.4.2	Experimental Evaluation	108
4.4.3	Discussion	112
4.5	3D Reconstruction of Vehicles	113
4.5.1	Proposed Method	115
4.5.2	Experimental Evaluation	119
4.5.3	Discussion	126
5	Conclusions and Future Work	130
5.1	Depth Map Representations for Face Recognition	131
5.2	Dynamic Hand Gesture Recognition	132
5.3	Human Pose Estimation and Refinement from Depth Maps	132
5.4	Estimation of Anthropometric Measurements	133
5.5	3D Reconstruction of Vehicles	134

Appendices	136
A List of publications	136
B Activities carried out during the PhD	140
Bibliography	145

List of Figures

3.1	Samples from the MultiSFace dataset.	33
3.2	Samples from the Briareo dataset.	35
3.3	Gesture classes included in the Briareo dataset.	36
3.4	Overview of the human pose annotation tool.	38
3.5	Samples from the Watch-R-Patch dataset.	40
3.6	Samples from the Baracca dataset.	41
4.1	Facial depth map representations.	55
4.2	Pre-processing techniques applied to depth images.	60
4.3	Proposed dynamic hand gesture recognition method.	72
4.4	Sample depth maps and estimated surface normals from the NVGestures dataset.	75
4.5	Sample depth maps and estimated surface normals from the Briareo dataset.	76
4.6	Confusion matrix of the multimodal approach on NVGestures.	81
4.7	Overview of the proposed HPE method.	86
4.8	Qualitative results on the Watch-R-Patch dataset (kitchen).	90
4.9	Qualitative results on the Watch-R-Patch dataset (office).	91
4.10	Recovering a 3D human pose from a depth image with RefiNet or a baseline approach.	93
4.11	Overview of the modules that compose the RefiNet framework.	94
4.12	Ablation study on RefiNet Module A parameters.	100
4.13	Ablation study on RefiNet Module B parameters.	101
4.14	Ablation study on RefiNet Module C parameters.	101
4.15	Results of the RefiNet framework.	103
4.16	Overview of the proposed 3D reconstruction approach.	114

4.17	Detailed diagram of the proposed 3D reconstruction method.	116
4.18	Meanshapes learned training on Pascal3D+ (aeroplane, car).	122
4.19	Meanshapes learned training on CUB.	124
4.20	Meanshapes learned training on Pascal3D+ (bicycle, bus, car, motorbike).	125
4.21	Qualitative results on CUB and Pascal3D+.	127
4.22	Qualitative results on Pascal3D+ (all classes).	128

List of Tables

2.1	Infrared wavelength bands.	15
2.2	Datasets for the hand gesture classification.	24
2.3	Summary of literature methods for 3D reconstruction from monocular images.	29
3.1	Statistics of the MultiSFace dataset.	34
3.2	Statistics of the Watch-R-Patch dataset.	39
3.3	Statistics of the Baracca dataset.	42
4.1	Comparison of depth-based face datasets.	58
4.2	Training, validation and testing splits adopted for each dataset.	59
4.3	Depth-based face recognition, intra-dataset results (depth images).	61
4.4	Depth-based face recognition, intra-dataset results (point clouds).	62
4.5	Depth-based face recognition, intra-dataset results (voxels).	62
4.6	Depth-based face recognition, cross-dataset results (same sensor technology).	63
4.7	Depth-based face recognition, cross-dataset results (different sensor technology).	64
4.8	Depth-based face recognition, results on MultiSFace.	66
4.9	Depth-based face recognition, computational analysis of different methods.	67
4.10	Unimodal results on NVGestures.	78
4.11	Multimodal results on NVGestures.	79

4.12	Multimodal results on NVGestures, comparison with competitors.	80
4.13	Unimodal and multimodal results on Briareo.	82
4.14	Unimodal and multimodal results on Briareo, comparison with competitors.	83
4.15	Performance analysis of the proposed method.	84
4.16	Results on the Watch-R-Patch dataset.	88
4.17	Per-joint results on the Watch-R-Patch dataset.	89
4.18	Results of RefiNet and its single modules.	102
4.19	Comparison of RefiNet with literature approaches.	104
4.20	Performance analysis of RefiNet.	105
4.21	Results of the anthropometric measurement estimation on the depth domain.	109
4.22	Results of the anthropometric measurement estimation on the IR domain.	109
4.23	Results of the anthropometric measurement estimation on the RGB domain.	110
4.24	Results of the anthropometric measurement estimation on the thermal domain.	110
4.25	Results of the soft-biometrics estimation.	111
4.26	Computational evaluation of the proposed baselines.	111
4.27	Comparison with competitors in terms of 3D IoU on Pascal3D+.	121
4.28	Comparison with competitors in terms of Mask IoU and texture metrics on CUB.	123
4.29	Ablation study on the number of meanshapes on Pascal3D+.	125
4.30	Ablation study on different subdivision levels on Pascal3D+.	126

Chapter 1

Introduction

The automotive industry has witnessed stunning progress in recent years. Remarkable improvements have been made in passive and active safety, energy efficiency and vehicle performance, not to mention the rapid transition from traditional internal combustion engines to hybrid powertrains and fully electric vehicles. In the meantime, digital devices have spread in all aspects of everyday life and the price of their components has steadily decreased. At the same time, however, the human-vehicle interaction has seen very little changes. The most common interactions between the driver and the vehicle, such as accessing the car, adjusting the car configuration, interacting with the infotainment system and the car dashboard, has remained almost untouched in the last decade. Indeed, most of these interactions still require physical contacts between the user and the system. Even though new promising technologies are now available, such as touch interfaces, miniaturized cameras and voice commands, they have not been widely adopted in the automotive industry. Two main factors could explain the hesitancy in adopting these technologies. Firstly, automotive systems have to comply with strict regulations and requirements in order to be safe, robust and reliable. Secondly, the final product has to be compact, inexpensive and efficient in order to be viable and market profitable. Unfortunately, in this challenging context traditional human-vehicle interaction means are still preferred over technologies that could revolutionize the way we interact with our cars.

In this thesis, we aim at improving the current human-vehicle interaction

systems by investigating a new vision-based approach to the field. In particular, we study computer vision techniques and solutions that could replace or integrate existing means of human-computer interaction and form the foundation of the next generation of in-vehicle infotainment systems. We direct our attention to vision systems based on both visible light and non-visible spectrum, leveraging on the recent introduction of affordable and compact range and infrared cameras. Indeed, needing reliable sources of data under different illumination conditions, ranging from daylight to night-time, we focus on infrared-based devices such as depth and thermal sensors, rather than standard RGB cameras.

Given the scarcity of data collections containing depth maps and infrared images, we have collected several novel datasets throughout the PhD. In particular, we collected and publicly released four datasets, covering the face recognition, the recognition of dynamic hand gestures, the human pose estimation and the contact-less estimation of anthropometric measurements. At the same time, we embrace recent deep learning approaches and focus on efficient neural networks that can seamlessly run on embedded boards. We use them to study novel solutions for several computer vision tasks, aiming at covering the majority of the human-car interactions. To this end, we investigate the usage of range cameras and depth maps for the face recognition task, showing how depth map representations and deep neural models affect the recognition performance. Moreover, we address the problem of real-time recognition of dynamic hand gestures to control the car infotainment. We propose a novel architecture that obtains state-of-the-art results using non-RGB cameras, *i.e.* depth and infrared sensors. We also focus on the analysis of the human body as a whole, from two different perspectives. On the one hand, we study new approaches for the 2D and 3D human pose estimation and refinement from depth maps. On the other hand, we investigate the contact-free estimation of anthropometric measurements using color, depth, infrared and thermal cameras. Finally, we shift our attention to the external area surrounding the vehicle and present a novel method for the 3D reconstruction of objects from 2D images. We consider this work as a first step towards a virtual 3D representation of the vehicle surroundings, which could be particularly useful when replacing rear and side mirrors with cameras and digital screens.

A considerable part of the PhD has been carried out within the RedVision Laboratory, a collaboration between AImageLab, the computer vision lab of the University of Modena and Reggio Emilia, and Ferrari S.p.A.,

with the goal of improving the interaction between the car and its users.

The rest of the thesis is organized as follows. Chapter 2 describes the background to the topics discussed in this thesis. Firstly, we present an overview of sensors which capture data from the non-visible spectrum, *i.e.* depth, infrared and thermal cameras, and an overview of different depth map representations. Then, we report a brief survey of literature approaches related to the addressed topics. In Chapter 3, we introduce the datasets used in this thesis, starting with the ones collected during the PhD and concluding with public datasets that we used to compare with the literature. Chapter 4 is the main core of the thesis, including the proposed methods and their experimental evaluation. We firstly present a thorough analysis of depth map representations and methods for the depth-base face recognition, followed by a transformer-based method for the dynamic hand gesture recognition in automotive. Then, we report deep methods for the 2D human pose estimation and its 3D refinement using depth maps and a set of baselines to regress anthropometric measurements from several data modalities. Finally, we propose a novel architecture that reconstruct the 3D shape, pose and appearance of objects from a single 2D image. Chapter 5 draws conclusions and future works for the approaches presented in the previous chapters.

The list of published articles is reported in Appendix A, while activities carried out during the PhD are listed in Appendix B.

Summary of Contributions

In the following, we briefly report the thesis original contributions and highlight the author’s personal contributions to each work. We group them according to four main areas related to the human-vehicle interaction. We refer the reader to Chapter 4 for a broader discussion of these areas.

Identification

Given that accessing the vehicle is among the first interactions between a person and a car, we focus our research on the face recognition, which is a natural, object-less and contact-less method to identify a person. Taking into account the constraints of the automotive context, we investigate the usage of depth cameras for the face recognition under different acquisition

scenarios. To overcome the limitation of the existing public datasets, we collect and publicly release MultiSFace, a novel dataset that contains recordings of human faces with multiple synchronized cameras, having different resolution and quality, and under different acquisition settings (Section 3.1.1). Moreover, we present an extensive investigation of several depth map representations and their effects on the recognition accuracy and on the generalization of the trained deep models (Section 4.1).

The author’s main contributions in this area are: the design and acquisition of the dataset; the design and implementation of the experimental section; the analysis of the results.

Gestures

Interactions involving the in-vehicle infotainment system are one of the most common among the car users. In this thesis, we propose a novel vision-based gesture recognition system to control the infotainment system by means of contact-less interactions. In particular, we collect and publicly release a new dataset, called Briareo, of dynamic hand gestures that are specifically designed for the control of an infotainment system (Section 3.1.2). Moreover, we present a novel deep architecture for the real-time recognition of dynamic hand gestures, recorded with depth, infrared or RGB cameras (Section 4.2). The proposed model obtains state-of-the-art results on two public datasets.

The author’s main contributions in this area are: the design of the dataset structure, the transformer-based neural network, and the experimental evaluation; the analysis of the results.

Posture and anthropometry

The estimation of the human pose and several anthropometric measurements could improve the human-vehicle interaction, *e.g.* enabling the analysis of the driver posture and the automatic adjustment of the car settings. Therefore, we investigate both the 2D/3D human pose estimation and the estimation of anthropometric measurements from visual data. Regarding the former, we present Watch-R-Patch, a novel dataset containing refined joint annotations for an existing depth dataset (Section 3.1.3). Moreover, we present a depth-based 2D pose estimator and a novel refinement framework, named RefiNet, that predicts an accurate 3D human pose given a 2D pose and a depth map, obtaining promising results (Section 4.3). Compared to

other approaches, our refinement method predicts 3D human joints in the absolute camera coordinate system, rather than in relative coordinates. In addition, the approach is modular and can be adapted according to the available computational power. Regarding the latter, we collect and release Baracca, a dataset of anthropometric measurements and images acquired with different cameras, *i.e.* RGB, IR, depth, thermal (Section 3.1.4). To the best of our knowledge, there are no publicly available datasets of this kind. Using Baracca, we investigate the efficacy of several estimation techniques, ranging from geometrical approaches to machine and deep learning methods (Section 4.4). Our analysis demonstrates that anthropometric measurements can be successfully estimated from a wide range of different visual data. This vision-based approach is cheaper, faster and less invasive than existing methods based on manual measurements or 3D scanners, enabling its use in the automotive context.

The author’s main contributions in this area are: the analysis of the existing dataset limitations; the design of the modules of RefiNet and of their experimental evaluation; the design and collection of the anthropometric dataset; the implementation of the anthropometric baselines; the analysis of the results.

Digital mirrors

Given the recent trend of replacing traditional mirrors with digital counterparts, we investigate the disentanglement of the scene shown by the digital mirrors from the location of the external cameras, aiming at providing a user-controllable view of the car surroundings. To this end, we present a semi-supervised approach that performs the 3D reconstruction of vehicles of different categories from a single 2D image (Section 4.5). The proposed method is trained on in-the-wild images of objects belonging to different categories and does not use direct 3D supervision. To the best of our knowledge, the proposed method is the first that can handle multiple object categories during both training and inference. Moreover, we present a novel module that produces naturally smooth shapes thanks to its design.

The author’s main contributions in this area, equally shared with the co-author Alessandro Simoni, are: the extensive analysis of the literature; the design and implementation of the proposed method, the experimental evaluation and the ablation studies; the analysis of the results.

Chapter 2

Background

In this chapter, we present the background to the topics presented in this thesis, analyzing the machine vision sensors that are suitable for the automotive context and reviewing the literature that covers several tasks related to the human-vehicle interaction. On the one hand, we introduce the depth sensors and the related technologies that are inherent to this thesis, *i.e.* stereo, Structured-Light and Time-of-Flight cameras. We also provide a formal definition of depth maps and their most common representations. Then, we briefly introduce the near-infrared and thermal cameras, focusing on their types and properties. On the other hand, we propose a literature survey over several topics, aiming at covering most of the human-vehicle interactions. In particular, we focus on face recognition, dynamic hand gesture recognition, human pose estimation and refinement, contact-less anthropometric measurement estimation and 3D object reconstruction from monocular images.

The rest of the chapter is organized as follows. Section 2.1 contains an analysis of infrared-based vision sensors, *i.e.* depth sensors and thermal cameras. The most common representations for the data acquired by these sensors are also illustrated. Section 2.2 presents the literature survey, covering several topics that will be addressed in the following chapters.

2.1 Vision Beyond the Visible Spectrum

The Computer Vision community is mainly focused on standard RGB images and videos. As a result, the majority of literature methods and publicly available datasets are biased towards RGB data. Even though color cameras mimic the human visual system, several fields need computer vision techniques to process other kinds of data. While some sectors make inherently use of non-RGB data, *e.g.* the medical imaging and in specific industrial applications, others mainly use RGB data due to their ubiquitousness, but they could benefit from using other types of data.

During the PhD, we have investigated several different imaging systems that can benefit the human-vehicle interaction, when used in addition to or as a replacement of RGB cameras. In particular, we looked for the ideal sensor suite for computer vision-based human-vehicle interaction systems, taking into consideration three major requirements that the automotive scenario poses. The first requirement is the *illumination invariance*. That is, vision-based human-computer interaction systems have to be invariant to the illumination conditions and be able to work even in the dark or during severe lighting changes. These conditions are very common in automotive: shadows, tunnels, variable weather conditions, night-time driving and many other situations affect the illumination of the car cockpit. The second requirement is the *non-invasiveness*. To avoid interfering with the driving activity, human-vehicle interaction systems have not to obstruct the driver's movement nor reduce their visibility. Moreover, they should not cause the driver's distraction. Last but not least, the third requirement is the *low latency*. Indeed, the system has to be reactive and the interaction with the driver should be natural and smooth. As such, the vision system should have negligible latency or, in other words, guarantee a high frame rate.

Taking the aforementioned requirements into account, we identified two main types of vision sensors that are suitable in automotive when taking the human-vehicle interaction into account. They are the depth sensors, also known as range cameras, and the infrared cameras. In this section, we briefly describe them. In details, we firstly present the main depth sensor technologies, with a particular focus on their differences. Then, we formally define the data format known as depth map and the most common depth map representations, *i.e.* depth images, surface normals, voxels and point clouds. Finally, we briefly describe the near-infrared and thermal cameras, focusing on their properties and common formats of the recorded data.

2.1.1 Depth Sensors

In contrast with common cameras, which measure the intensity of the light emitted or refracted from the framed objects, depth sensors measure the distance of the objects from the camera itself. In particular, they record the distance of each point from the camera image plane and usually provide the acquired data in millimeters, saved in one-channel 16-bit images. To obtain this type of measurement, there exists several different technologies, widely studied in literature [58, 73, 174, 185, 81, 62, 179]. Among them, we review those that can provide good-quality data and can be implemented in affordable, inexpensive and compact sensors, *i.e.* stereo cameras, Structured-Light sensors and Time-of-Flight cameras. LiDARs and 3D laser scanners are out of the scope of this thesis.

Stereo vision

Cameras based on stereo vision make use of two sensors in order to simulate the human binocular vision and capture 3D data. The distance between the sensors can be adapted depending on the application and the camera operating range. Given two images of the same scene acquired by two sensors that are split by a known distance, and assuming the sensors to approximate the pinhole camera model, it is possible to recover the 3D information of the scene, *i.e.* range data, using the principles of epipolar geometry. However, epipolar geometry need corresponding points in the two images. Thus, it is necessary to find sparse or dense correspondences to recover depth values. This task is known as correspondence problem and it has been widely studied in the last decades [117, 72, 111]. The main limitation of stereo matching algorithms, and thus of stereo cameras, is that the correspondences are hard to find in weakly textured areas or when there are repetitive patterns or occlusions.

Most of the stereo cameras are passive sensors: they use two color or infrared cameras that react to natural light hitting the imaging sensors. Recently, active stereo cameras have been presented, *e.g.* the Intel RealSense family, to lessen the issues of standard stereo vision. These sensors usually make use of an infrared illuminator or projector to ease the correspondence problem. However, the depth estimation and the camera speed are still dependent on the performance of the stereo matching algorithm, limiting their usage in applications requiring high-quality depth maps. On the other

hand, stereo cameras usually have a high resolution and a wide operating range, being able to estimate depth up to dozens of meters.

In the activities presented in this thesis, we make use of an active infrared stereo camera: the Leap Motion controller. It contains two wide-angle lens sensors and three infrared illuminators in a compact form factor. The dense stereo matching result is not computed nor returned by the device, which has a limited operating range, since it is specifically designed to compute accurate 3D hand joint detection and tracking, rather than depth estimation.

Structured light

A Structured-Light camera is composed of a camera and a texture projector. The projector illuminates the scene with a known pattern, usually corresponding to vertical or horizontal lines or to a matrix of dots, and the sensor captures the pattern reflected by the objects. Analyzing how the pattern is deformed when reflected back by the illuminated objects, it is possible to geometrically compute the depth map of the scene with high accuracy. Usually, both the camera and the projector use the near-infrared light to avoid interfering with RGB cameras and to be invisible to the human eye. Moreover, different projectors or patterns can be used simultaneously or sequentially to improve the depth estimation and the surface reconstruction. Other devices make use of multiple cameras, in combination with one or more illuminators, merging the Structured-Light technology with the stereo vision approach.

Structured-Light sensors are recommended in indoor environments, given their high accuracy and relatively high spatial resolution. On the other hand, they are not suitable for outdoor usage since they suffer from interference from other infrared lights, such as the sunlight. Moreover, multiple sensors may interfere to each other when used simultaneously, limiting their application in multi-camera settings. In addition, the continuous projection of patterns results in a high energy consumption compared to stereo cameras and Time-of-Flight sensors, hindering their usage on battery-powered devices or when the available energy is limited.

In the activities presented in this thesis, we do not make use of any structured-light sensor, preferring the usage of Time-of-Flight cameras.

Time-of-Flight

A Time-of-Flight, or simply ToF, camera is a sensor that measures depth by measuring the round trip time that light takes to travel from an emitter to the framed object then back to the sensor. It is composed of an infrared emitter, that illuminates the entire scene, and a receiver, that records the light reflected by objects. Both the components work on a specific wavelength in the infrared band. Compared to LiDARs, that usually make use of a point-wise laser beam and require a scanning operation and moving parts, ToF cameras are scannerless: they capture the entire scene with a single light pulse and do not require any moving parts. Combined with a simple processing pipeline, this fact enables a quick acquisition and a high frame rate.

Time-of-Flight cameras use one of two different methods to measure light's time of flight and thus the object distance. One approach is based on pulsed-light and the direct measurements of the round-trip time. This type of sensor can emit very powerful light in a fraction of a millisecond, enabling it to work outdoors and to capture data up to tens of meters. However, the emission of powerful light is the cause of high energy consumption and the pulse approach limits the maximum frame rate. The second approach is based on Continuous-Wave modulation and the measurement of the phase difference between the emitted sinusoidal light and the reflected one captured by the receiver. Compared to the previous method, the light emission is continuous, but less powerful, requiring less energy and less powerful emitters. On the other hand, the maximum detectable distance is limited to few meters by the less powerful light and by the distance ambiguity occurring when the continuous sinusoidal signal wraps. This type of sensor is suited for indoor usage, but recent devices can work in outdoor environments too, if they are not directly hit by sunlight.

In the activities presented in this thesis, we make use of several Time-of-Flight cameras. In particular, we employ the well-known Microsoft Kinect v2, the compact Pico Zense DCAM710 and the miniaturized Pmdtec CamBoard Pico Flexx. These sensors are Continuous-Wave modulated. Excluding the Kinect, they are portable and USB-powered and they have multiple working configurations, letting the user to adjust the acquisition range and frame rate based on their needs.

2.1.2 Depth Map Representations

Depth sensors provide data in several formats, which can be represented as depth maps. Formally, a depth map can be defined as $D_M = \langle \mathbf{D}, \mathbf{K} \rangle$, where $\mathbf{D} = \{d_{ij}\}$, with $d_{ij} \in [0, R]$, is a matrix of distance values between 0 and the maximum measurable range R , and \mathbf{K} is the perspective projection matrix that is obtained with the intrinsic parameters of the sensor. More specifically, d_{ij} is the distance between the optical center and a plane parallel to the image plane containing the physical point or, in other words, the distance between the image plane and the physical point. The 3D coordinates of each captured point can be recovered from \mathbf{D} and \mathbf{K} , then used to compute point clouds and voxels. Most of the computer vision algorithms do not directly exploit D_M as input, but they convert D_M in depth images, surface normals, voxels or point clouds, as described in the following paragraphs.

Depth maps as depth images

The depth image, also referred to as range or 2.5D image, is the most used representation of range data and can be considered a re-quantization of the \mathbf{D} distance matrix. A depth image I_D is encoded as a one-channel gray-scale image, in which the intensity of each pixel represents the quantized version of d_{ij} . Spatial resolution, depth precision and data format strictly depend on the acquisition device. Frequently, 8-bit gray-scale image formats are used to increase the compatibility and facilitate the viewing. Consequently, the computed depth image loses the full 3D content of the original depth map in exchange for a 2D representation, which is easier to manage.

A huge number of works combine the use of Convolutional Neural Networks (CNNs) and depth images, used as standard intensity images, in a variety of tasks [189, 244, 188, 16, 19, 149, 235]. The major drawback of using depth images is that their visual appearance is not device-invariant: they depend on the sensor lens and the camera intrinsic parameters. Moreover, it presents variations based on the sensor technology and the acquisition setup. In addition, pre-processing steps, which are usually employed with intensity images, could partially or completely remove the metric depth data and destroy the 3D consistency.

Depth maps as surface normals

A complementary depth map representation that aims at reducing the aforementioned issues is the surface normals, also called normal image. We define a normal image as a matrix of pixels with three channels $\hat{I}_N = \{\hat{\mathbf{v}}_{ij} = \langle \hat{v}_x, \hat{v}_y, \hat{v}_z \rangle\}$, where each pixel encodes the (x, y, z) components of the estimated surface normal vector in that point. In this thesis, we follow Besl and Jain [11] to obtain an estimation of surface normals starting from depth images. Specifically, given the depth matrix \mathbf{D} and defining $Z(x, y)$ as its pixel value in (x, y) , the direction $\mathbf{d} = \langle d_x, d_y, d_z \rangle$ of the surface normal is defined as:

$$\mathbf{d} = \left(-\frac{\partial Z(x, y)}{\partial x}, -\frac{\partial Z(x, y)}{\partial y}, 1 \right) \quad (2.1)$$

where $\partial Z(x, y)/\partial x$, $\partial Z(x, y)/\partial y$ are the gradients obtained on the depth in the x and y directions [150]. These directions can be calculated as:

$$\begin{aligned} \frac{\partial Z(x, y)}{\partial x} &\approx Z(x + 1, y) - Z(x, y) \\ \frac{\partial Z(x, y)}{\partial y} &\approx Z(x, y + 1) - Z(x, y) \end{aligned} \quad (2.2)$$

In practice, this operation can also be implemented with a Sobel filter on the x and y direction. Finally, a normalization step [9] is applied to obtain unit-magnitude normal vectors $\hat{\mathbf{v}}$:

$$\hat{\mathbf{v}} = \frac{1}{B} \langle d_x, d_y, 1 \rangle, \quad B = \sqrt{d_x^2 + d_y^2 + 1} \quad (2.3)$$

It is worth noting that only few literature works exploit normal images directly obtained from depth maps.

Depth maps as point clouds

Depth maps can be converted into a 3D point cloud whose coordinates are defined in the camera reference frame. Formally, a point cloud can be represented as an unordered set of points $P = \{p_k = \langle p_{k_x}, p_{k_y}, p_{k_z} \rangle\}$, where a generic point p_k is a vector containing its 3D coordinates [171].

The conversion from the depth map to the point cloud can be defined as

$$p_{k_x} = (x_i - c_x) \cdot \frac{Z(x_i, y_j)}{f_x} \quad (2.4a)$$

$$p_{k_y} = (y_j - c_y) \cdot \frac{Z(x_i, y_j)}{f_y} \quad (2.4b)$$

$$p_{k_z} = Z(x_i, y_j) \quad (2.4c)$$

where the 3D point $p_k = \langle p_{k_x}, p_{k_y}, p_{k_z} \rangle$ corresponds to the value that is sampled over the depth map at a generic location (x_i, y_j) and the constants f_x, f_y, c_x, c_y are the elements that define the camera intrinsic parameters \mathbf{K} (assuming that the pixels of the sensors are squared, *i.e.* having skew $s = 0$). In practice, the majority of the depth sensors, in particular the ones based on active illumination (*e.g.* Microsoft Kinect, Pico Zense), directly provides the 3D point cloud in addition to the depth maps.

It is worth noting that depth maps contain only 2.5D information. Thus, the extracted point cloud contains partial 3D information, *i.e.* a single view of the 3D scene. Moreover, since point clouds are unordered, with a variable length and sparse in the 3D space, they are more difficult to process with deep neural networks.

Depth maps as voxels

A voxel is a point-wise three-dimensional volumetric representation, the 3D equivalent of a 2D pixel in standard intensity images [98].

In the literature, the term voxel is also used to represent a 3D volume that is defined as three-dimensional matrix $V^m = \{v_{ijh}, i, j, h = 1, \dots, m\}$, where m is the number of elements for each side of the 3D cube and each element $v_{ijh} \in \{0, 1\}$ is a binary value, with 0 representing an empty space and 1 an occupied one.

A 3D point cloud P can be converted in a voxel V^m with the following procedure. Defining a 3D cube with side length L centered in $p_c = (p_{c_x}, p_{c_y}, p_{c_z})$ (which usually corresponds to the center of the point cloud) and the number m of binary voxels for each side of the cube, the 3D volume is split into $m \times m \times m$ binary elements of side $l = \frac{L}{m}$. Each binary element v_{ijh} represents the presence of at least one point lying inside its

corresponding 3D volume s_{ijh} of side l :

$$v_{ijh} = \begin{cases} 1 & \exists p_k \in P \mid p_k \in s_{ijh} \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$

In other words, $v_{ijh} \in V^m$ is a binary value that indicates whether at least one point of the point cloud lies in the 3D volume s_{ijh} corresponding to its cell. Unlike voxels computed from 3D models, which report the whole volume of the 3D object as occupied, only the voxels that correspond to the external visible surface of the object are identified as occupied, *i.e.* only the 3D data that the depth sensor is able to acquire.

At the time of writing, only few works propose analyzing voxels obtained from depth maps with deep approaches.

2.1.3 Near-Infrared and Thermal Cameras

Infrared cameras are a group of sensors that are able to acquire non-visible light, *i.e.* electromagnetic radiation, in the infrared wavelength range (0.7 μm - 1 mm) [60]. They are usually classified according to the infrared range that their sensors detect, following the division of the infrared spectrum in five bands, as reported in Table 2.1. In the works presented in this thesis, we used Near-Infrared cameras, by means of infrared amplitude recorded by depth sensors, and both radiometric and non-radiometric LWIR thermal cameras.

Each band has different properties and applications, which are discussed in the following. Near Infrared (NIR) cameras record images that are very similar to gray-level images. As the standard RGB cameras, this type of sensor receives ambient light which is reflected by the framed objects. For this reason, some NIR acquisition devices are equipped with an infrared illuminator, in order to improve the visibility of the scene without interfering with other RGB cameras nor being visible to the human eye. Moreover, active depth and stereo cameras that make use of an infrared emitter usually provide the infrared amplitude recorded by the sensor, that roughly corresponds to a NIR image. Moving towards longer infrared wavelengths, Short-Wave Infrared (SWIR) cameras record an intermediate representation of the scene, which is characterized by capturing both reflected light and emitted thermal radiation (in the higher band). This type of sensor is seldom used in machine vision applications. The following group of sensors, *i.e.*

Table 2.1: Infrared wavelength bands.

Band name	Band acronym	Wavelength (μm)
Near Infrared	NIR	0.7 – 1.4
Short-Wave Infrared	SWIR	1.4 – 3
Mid-Wave Infrared	MWIR	3 – 8
Long-Wave Infrared	LWIR	8 – 15
Far Infrared	FIR	15 – 1000

the Mid-Wave Infrared (MWIR) and Long-Wave Infrared (LWIR) cameras, are fully passive thermal sensors that record only the thermal radiation emitted by the objects according to their temperature, in terms of both wavelength and intensity. These wavebands are the most used by thermal cameras and can be further split in two different types, as detailed in the following section.

Thermal radiometry

Thermal cameras that operate in the MWIR and LWIR range can be further classified in radiometric and non-radiometric. Radiometric thermal cameras measure the exact temperature of the framed objects and return it in kelvin. Thus, each pixel in the thermal image can be converted into a precise temperature value. However, a reference point at a known temperature has to be within the camera field of view to compute accurate temperature measurements. In fact, measured temperature values can be affected by natural factors and by the temperature of the sensor. To take these effects into account, a visible point at a known temperature is used to rescale the recorded values to the correct temperature range. On the other hand, non-radiometric thermal cameras are not capable of recording the temperature of the framed object. While the acquired images still show proper relative temperatures (showing warmer objects with a lighter color compared to colder objects), it is not possible to retrieve their absolute temperature. Thus, they cannot be used when the absolute temperature of an object is needed, such as in vision systems that measure the human body temperature.

Data representation

NIR and SWIR cameras usually provide single-channel gray-level images which are similar to gray-scale intensity images. In some cases, in particular for NIR images provided by active depth sensors, images are encoded using 16 bits per pixel to increase the intensity resolution.

MWIR and LWIR cameras usually provide single-channel 8-bit or 16-bit images containing normalized values (according to the colder and warmer framed object) or RAW data. The latter can be converted to the absolute object temperature in kelvin (K). Another popular representation of thermal images converts the single-channel images into human-readable RGB images by applying a color palette or a color-conversion function.

2.2 Literature Survey

In this section, we present an overview of public methods and datasets that are related to the topics discussed in this thesis. In particular, the survey firstly contains a summary of face recognition algorithms, followed by an analysis of methods that tackle the recognition of dynamic hand gestures. Then, the attention is shifted to the human body, presenting methods for the human pose estimation and datasets for the estimation of anthropometric measurements. To conclude, approaches to reconstruct the 3D shape of objects from monocular images are discussed.

All but the last section are focused on methods that make use of depth maps or infrared data, rather than intensity images. As discussed in Section 2.1, these sensors are robust and illumination-invariant, and provide complementary data with respect to RGB cameras. Thus, they are suited to be used in the automotive context.

2.2.1 Face Recognition

Face recognition is a widely studied task in the computer vision community. While most of the literature is focused on RGB images, interest in depth cameras and depth maps has steadily grown in recent years. In the following, we report a brief overview on face recognition methods that rely on intensity (RGB) images. Then, we present a thorough survey of methods that rely on depth data, classifying them according to the used depth map representation (see Section 2.1.2 for a discussion on different depth map representations).

While methods based on intensity images generalize well on a wide range of hardware and image quality, the literature lacks a clear understanding of whether depth map-based methods can scale well to different sensors and acquisition settings, and of the effects that the depth map representation has on the model generalization. Indeed, public depth dataset are often collected using a single depth sensor in limited acquisition settings and the approaches that use them train and evaluate the proposed recognition methods on the same depth dataset. Thus, they do not evaluate the model generalization to data obtained by other cameras or in other settings.

RGB-based face recognition

The majority of face recognition approaches is based on RGB images. In the last decades, a vast body of literature has focused on algorithms tackling

the classification of hand-crafted features [6, 89, 100, 211, 248, 3, 93].

Recently, impressive results have been achieved in the RGB domain [202, 187, 200, 129, 126, 42], thanks to the adoption of deep neural networks [77, 201, 194] and very-large datasets [25, 70, 85]. For instance, Taigman et al. [202] presented DeepFace, a deep convolutional network designed for the verification task. A Siamese architecture is proposed and used in conjunction with pre-processing steps, such as face alignment and face frontalization.

A well-established line of research consists in incorporating a margin in loss functions. The pioneering work of Schroff et al. [187] proposed the use of a Triplet Loss to learn a face embedding space where faces that belong to the same identity are clustered together and far from other clusters. Recently, Deng et al. [42] proposed the use of an additive angular margin loss, called ArcFace, to learn highly discriminative features for face recognition. Extensive experimental evaluations on ten face recognition benchmarks based on intensity images showed state-of-the-art performance. Similarly, a learned Cluster-based Large Margin Local Embedding and a k-nearest cluster algorithm are combined obtaining significant improvements over existing methods on both face recognition and face attribute prediction in the work of Huang et al. [84]. Zhao et al. [251] proposed to distance the representations of the identities through an exclusive regularization to obtain more discriminative features.

Depth map-based face recognition

Depth sensors provide data in several formats, which can be represented as depth maps. In a depth map, each pixel corresponds to the 3D distance between the camera image plane and the physical point. Being an intermediate representation between 2D and 3D data, depth maps can be represented in different formats, such as depth images, surface normals, voxels or point clouds. Most of the computer vision algorithms use depth maps encoded using one of these representations.

In the following, we present face recognition methods that make use of depth maps, in one of their representations. For a formal definition of depth maps and their representations, we refer the reader to Section 2.1.1.

Depth maps as depth images. The depth image, also referred to as range or 2.5D image, is the most used representation of range data and it is usually encoded as a one-channel gray-scale image.

While traditional approaches [134, 135] formulate the task as the extraction and classification of hand-crafted features from the depth images, deep learning-based methods combine the use of Convolutional Neural Networks (CNNs) and depth images as standard intensity images addressing a variety of tasks, including face recognition. Some literature works [2, 16] combine depth images with standard CNNs to regress the 3D head pose, while many others [153, 19, 20, 149, 83] apply the same strategy for the face recognition. Neto and Marana [153] propose the use of a CNN for the face recognition task, based on low-level 3D local features (3DLBP) extracted from depth maps. In the work of Mu et al. [149], several pre-processing steps are applied on facial depth images, including hole filling (to reduce the areas with invalid depth values), depth range normalization (based on the nose tip detection) and outlier removal. Then, a 2D CNN is used as discriminative feature extractor. Hu et al. [83] present a method for boosting depth-based face recognition through the combined use of high-quality depth data that were acquired by a 3D scanner and depth images. In the work of Borghi et al. [19], a Siamese network that processes pairs of facial depth images is proposed without exploiting any specific image pre-processing algorithms. The approach exploits RGB images during training as Privileged Information (also called Side Information) while using only depth images at inference. Some pre-processing methods for depth images have also been proposed [118], including nose tip detection for face crop and head pose correction.

Other works use depth images in combination with other types of data that are obtained from depth or RGB-D devices, like intensity images [244, 17, 235] or human body joints [188, 189]. In particular, several face recognition methods [184, 114, 124] combine the use of both RGB and depth data, assuming the presence of both type of data at training and testing time, to compensate for the relatively low resolution of depth maps and the lack of texture information. For instance, Lee et al. [114] proposes a pipeline consisting in depth image recovery, feature extraction through a deep learning-based approach and joint classification to recognize faces based on both color and depth information. In addition, most of these methods are based on facial landmark detection to perform face alignment and frontalization and on a supplementary classifier to perform a joint classification of multi-modal features.

In general, the aforementioned methods are not able to generalize well on unseen domains and sensors. In fact, the visual appearance of depth images

is not device-invariant and it is strictly related to the sensor technology and the acquisition setup. Moreover, pre-processing steps, which are useful on intensity images, could partially or completely remove the metric depth information and destroy the 3D consistency.

Depth maps as surface normals. A complementary depth map representation that aims at reducing these issues is the surface normals, also called normal image. In a normal image, each pixel represents the direction of the estimated surface normal vector in that point. A rough estimation of the surface normals can be easily computed from a depth image (see Section 2.1.2). However, few works [91, 239, 149] use normal images obtained from depth maps for face recognition. Mu et al. [149] exploit the discriminative content of normal images [91, 239], in combination with the depth images, in a face recognition framework.

Depth maps as point clouds. Depth maps can be converted into the corresponding 3D point cloud with coordinates that are defined on the camera reference frame.

Point clouds are unordered, variable in length and sparse in the 3D space. Thus, it is more complex to use them as input of deep networks. Moreover, since depth maps only contain 2.5D information, the extracted point cloud contains partial 3D information, *i.e.* a single view of the 3D scene. As a result, to the best of our knowledge, no works propose using point clouds directly obtained from depth maps for the face recognition task. Some literature works exploit 3D facial scans to build facial 3D models or convert them to one or more depth maps. For instance, Kim et al. [101] propose a transfer learning technique in order to train a CNN on 2D face images and to test it on 3D facial scans, represented as frontalized depth images, after a fine-tuning phase with a limited number of point clouds.

On the other hand, point clouds are adopted for the 3D object recognition task, often on synthetic datasets, as in the work of Qi et al. [171]. The proposed network, called PointNet, is directly fed with unordered 3D point sets and it is robust to input rotation, corruption and perturbation. Its evolution, named PointNet⁺⁺ [172], consists in a recursive use of the PointNet model on subsets of neighboring points and it is able to learn local features with increasing contextual scale. Similarly, recent works [104, 224] propose to increase the model capacity stacking several PointNets hierarchically. Other works [205, 232] propose the use of local convolutions on point clouds. Still, the accuracy improvement with respect to earlier

work is limited. It is worth noting that deep learning-based models that deal with point clouds are often computationally inefficient [223, 128] and they require a great amount of memory. Only recently, Wang et al. [223] investigated how to reduce the memory consumption and inference time.

Depth maps as voxels. A voxel is a point-wise three-dimensional volumetric representation, corresponding to the 3D equivalent of a pixel in intensity images [98]. In literature, the same term is used to refer to a 3D volume grid, where each element is a binary value representing whether that specific space is occupied.

Currently, only few works propose analyzing voxels obtained from depth maps with deep approaches. Moon et al. [146] proposed the use of a specific 3D CNN, called V2V-PoseNet, to tackle hand and human pose estimation. A voxel-to-voxel architecture is developed to predict 3D heatmaps, from which 3D coordinates of hand keypoints or human body joints are obtained. A considerable number of methods are based on voxels obtained from 3D scanners or LiDARs. For instance, in the work of Maturana and Scherer [141] voxels are the input of a supervised 3D CNN for the object detection task. The experimental results are collected processing voxels that were obtained from 3D scanners. Zhou and Tuzel [253] propose VoxelNet, a generic detection network able to work with voxels obtained from LiDAR data. Recently, Riegler et al. [178] propose partitioning sparse 3D data through a set of unbalanced octrees, in which each leaf node stores a pooled feature representation.

In general, the use of voxels, together with deep learning models, is limited, since a reference point, *i.e.* the point around which the 3D space (usually a 3D cube) is sampled, is needed. Furthermore, it is necessary to define the volume of 3D space around the reference point and the size of the single voxels, *i.e.* the level of quantization. All of these elements deeply influence the final performance of systems based on voxels used as input data [146, 112].

2.2.2 Hand Gesture Recognition

In this thesis, we focus on dynamic hand gestures, composed of a combination of hand poses and motion. Differently from static hand gestures, dynamic hand gestures require recognition systems to take into account the temporal dimension, resulting in a higher level of complexity. As such, existing approaches that can run in real time still do not reach satisfactory

performance on the depth domain. Moreover, the majority of the public depth datasets are limited in size, do not contain gestures related to the infotainment system or were recorded with Structured-Light depth sensors rather than Time-of-Flight cameras.

In the following, we present an overview of hand gesture recognition articles from two points of view: recognition methods and gesture datasets.

Methods

In the literature, the dynamic hand gesture recognition task has been approached using different strategies which enable the temporal observation of an action performed by a human. Recent architectures [206, 144, 29], which exploit the potential of 3D Convolution in extracting temporal features from videos, have become the foundation of most of the action recognition systems. As many tasks in the computer vision field, the hand gesture recognition task can rely on different types and combination of input data. Therefore, from a general point of view, methods available in the literature can be grouped as unimodal and multimodal.

In the unimodal case, a single input (*e.g.* RGB, depth) is used at a time. Köpüklü et al. [106] adapt state-of-the-art architectures, *i.e.* C3D [206] and ResNet [77], in a lightweight framework composed of a detector, that detects the beginning and the end of a gesture, and the gesture classifier. Since 3D CNNs needs more training data due to the larger number of parameters with respect to 2D CNNs, the networks are pre-trained on one of the largest public hand gesture datasets, called Jester [140], and then fine-tuned on other datasets. De Smedt et al. [39] exploit 3D hand joints to reconstruct the hand skeleton and then perform the gesture classification capturing the motion and the hand shape through a video sequence. Unfortunately, results are unsatisfactory on datasets that do not contain high-quality hand skeleton annotations. With the recent success of the self-attention mechanism [215], an attention-based network has been introduced by Dhingra and Kunz [46]. They use a 3D CNN model in which 3 attention blocks are positioned between the residual modules in order to learn features at different scales. Since they train their network from scratch, they obtain good results only on datasets with a large amount of training data.

In the multimodal setting, two or more data types form the input of the recognition method. Narayana et al. [152] propose a novel architecture that exploits 4 different data types (RGB and depth data, along with their

computed optical flow) to analyze the body motion. The network uses a spatial focus attention mechanism that restricts the focus on specific body parts (*e.g.* global, right hand, left hand). Having a total number of 12 features channels, they face the problem of gesture classification weighting each channel with respect to its importance to a specific gesture. A different multimodal approach has been introduced by Kopuklu et al. [105]. In this case, they apply a data level fusion between an RGB frame and several optical flow images computed on the previous frames. They are given as input to a deep network that extracts spatio-temporal features and classify the gesture with a fully connected network. An inspiring work by Abavisani et al. [1] presents a method exploring the performance of multimodal training and its effects on unimodal testing. The authors fine-tune a pre-trained 3D CNN network [29] on multiple source data (*e.g.* RGB, depth, optical flow) and introduce a loss, called spatio-temporal semantic alignment, which encourages the network to learn a common understanding of the different data types.

Authors of another set of works [64, 107, 38] propose transformer-based approaches to tackle the action and the sign language recognition tasks. Girdhar et al. [64] propose a slightly modified version of the transformer architecture as part of an action localization and recognition framework, resembling the structure of Faster R-CNN. In the work of Kozlov et al. [107], a transformer-like architecture is used in combination with a feature extractor for real-time action recognition. It makes use of 1D convolutional layers between sequential decoder blocks, but it does not use any kind of positional encoding thus the temporal relationships are not explicitly modeled. This method is not developed for the usage with depth sensors and does not propose the usage of surface normals as a different depth map representation.

Datasets

Recently, several datasets addressing the driver gesture classification have been publicly presented [136, 156, 144]. These datasets propose various gesture classes, performed by multiple subjects, with diverse gesture complexity and acquisition sensors. A summary of these datasets is reported in Table 2.2.

The dataset proposed in the work of Marin et al. [136] contains both 3D hand joint location and depth maps, acquired jointly with a Leap Motion

Table 2.2: Datasets for the hand gesture classification. We report the number of subjects and gesture classes and the included data types: RGB images, depth maps acquired with Structured Light sensors (SL), depth maps acquired with Time-of-Flight (ToF) sensors, infrared images. Moreover, we report the presence of 3D hand joints (3DJ) and dynamic gestures.

Dataset	Year	#sub.	#ges.	RGB	Depth	IR	3DJ	Dynamic
Unipd [136]	2014	14	10	✓	SL		✓	
VIVA [156]	2014	8	19	✓	SL	✓		✓
Nvidia [144]	2015	20	25	✓	SL	✓		✓
LMDHG [22]	2017	21	13			✓	✓	✓
Turms [18]	2018	7	-			✓		✓
Briareo § 3.1.2	2019	40	12	✓	ToF	✓	✓	✓

and the first version of the Microsoft Kinect. There are 10 different gestures performed by 14 people and each gesture is repeated for 10 times. The acquisition was conducted in an indoor environment and the devices were frontally placed with respect to the subjects. Unfortunately, hand gestures are static and belong to the American Sign Language.

The *VIVA Hand Gesture* dataset [156] was released for the namesake challenge, organized by the Laboratory for Intelligent and Safe Automobiles (LISA). It is designed to study natural human activities in confused and difficult contexts, with a variable illumination and frequent occlusions. 19 gesture classes are reported, taken from 8 different subjects, simulating real driving situations. Authors provide both RGB and depth maps acquired using the first version of the Microsoft Kinect. It is worth noting that users perform gestures around the infotainment area, placing the right hand on a green and flat surface to facilitate vision-based algorithms. The best gesture recognition method proposed in the challenge consists of a 3D CNN presented by Molchanov et al. [143].

The *Nvidia Dynamic Hand Gesture* dataset [144] presents 25 types of gestures recorded by two sensors (the active RGB-D sensor SoftKinetic DS325 and the infrared stereo camera DUO 3D) from different points of view. The acquisition devices are respectively placed frontally and top-mounted with respect to the driver position. The acquisition has been carried out in an indoor car simulator. Users perform gestures with the right hand while the left one grasps the steering wheel. The dataset contains the

recordings of 20 subjects, even if some of them contributed only partially, not performing the entire recording session.

The *Leap Motion Dynamic Hand Gesture* (LMDHG) dataset [22] contains unsegmented dynamic gestures, performed with either one or two hands. The Leap Motion sensor was employed as acquisition device and its SDK was used to extract the 3D coordinates of 23 hand joints. This dataset is composed of several sequences executed by 21 participants and contains 13 types of gestures performed randomly alongside an additional no-gesture action. Overall, 50 sequences are released, leading to a total of 608 gesture instances.

The automotive dataset called *Turms* [18] is acquired in a real automotive context, but it is focused on driver’s hand detection and tracking, thus no hand gestures are performed.

2.2.3 Human Pose Estimation and Refinement

In this section, we present an overview of methods related to the human body, in terms of Human Pose Estimation and Refinement in 2D and 3D.

While impressive results have been obtained in the 2D domain (in particular using RGB data), methods that estimate or refine the 3D human pose are still inaccurate and usually predict the pose in relative coordinates rather than in the 3D reference frame of the camera.

RGB-based human pose estimation

Intensity images represent the input of the large majority of human pose estimation methods available in the literature. Recently, most state-of-the-art 2D pose estimators exploit CNNs [30, 226, 154, 181, 26, 234, 198]. Wei et al. [226] propose a sequential architecture that learns implicit spatial models. Dense predictions, that represent the final human body joints, are increasingly refined through sequential stages within the network model. This approach is extended in the well-known work of Cao et al. [26], proposing the use of Part Affinity Fields (PAF) to learn the links between body parts. Recently, Sun et al. [198] introduced a model that preserve high-resolution representations through the whole pose estimation pipeline, repeating multi-scale fusions inside the deep model and achieving state-of-the-art results. Since all these methods achieve a good accuracy in the 2D

domain, we believe they can be successfully exploited also in other domains, *e.g.* the depth domain.

Depth map-based human pose estimation

Only a limited number of works tackles the problem of human pose estimation using depth maps, probably due to the limited number of datasets containing real or synthetic labelled depth data. Indeed, most of depth-based datasets are relatively small, *i.e.* not oriented to deep learning-based approaches, and automatically annotated, *i.e.* the annotations about the position of the body joints are extracted through existing methods [189], resulting in unreliable and imprecise annotations. The work of Shotton et al. [189] represents a milestone in the human pose estimation from depth maps. The approach frames the problem as a per-pixel classification task and uses Random Forests trained on a private synthetic dataset to solve it. The method reaches a reasonable accuracy at real-time speed and is publicly available in the Microsoft Kinect SDK, leading to its widespread use in both the gaming and the research community. Girshick et al. [65] propose a method, based on Hough forests, that directly regresses body joint coordinates from depth maps, without the use of intermediate representations. The system is able to localize visible as well as occluded body joints. In the work of Jung et al. [90], random trees are employed for the body joint localization from a single depth image. Then, joints are classified using a nearest neighbor approach. Haque et al. [74] present the *Invariant-Top View* dataset (ITOP), which contains about 50k low-quality depth images from both top and side views and manually annotated body joints. In the same work, the authors propose a deep model that embeds local regions into a view-invariant feature space and use them to regress the human pose. The *Watch-n-Patch* dataset [230] was collected for the unsupervised learning of relations and actions task. Its body joints annotation are obtained applying an off-the-shelf method [189], therefore they are not particularly accurate, in particular when subjects stand in a non-frontal position.

Human pose refinement

Most of the existing methods for the Human Pose Refinement are based on the 2D information available in the intensity images. Generally, these methods [33, 154, 24] exploit a multi-stage architecture, trained end-to-end,

in order to iteratively refine the pose estimation of previous stages or models. Other methods [30] exploit a shared weight model to estimate the error on the pose prediction. As reported in the work of Moon et al. [147], all these methods merge in a single model the pose estimation and the refinement task, obtaining a refinement module that is strictly dependent on the pose estimation approach. The same authors propose a solution called PoseFix, a model-agnostic human pose refinement network which is trained with synthetic poses generated exploiting general human pose error statistics [182]. A similar approach has been introduced by Fieraru et al. [57]: a simple post-processing network is trained through synthetic poses generated using hand-crafted rules. Zhang et al. [250] recently proposed a method that predicts an initial 3D pose which is then refined by a point cloud-based network, while Wan et al. [220] proposed an approach, based on RGB and segmentation images, that focuses on body parts to refine the 3D human pose.

2.2.4 Anthropometric Measurements

In this section, we present a brief summary of datasets and methods related to the estimation of anthropometric measures using visual cues, rather than manual, physical or invasive measurements. In general, there are currently very few public datasets and methods related to this field.

In the literature, there is a lack of depth-based public datasets containing visual data and anthropometric measurements. Indeed, no real-world datasets containing multimodal visual data are publicly available at the time of writing. To deal with this lack, several methods generate synthetic datasets that are easily recorded and annotated with ground truth measurements and collect or make use of private datasets, mainly for testing purposes.

The *CAESAR 3D Anthropometric Database* [180] includes measurements for 2k American and European subjects. It consists of 3D model scans and manual anthropometric measurements. For each subject a complete 3D model is provided, along with scans of standing and seating poses. This dataset is available upon payment of a fee. A variety of full-body 3D scans, captured with an expensive laser scanner, is introduced by Hasler et al. [76]. The database contains the scans of 59 males and 55 females, which are fit on a single 3D template model. In the work of Weiss et al. [227], a small dataset is proposed, containing only 4 subjects that are acquired through

the first version of the Microsoft Kinect. Each subject is standing in the “T pose” and acquired from four different directions: frontal, profile, back and halfway between frontal and profile.

In the work of Probst et al. [170], three different datasets are introduced but not publicly released. Two datasets are synthetically created, starting from the MPII human shape model [167], to obtain the 3D model of the human body. The first dataset contains subjects with the same pose but different body shapes, while the second one presents concurrent pose and shape variations. Ground truth anthropometric measurements are obtained using geodesic distances on meshes and body joints. Body height, shoulder width, leg and foot length, as well as a set of circumferences and thicknesses are computed. Using a virtual depth camera, depth maps are collected through simulation aiming to mimic the projection and the noise of real depth sensors. The real-world dataset includes 20 subjects wearing clothes in upright and lie-down poses. The first version of the Microsoft Kinect is used as acquisition device. Another synthetic dataset is introduced by Jain et al. [88], though it is strongly limited in shape and body variations.

There exists some works focused on specific anthropometric measurements. For instance, Guan et al. [68] propose a method to estimate the body height taking into account the subject’s face only. The method assumes that the body vertical proportions are constant during the human growth and approximately the same across subjects. Thus, they can be exploited to approximate anthropometric measurements. This method relies on an accurate camera calibration procedure. Momeni-k et al. [145] propose to exploit the knowledge of the camera pose (*i.e.* height and pitch angle of the camera with respect to the ground and a vanishing point) to regress the height of the acquired body or object regardless of its distance from the camera. Bieler et al. [13] propose a method to estimate the body height, exploiting the earth gravity. In addition, a novel dataset is presented, but it contains only RGB videos of jumping subjects and assuming asymmetric and articulated poses.

2.2.5 3D Object Reconstruction

In the following, we present an overview of articles addressing the task of 3D reconstruction of objects, in terms of shape, pose and texture, from 2D images. In particular, we focus on methods that rely on a single image and on the supervisory signals required during training. Currently, these

Table 2.3: Comparison between available approaches for 3D reconstruction from monocular images without corresponding 3D models. The comparison is based on training supervision, independence from offline-computed 3D templates, multi-category and dynamic subdivision support.

Approach	Supervision			w/o 3D Template	Multi category	Dynamic subdiv.
	Keypoint	Camera	Mask			
CSDM [94]	✗	✗	✗			
CMR [92]	✗	✗	✗			
VPL [95]		✗	✗			
CSM [109]			✗			
A-CSM [110]			✗			
IMR [210]			✗			
U-CMR [66]			✗			
UMR [121]			✗	✓		
MCMR § 4.5		✗	✗	✓	✓	✓

methods are limited to a single object category and are not able to generalize to several object classes. In particular, most of them require a pre-defined 3D template of the selected object category as a base shape, which is then deformed to fit the object in the input image.

In the last decade, many methods have been proposed to tackle the task of 3D reconstruction from a single image. However, the majority of these methods require supervisory signals which are hard to obtain in the real world and in the wild, such as 3D models [34, 63, 53, 255, 132, 222, 177, 236, 7, 119] or multi-view image collections [203, 176, 237, 71, 228, 208, 209, 87, 122].

Recently, thanks to the development of several differentiable renderers [130, 97, 157, 125, 32], a handful of methods [94, 79, 92] have shown that the task can be addressed as an inverse graphics problem using fewer supervisory signals, such as 2D segmentation masks and object keypoints. Following methods have even relaxed these constraints, training without keypoint supervision [32, 96, 95] or known camera poses [210, 66, 121]. However, these methods require image collections of a single object category and some of them need a meaningful initialization of a category-specific shape. Another group of works that exploit differentiable renderers address the

reconstruction task as a canonical surface mapping [109, 110] or a surface estimation task [116]. These methods usually require 3D supervision [116] or category-specific shape templates [109, 110]. In the work presented in this thesis, we focus on the 3D mesh reconstruction from single-view images without any category-specific template. Recently, Li et al. [120] proposed a video-based method and the use of multiple base shapes that are combined to produce a single deformable shape. These base shapes are introduced to cover the intra-class variation and are defined offline, thus they are fixed during training.

A comparative study of literature methods is proposed in Table 2.3, highlighting the differences in terms of training supervision, independence from offline-computed 3D templates, multi-category and dynamic subdivision support.

Chapter 3

Datasets

Deep learning has achieved stunning performance in the last decade, outperforming traditional machine learning methods in almost any computer vision field. In addition to breakthroughs in architecture design, optimization techniques and compute capabilities, large-scale datasets have played a crucial part in this success. Unfortunately, the majority of public vision datasets contain only color (RGB) images. Just a minor subset of them contains other kinds of data, such as depth maps, infrared images or thermal scans. Therefore, during the PhD we studied existing non-RGB datasets and recorded new ones, based on our needs. In the last years, we collected several vision datasets, aiming to cover most of the human-vehicle interactions using acquisition devices that are suitable for the automotive scenario, as discussed in Section 2.1. Indeed, we recorded a multi-modal multi-device dataset for face recognition, a dataset of dynamic hand gestures in a car simulator, refined annotations for depth-based human pose estimation and a set of anthropometric measurements of several people associated with their visual appearance.

In the rest of the chapter, we report the databases used in this thesis. We firstly present MultiSFace, Briareo, Watch-R-Patch and Baracca, the datasets that we collected to overcome the lack of existing datasets recorded using non-RGB sensors. Then, we present the public datasets that we used to assess the proposed methods and to compare with literature competitors.

This chapter is related to the author’s publications (ii), (iii), (ix), (xvii), listed in Appendix A.

3.1 Collected Datasets

To support the research activities carried out during the PhD, we collected several new datasets. Each dataset has peculiar features that were not found in the public datasets available at the time, but were required to investigate specific scenarios. In the following, we present four datasets that we collected for the tasks of (i) depth-based face recognition, (ii) dynamic hand gesture recognition, (iii) depth-based human pose estimation, (iv) estimation of anthropometric measurements.

3.1.1 Faces - MultiSFace

MultiSFace is a new cross-device dataset for the evaluation of multi-device and multi-distance face recognition based on depth maps or infrared data. The dataset is intended as a testing dataset, aiming at creating an extremely challenging benchmark and making it available to the research community. In particular, MultiSFace is designed to investigate how much the face recognition accuracy is impacted by sensors that have different quality and resolution and by acquisitions at different distances. Each subject is acquired with different synchronized sensors, capturing color, depth, near-infrared and thermal data. To the best of our knowledge, MultiSFace is the first publicly available¹ dataset of this kind.

In this thesis, we focus on the 2.5D data provided by the depth sensors.

Acquisition devices

To acquire the dataset, we selected pairs of high- and low-quality sensors able to capture intensity (RGB) images, depth maps and near-infrared images, thermal data.

As high-quality depth sensor, we chose the Pico Zense DCAM710², a high-resolution depth camera based on the Time-of-Flight technology. It acquires low-noise depth frames and infrared amplitudes at a resolution of 640×480 pixels at 30 fps in a range of 0.2 – 5 m with millimeter resolution. It also records RGB frames at a resolution of 1920×1080 pixels. The device has a small form factor ($103 \times 33 \times 22$ mm) and low power consumption (2.5 – 7.5 W). Moreover, the camera is suitable for tight spaces, having an

¹<https://aimagelab.ing.unimore.it/go/multisface>

²<https://www.picozense.com/>



Figure 3.1: Samples from the MultiSFace dataset. In the first row, the subject is near the sensors (1 m distance) while in the second row the subject is far from the sensors (2.5 m distance). Starting from the left: RGB and infrared images, high- and low-resolution depth maps, and high- and low-resolution thermal images.

infrared/depth sensor with a relatively wide field of view (69° horizontal, 51° vertical).

As low-quality depth sensor, we employed the low-resolution depth camera Pmdtec CamBoard Pico Flexx³, a ToF device focused on portability, in terms of weight (8 g) and form factor ($68 \times 17 \times 7.35$ mm). The camera has an extremely low power consumption (300 mW on average) and can capture depth data at up to 45 fps in the range 0.1 – 4 m with millimeter resolution. As shown in the fourth column of Figure 3.1, depth maps acquired by the sensor present a high level of noise and a limited resolution (171×224 pixels).

As high-quality thermal sensor, we used the Flir Boson 640⁴, which is a long-wave infrared (LWIR) thermal camera having high resolution (640×512 pixels) and frame rate (up to 60 fps). The sensor has a configurable power consumption (down to 500 mW) and a small form factor ($21 \times 21 \times 11$ mm, 7.5 g), but additional space is required by the lens. The camera was equipped with 14mm lens.

As low-quality thermal sensor, we employed the GroupGets PureThermal 2 board⁵, equipped with the long-wave infrared (LWIR) radiometric thermal

³<https://pmdtec.com/picofamily/flexx/>

⁴<https://prod.flir.it/products/boson/>

⁵<https://groupgets.com/manufacturers/getlab/products/purethermal-2-flir-lepton-smart-i-o-module>

Table 3.1: Statistics of the MultiSFace dataset.

Dataset	MultiSFace	Test set (frontal depth data)
Subjects	31	31
Chance (%)	3.2	3.2
Frames	621k	3.5k
Sensors	6	2
Streams	8	2
Poses	3	3
Settings	2 (near, far)	2 (near, far)
Sessions	2	2

sensor Teledyne Flir Lepton 3.5⁶, which has a relatively low resolution (160×120 pixels) and frame rate (8.7 fps). The sensor has a low power consumption (300 mW on average, 2.5 W maximum peak) and an extremely small form factor ($22 \times 30 \times 8$ mm).

In addition, we acquired standard color images using two RGB sensors: the low-quality RGB camera integrated in the Pico Zense DCAM710 and a high-quality Logitech webcam. Both cameras record data at Full HD (1920×1080 pixels) resolution.

Statistics

The dataset contains the recordings of 31 subjects, acquired in three different poses – frontal, side, and back – at two different distances – near (1 m) and far (2.5 m) – through the devices listed before. Samples from the acquired data are shown in Figure 3.1.

Table 3.1 reports the statistics of the dataset. In particular, they include the number of recorded subjects, the recognition chance probability, the overall number of frames (sampled for the testing set) and the number of acquisition devices, provided streams, subjects’ poses, acquisition settings, and recording sessions. The recognition chance probability is a measure of the level of complexity of the dataset by means of the accuracy in the face recognition task using random predictions.

⁶<https://groupgets.com/manufacturers/teledyne-flir/products/lepton-3-5>



Figure 3.2: Samples from the Briareo dataset. From left to right, infrared, RGB, and depth data are shown.

3.1.2 Dynamic Hand Gestures - Briareo

To investigate the hand gesture-based interaction between a car passenger and the infotainment system, we collected *Briareo*, a new dataset specifically designed for the driver hand gesture classification and segmentation with deep learning-based approaches. Aiming at a smooth and natural interaction, we focused on dynamic hand gestures, *i.e.* gestures that are a combination of motion and one or multiple hand poses. Differently from previous works, the visual data was collected from an innovative point of view: the acquisition devices were placed in the central tunnel console between the driver and the passenger seat, orientated towards the car ceiling. This configuration reduces the visual occlusions that may be produced by the user’s body, protects the sensors from being hit by direct sunlight, (which is a potential cause of critical failure for infrared-based sensors) while it can be easily integrated in the car cockpit. The dataset is publicly available⁷.

Acquisition devices

Three different cameras were used for the acquisition, focusing on devices that are reliable in low-light conditions and robust to severe light changes, such as the infrared-based sensors.

We used the Time-of-flight depth sensor Pmdtec CamBoard Pico Flexx, presented in the previous section. Limiting the acquisition range to 0.1 – 1 m, sufficient for the in-car setting, the camera is able to acquire at 45 frames per second. In addition, we employ the Leap Motion⁸, an infrared stereo

⁷<http://imagelab.ing.unimore.it/briareo>

⁸<https://www.leapmotion.com>

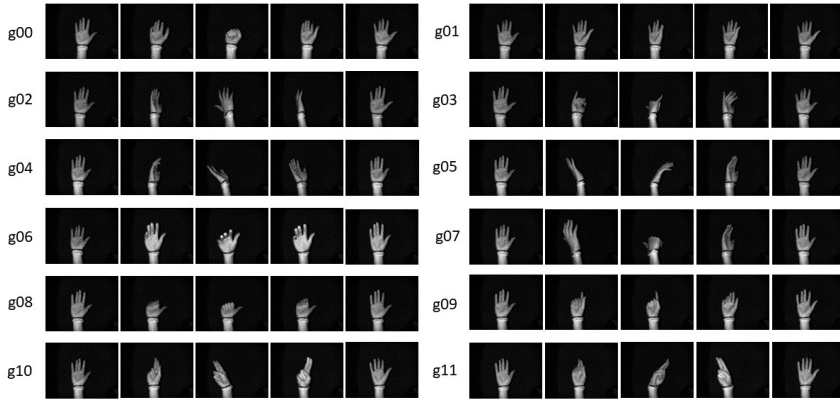


Figure 3.3: Gesture classes included in the Briareo dataset.

camera with fish-eye lens providing 150-degree field of view. The device captures pairs of distorted frames with a spatial resolution of 640×240 and provides rectified frames of size 400×400 pixels. The camera runs at up to 200 frames per second and is very compact and lightweight ($70 \times 12 \times 3$ mm, 32 g). Moreover, the Leap Motion SDK provides the location of several hand joints, along with their orientation and the bone lengths. Finally, a traditional RGB camera was used, acquiring up to 30 frames per second. To simulate the automotive environment, no external light sources were added. Thus, images captured by this camera are dark and low-contrast intensity images.

The cameras were used simultaneously and acquired synchronized data. The released dataset contains depth maps and infrared amplitudes (Pico Flexx), raw and rectified infrared images (Leap Motion), 3D hand joints (Leap Motion SDK), RGB images (standard camera). Frame samples are shown in Figure 3.2.

Statistics

The Briareo dataset contains the following 12 dynamic gesture classes: *fist* (g00), *pinch* (g01), *flip-over* (g02), *telephone* (g03), *right swipe* (g04), *left swipe* (g05), *top-down swipe* (g06), *bottom-up swipe* (g07), *thumb* (g08), *index* (g09), *clockwise rotation* (g10) and *counterclockwise rotation* (g11).

They are represented with sequences of frames in Figure 3.3

A total of 40 subjects (33 males and 7 females) took part to the data collection. Every subject performed each gesture 3 times, leading to a total of 120 collected sequences. Each sequence lasted at least 40 frames. In addition, each subject recorded an additional sequence performing all hand gestures one after another.

3.1.3 Human Poses - Watch-R(efined)-Patch

Along with the analysis of dynamic hand gestures, we focus our attention on the estimation of entire body pose and collect Watch-R-Patch, a new set of annotations for the existing Watch-n-Patch dataset [230]. Watch-n-Patch is designed for the action recognition task: it contains RGB frames and depth maps of indoor environments with a single person performing an action. In addition to action annotations, ground-truth human poses are provided for every frame, making it suitable for the evaluation of human pose estimation methods. However, human poses have been annotated using the Random Forest-based method proposed by Shotton et al. [189], which is available in the Microsoft Kinect SDK. This algorithm runs in real time, but its annotations have limited accuracy: the authors report a mean average precision of 0.655 on synthetic data with full rotations [189]. A detailed description of Watch-n-Patch is provided in Section 3.2.3.

To overcome imprecise automatic annotations, we collected Watch-R-Patch, a refined set of annotations for the Watch-n-Patch dataset. Original wrong, imprecise or missing body joints have been manually corrected for 20 training sequences and 20 testing sequences, equally split between the different scenarios of the dataset, *i.e.* *office* and *kitchen*.

The dataset is publicly available⁹.

Annotation procedure

We collect refined annotations for Watch-n-Patch using a quick and easy-to-use annotation tool. The developed tool shows the original body joints (*i.e.* the Watch-n-Patch joints) on top of the acquired depth map. The user is then able to move the incorrect joints in the proper positions and add missing joints using the mouse in a drag-and-drop fashion. As soon as every incorrect or missing joint is positioned in the correct location, the

⁹<http://aimagelab.ing.unimore.it/depthbodypose>

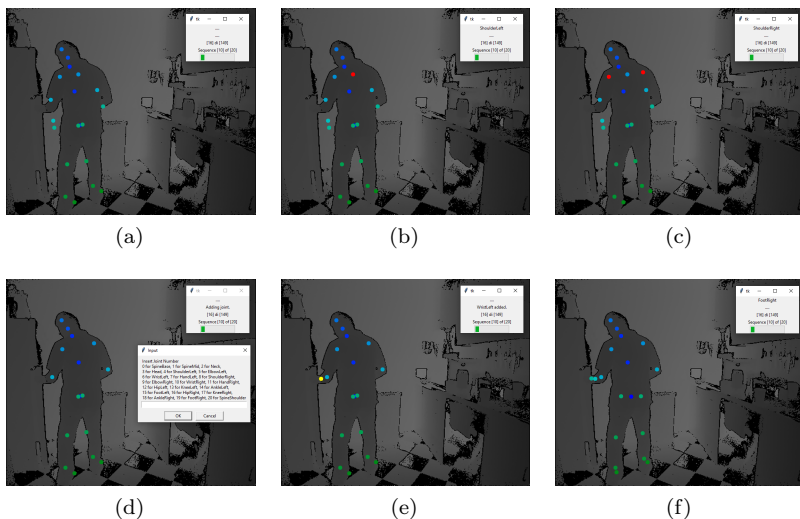


Figure 3.4: Overview of the annotation tool. The tool shows the original joint locations (a) and each joint can be selected to view its name (b) or to move it in the correct location (c). Missing joints can be added (d) (e). Then, annotations (f) can be saved and the next frame is shown.

user can save the new annotation and move to the next frame. It is worth noting that, in this way, the user has only to move the joints in the wrong position while already-correct joints do not have to be moved or inserted. Therefore, original correct joints are preserved, while improving wrongly predicted joints. We have ignored finger joints (tip and thumb) since original annotations are not reliable and these joints are often occluded.

The annotation tool is publicly released¹⁰. An overview of its interface is shown in Figure 3.4.

Statistics

We manually annotate body joints in 20 sequences from the original training set and 20 sequences from the original testing set. Sequences are equally split

¹⁰<https://github.com/aimagelab/human-pose-annotation-tool>

Table 3.2: Statistics of the Watch-R-Patch dataset.

Split	Sequences		Frames	Annotated frames	Modified joints (%)	mAP
	Kitchen	Office				
Train	data_02-28-33	data_01-50-09	3385	1135	75.7	0.574
	data_03-22-44	data_03-28-59				
	data_03-38-20	data_04-02-43				
	data_03-42-37	data_04-31-13				
	data_03-46-49	data_04-41-55				
	data_03-50-38	data_04-47-41				
	data_04-07-17	data_04-56-00				
	data_04-17-37	data_05-31-10				
	data_04-31-11	data_05-34-47				
	data_04-34-13	data_12-03-57				
Val	data_01-52-55	data_02-32-08	995	766	64.3	0.600
	data_03-53-06	data_02-50-20				
	data_04-52-02	data_03-25-32				
Test	data_02-10-35	data_03-04-16	2213	1428	55.5	0.610
	data_03-45-21	data_03-05-15				
	data_04-13-06	data_03-21-23				
	data_04-27-09	data_03-35-07				
	data_04-51-42	data_03-58-25				
	data_05-04-12	data_04-30-36				
	data_12-07-43	data_11-11-59				
Overall	-	-	6593	3329	64.4	0.595

between office and kitchen sequences. To speed up the annotation procedure and increase the scene variability, we decided to refine the annotation of a frame every 3 frames in the original sequences. In some test sequences, every frame annotation has been refined. The overall number of annotated frames is 3329, 1135 in the training set, 766 in the validation set, and 1428 in the testing set. We also propose an official validation set for the refined annotations, composed of a subset of the testing set, in order to standardize the validation and testing procedures.

Additional statistics about the annotated sequences and the proposed train, validation, and test splits are reported in Table 3.2. A qualitative overview of the dataset is reported in Figure 3.5.

3.1.4 Anthropometry - Baracca

Baracca is a new challenging and multimodal dataset collected for the estimation of anthropometric measurements, focusing on the automotive context. Indeed, an automatic estimation of the anthropometric measure-

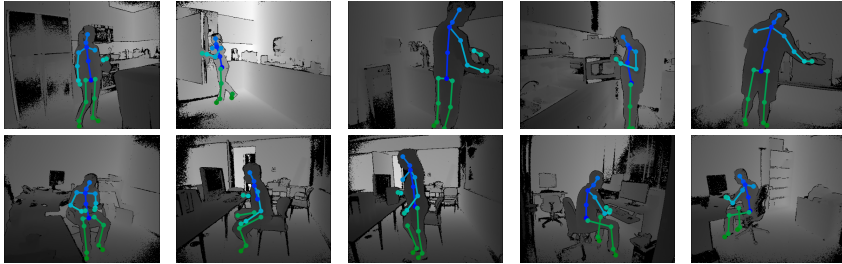


Figure 3.5: Samples from the Watch-R-Patch dataset.

ments of the driver (and passengers) – approaching or inside the car – can be used to improve in-cabin ergonomics and human-car interactions (for instance, adjusting the position of seats or rear mirrors). Therefore, the dataset includes in-car and outside views and contain RGB, depth, infrared, and thermal data, along with a set of anthropometric measurements of the participants. To the best of our knowledge, this is the first publicly released dataset that contains depth, infrared, thermal and RGB images, along with manually collected human body measurements.

The dataset is publicly released¹¹.

Acquisition devices

Considering the requirements imposed by the automotive context, we select two infrared sensors as acquisition devices: the depth sensor Pico Zense DCAM710 and the board PureThermal 2 equipped with the radiometric thermal sensor FLIR Lepton 3.5. Both of them are described in Section 3.1.1. These devices are suitable for the automotive context, thanks to the compact form factor and low power consumption.

Statistics and annotations

We synchronously collect data from the 2 sensors presented above, recording 4 different data streams (RGB, depth, infrared amplitude, thermal). Overall, the dataset consists of more than 9k frames. 30 subjects (26 males, 4

¹¹<https://aimagelab.ing.unimore.it/go/baracca>

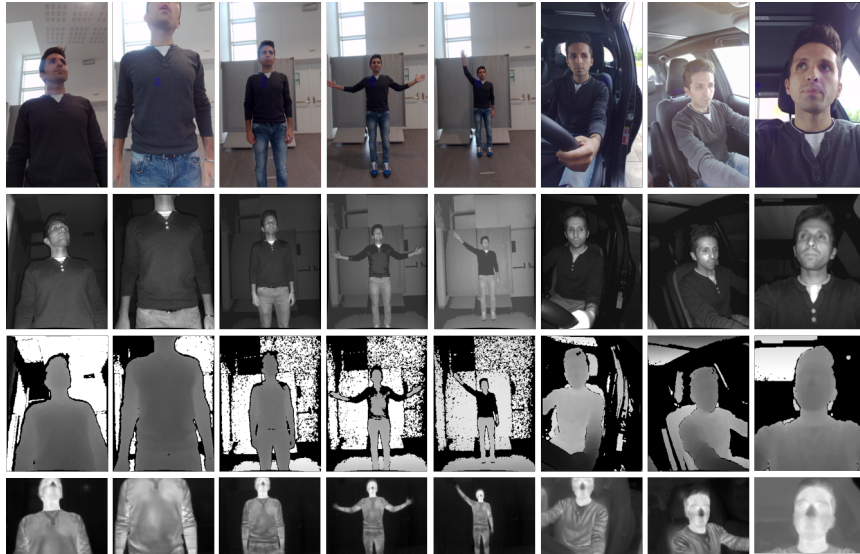


Figure 3.6: Samples from the Baracca dataset. Rows contain RGB, infrared (IR), depth, and thermal data; columns contain different acquisition points of view (5 indoor views, 3 in-car views).

females) participated in the data collection. For each subject, 5 outside-view (recorded indoor) and 3 in-car sequences were recorded using different points of view. In the outside-view sequences, the subject stands in front of the acquisition devices at different distances. The first two sequences are recorded at 0.6 m with two different camera viewpoints: top-view and frontal. Then, data are collected frontally at 1 m, 1.5 m and 2 m. In the in-car sequences, cameras are placed on the left A pillar, on the rear-view mirror, and behind the steering wheel. In this setting, only the upper body part of the subject – the driver – is visible. For each sequence, we recorded 10 consecutive frames while the subject was free to move the upper part of the body. Samples from the dataset are shown in Figure 3.6.

After the acquisition, the following set of anthropometric measurements was collected for each participant: *height*, *shoulder width*, *forearm and arm length*, *torso width*, *leg length* and *eye height from the ground*. We also recorded some soft-biometric traits: *age*, *sex* and *weight*. Anthropometric

Table 3.3: Statistics of the Baracca dataset.

Measure	Mean	Std. Dev.
Height (cm)	175.2	7.100
Eye Height (cm)	164.6	7.059
Forearm (cm)	25.73	1.879
Arm (cm)	26.67	2.134
Shoulders (cm)	42.27	3.255
Torso (cm)	38.63	2.702
Leg (cm)	103.8	5.536
Age (years)	26.57	3.981
Weight (kg)	72.03	12.71
BMI (kg/m^2)	23.35	3.222

measurement statistics are reported in Table 3.3.

For fair experiments and comparisons, we split the dataset in official cross-subject train and test splits. 24 subjects (including 3 females) are included in the training set while 6 subjects (including 1 female) are included in the test set.

Additional annotations

We computed and released the body pose of the subjects for each recorded image using a deep neural network. Specifically, the dataset includes the position of 15 skeleton joints in (x, y) image coordinates. Joint prediction is performed using HRNet [198], a recent human pose estimation method. The network is trained for 210 epochs on the COCO dataset [123], which contains RGB images only, with severe data augmentation. Please refer to [198] for additional details. Thanks to the adopted augmentation technique, the network is extremely accurate and able to work in various scenarios. Therefore, we employ it to estimate the body joints of the subjects in each recorded image, obtaining accurate human poses on RGB, IR, and thermal images. In the latter case, images have been normalized and converted to 8-bit images before the pose estimation. Since the infrared images and the depth maps are aligned, annotations obtained on infrared images are valid also on the depth maps.

3.2 Existing Datasets

Along with newly collected datasets, we also used existing public datasets to evaluate the proposed methods and compare with literature approaches. As done in the previous section, the datasets are organized according to the tasks they have been used for: (i) depth-based face recognition, (ii) dynamic hand gesture recognition, (iii) depth-based human pose estimation, (iv) 3D object reconstruction.

3.2.1 Faces

Biwi

Introduced by Fanelli et al. [54], the *Biwi* dataset is designed for the task of depth-based face recognition and head pose estimation. It contains approximately 15k depth frames of the upper body part of 20 subjects, acquired with the first version of the Microsoft Kinect (based on the Structured-Light technology). Each subject is recorded in a single continuous sequence during which they were asked to rotate the head spanning all of the head angles they were capable of.

In the experiments presented in this thesis, we use the first half of each sequence as training set. The second half is randomly shuffled and split in the validation (40%) and the test set (60%). This is mandatory, since there is only one session per most of the subjects and each session contains a scripted set of head movements. Indeed, only four subjects are recorded twice. For the sake of fairness, we do not use the additional recordings of these subjects in our work.

Curtinfaces

Released by Li et al. [118], the *CurtinFaces* dataset aims at simulating a real-world uncontrolled face recognition problem. To this end, subjects are photographed under varying expressions, poses, illumination sources, and disguises. Data is acquired with the first version of the Microsoft Kinect (based on the Structured-Light technology), totalling 5044 images recorded from 52 subjects (97 images per subject). Each subject is recorded while performing 7 different expressions (neutral and the 6 basic emotions [49]), under 7 different poses and 5 illumination conditions. In addition, subjects

are recorded in frontal, left, and right poses and under two types of occlusions (sunglasses, hands).

In the experiments presented in this thesis, we use 18 images per subject as training set (as in the original paper), 8 images per subject as validation set, and the remaining images as test set (*i.e.* 71 images per subject). We refer the reader to [118] for more details regarding the training split. The validation split is sampled, including a different pose for every different expression and two illumination variations, in order to cover the dataset distribution.

Pandora

Proposed in the work of Borghi et al. [16], the *Pandora* dataset was collected for the head pose estimation task, but it has been used for the face recognition task too [19, 20]. Acquired using the second version of the Microsoft Kinect (Time-of-Flight device), it contains 22 subjects (10 males, 12 females) and 5 sequences for each subject. The dataset contains more than 250k RGB images (1920×1080 pixels) and 16-bit depth maps (512×424 pixels), containing the central and upper body of the subjects. The faces can be occluded by the presence of garments and extreme head poses. In particular, three sequences contain from almost none to wide head movements (up to $\pm 125^\circ$ yaw, $\pm 100^\circ$ pitch, $\pm 70^\circ$ roll) and two sequences contain free movements while subjects wear garments or hold objects.

In the experiments presented in this thesis, we use the sequences without garments and artificial occlusions (*i.e.* the first three sequences of each subject). In particular, for each subject, we use the first sequence as training set, the second one as validation set, and the third one as test set.

Lock3DFace

Published by Zhang et al. [245], the Lock3DFace database is designed for the task of depth-based face recognition. It contains more than 300k frames of 509 different subjects recorded with the second version of the Microsoft Kinect (Time-of-Flight device) in multiple acquisition sessions. Variations in poses, facial expressions, and occlusions are included and each variation is performed multiple times (from two to six recordings per variation). Moreover, 169 subjects were recorded in separate sessions with a temporal offset of up to 7 months. The dataset has been originally split in a gallery

set, which is composed of the first recording of the neutral type for each subject, and three different probe sets, containing different subsets of the other recordings. We refer the reader to the original paper [245] for further details.

In the experiments presented in this thesis, we select the first recording of each type for each subject as a training set, regardless of the temporal session. Subsequently, since the number of recordings per variation is subject dependent (it varies from 2 to 6), we select the first frame of the additional recordings as validation set and the following frames of each recording as test set.

3.2.2 Dynamic Hand Gestures

NVGestures

The *Nvidia Dynamic Hand Gesture* dataset [144], also called NVGestures, is the largest dynamic hand gesture dataset in an automotive setting, in terms of number of gestures, subjects and sequences. The dataset contains 25 different gestures performed by the users with the right hand while the left one grasps the steering wheel. Each gesture is repeated three times and acquired in 5-second video samples. Gestures range from swipes to rotations and from showing n fingers to showing the “okay” sign. The dataset contains the recordings of 20 subjects, but some of them contributed only partially, not performing the entire recording session. Video sequences are acquired in an indoor car simulator using two sensors: the SoftKinetic DS325, an active RGB-D sensor placed frontally, next to the infotainment system, and the DUO 3D, an infrared stereo camera mounted on top of the acquisition area facing downwards. As a result, the dataset contains 3 modalities (RGB, depth, IR) and 5 streams (color, depth, color mapped on depth, IR left, IR right).

In the experiments presented in this thesis, we employ the color (RGB), depth, and infrared (left IR) modalities. For a fair comparison with literature work, we compute the optical flow on color frames using the method proposed by Farneback [55], as done in previous work [144]. In addition, we compute an estimation of the surface normals from the depth maps. We report visual samples of the data in Section 4.2.2 while we refer the reader to the original paper [144] for further details.

3.2.3 Human Poses

Watch-N-Patch

Watch-n-Patch [230] is a challenging action recognition dataset with RGB-D data. RGB frames and depth maps are recorded with the second version of the Microsoft Kinect, capturing a single person performing an action in indoor environments. In addition to action annotations, ground-truth human poses are provided for every frame, making it suitable for the evaluation of human pose estimation methods. Authors recorded 7 people performing 21 different kinds of actions. Each recording contains a single subject performing multiple actions in one room chosen between 8 offices and 5 kitchens. In total, the dataset contains 458 videos, corresponding to about 230 minutes and 78k frames. Authors provide both RGB and depth frames (with a spatial resolution of 1920×1080 and 512×424 pixels, respectively) and human body skeletons (composed of 25 body joints). Human poses have been annotated using the Random Forest-based method proposed by Shotton et al. [189], available in the Microsoft Kinect SDK. Thus, the accuracy of body joint annotations is limited and errors are frequent.

ITOP

The *Invariant Top View* dataset [74], also known as ITOP, contains about 100k depth images from side and top views of a person performing an action, along with the 3D annotations of 15 human body joints. Authors recorded 20 subjects performing 15 different actions and split the dataset in a train set (80% of the data) and a test one (20% of the data). Depth images were recorded using two Asus Xtion Pro, a Structured-Light depth sensor having a resolution of 320×240 pixels. One sensor is placed above (“top-view”) and the other one in front of (“side-view”) the acquired subject. Exploiting the two points of view, the body joints are semi-automatically annotated and manually refined to lie inside the body of the subject, *i.e.* at the 3D center of the physical joint.

3.2.4 3D Objects

Pascal3D+

The *Pascal3D+* dataset [233] is a collection of 2D images with 3D annotations designed for the 3D object detection and pose estimation task. It contains more than 30k images of 12 rigid-object classes, from both PASCAL VOC [52, 75] and ImageNet [40], associated with 3D category-level models and coarse viewpoints [199, 158, 190, 191]. In addition, manually annotated foreground masks are available for the PASCAL VOC subset. For a fair comparison with the literature, we used Mask R-CNN [78] as an off-the-shelf segmentation algorithm to extract foreground masks for the rest of the dataset, as done in previous works [92, 66, 210].

In the experiments presented in this thesis, we use this dataset to train and evaluate 3D object reconstruction methods. We remove about 5% of images from the training set to create a balanced validation set, used for hyper-parameter selection. In addition, we use the segmentation masks obtained by the novel PointRend method [103] to obtain very-accurate foreground masks.

CUB

The *CUB-200-2011* dataset contains images of 200 bird species, annotated with bounding boxes, precise foreground masks, location of parts and keypoints, and categorical attributes (such as bill shape, wing color, tail pattern). To use the dataset for the 3D object reconstruction problem, we also employed the 3D camera poses computed on the dataset by Kanazawa et al. [92], as done in previous works [92, 66, 210]. Among the 312 attribute labels, divided in several categories, which are included in the dataset, there is the “has_shape” category, containing 14 possible shapes. In Section 4.5, we use 14 as a rough and unoptimized estimation of the number of different bird shape classes.

3.3 Connection to the proposed methods

In the next chapter, we present various approaches to tackle different aspects of the Human-Vehicle Interaction. To evaluate them, we make use of the datasets presented in this chapter, as follows:

- MultiSFace (Section 3.1.1) and other face datasets (Section 3.2.1) are used in Section 4.1, to evaluate different depth-map representations in the context of face recognition;
- Briareo (Section 3.1.2) and NVGestures (Section 3.2.2) are used in Section 4.2, to evaluate a transformer-based approach to the dynamic gesture recognition;
- Watch-R-Patch (Section 3.1.3) is used in Section 4.3.1, to evaluate a depth map-based 2D human pose estimation model;
- ITOP (Section 3.2.3) is used in Section 4.3.2, to evaluate a depth map-based 3D human pose refinement framework;
- Baracca (Section 3.1.4) is used in Section 4.4 to demonstrate that several solutions can perform the contact-free estimation of anthropometric measurements from visual data;
- Pascal3D+ and CUB (Section 3.2.4) are used in Section 4.5 to evaluate the proposed approach to the 3D reconstruction from 2D images;

Chapter 4

Proposed Methods and Experimental Results

At the time of writing, human-vehicle interactions are mostly based on manual actions requiring physical contacts. Indeed, common actions such as opening the car, setting up the infotainment system, adjusting the seat position and the rear mirrors, often require the user to physically interact with several manual or digital devices. In this context, Natural User Interfaces (NUIs) would greatly improve the user experience of drivers and car passengers, ensuring a smooth interaction with the car interfaces. Invisible interfaces supporting complex interactions in a natural way would indeed revolutionize how we interface with our vehicles. Thanks to the recent developments in precise depth and infrared cameras, in their miniaturization and in GPU-accelerated embedded boards, the development of computer vision-based natural user interfaces and their inclusion in passenger and commercial vehicles is now feasible.

To this end, in this chapter we investigate computer vision approaches that could arguably replace or integrate existing human-vehicle interaction means in the next few years. Considering the broad range of interactions occurring between a car and its drivers and passengers, we focus on four main areas and study the related computer vision tasks.

This chapter is related to the author's publications (ii), (ix), (xii), (xiii), (xvii), (xx), listed in Appendix A.

Identification

The first interaction occurring between a person and a car is the access. While existing access systems are based on something a person possesses, like the car keys or a registered smartphone, future access systems could be based on something a person is, like their fingerprint or their face. Face recognition is a natural, object-less and contact-less access method that is already widespread on both laptops and smartphones. Moreover, a face recognition system could be used not only to grant access to the car, but also to automatically adjust the car itself to the user's personal settings and preferences. Indeed, each user's preferences could be stored in advance and retrieved at the user's access to the car. Given that the automotive context has several constraints, in addition to safety and reliability (as described in Section 2.1), we propose the use of depth cameras for the face recognition task and investigate different depth map representations and deep models in Section 4.1.

Gestures

Another common interaction involves the in-vehicle infotainment system. Even though some car entertainment systems have introduced options to control them with voice commands, most of the interactions still require some sort of physical contact, potentially causing the driver distraction. At the same time, gesture recognition methods have become more and more accurate and reliable in recent years, while being extremely easy to use, paving the way to natural gesture-based interactions. To this end, we take into consideration dynamic hand gestures that are designed for the control of an infotainment system and introduce a simple yet effective deep architecture for their real-time recognition in Section 4.2. As done previously, we propose the sole use of depth and infrared cameras, obtaining remarkable results.

Posture and anthropometry

Along with the face and the gesture recognition, human pose and anthropometric measurements estimation can play an important role in improving the human-vehicle interaction, in two different scenarios. On the one hand, an in-car system could use the driver pose to verify its compliance with the recommended safe driving pose or to detect situations of distraction,

e.g. the use of a mobile phone. On the other hand, an acquisition device placed on the outside of the vehicle could detect an approaching person and scan them to estimate a set of anthropometric measurements. These measurements could, in turn, be employed to adjust the car settings, *e.g.* seat and steering wheel positions, according to the body of the approaching person. To the best of our knowledge, there are no publicly available products exhibiting these features. Therefore, applying the approach that we followed for the face and the gesture recognition, we focus on real-time solutions using infrared and depth data. In Section 4.3, we propose deep architectures for the 2D human pose estimation and its 3D refinement from depth maps. In Section 4.4, we investigate the estimation of anthropometric measurements from several data types, namely color, infrared, depth and thermal images.

Digital mirrors

A rising trend in the automotive industry is the replacement of traditional mirrors with their digital counterparts. However, digital cameras have to be placed in the same position and direction of the traditional mirrors in order to have a similar point of view, with strong implications on the car design and the aerodynamic efficiency. Moreover, the interaction with current digital mirrors emulates the traditional one. That is, the full potential of digital mirrors fuelled by cameras has not been investigated yet. In this context, we argue that a virtual 3D reconstruction of the vehicle surroundings from standard RGB cameras can be a feasible way to disentangle the camera location and the representation shown by the digital mirrors. Besides, the point of view of the computed 3D representation could be freely controlled by the car users. As a first step towards a full 3D reconstruction of the area surrounding the vehicle, we present a semi-supervised approach to the 3D reconstruction of vehicles of different categories from a single 2D image in Section 4.5. The evaluation shows that the proposed deep architecture can successfully reconstruct the shape, pose and appearance of objects from a single image.

Chapter structure

The rest of the chapter is organized as follows. Firstly, we present a thorough analysis of depth map representations for the face recognition task,

comparing different representations and algorithms and evaluating them on the collected MultiSFace dataset and on four public datasets. Secondly, we introduce a transformer-based architecture for the dynamic hand gesture recognition, making use of depth maps and other data modalities and achieving state-of-the-art results on two public datasets. Then, we expand our focus to the analysis of the human body as a whole, presenting methods for the 2D and 3D human pose estimation and refinement from depth maps and tackling the estimation of anthropometric measurements from several data types. Finally, we shift the view from the inside to the outside of the car, presenting a system that performs the 3D reconstruction of vehicles, in terms of shape, pose and texture, from 2D images.

4.1 Depth Map Representations for Face Recognition

In the computer vision field, face recognition is a widely studied task and impressive results have been obtained in the RGB domain [41, 126, 129], especially with frontal face poses and good lighting conditions. Moreover, a substantial improvement has been introduced by the adoption of deep neural networks [77, 201, 194] and huge datasets [25, 70, 85]. At the same time, interest in depth cameras and, consequently, depth maps, has steadily grown in the computer vision community. Their increasing popularity has been supported by the spread of inexpensive, but still accurate, active depth sensors and their ability to operate in dark or low-light conditions, thanks to the presence of infrared light or laser emitters [173]. For instance, in the automotive scenario [137, 156], depth sensors represent an effective solution to run non-invasive and vision-based algorithms, such as face verification [20], head pose estimation [16], or gesture recognition [45]. More generally, starting from the first release of the Microsoft Kinect device, depth cameras have enabled new interaction modalities between the users and the environment. Gaming [189], smartphones [213], health care [168] and human-computer interaction [217] are just some of the application fields where depth sensors have been used in addition to or in replacement of the RGB cameras.

However, the different building technologies of depth sensors—*e.g.* Structured Light (SL) and Time-of-Flight (ToF) to cite the most common — hinder the efficacy of deep learning-based models when working with depth maps acquired from different depth sensors or even with the same technology, but in different acquisition setups. Indeed, the problem of cross-dataset and cross-device generalization is very critical with depth data, especially with deep learning approaches.

Generally, the problem is mitigated in the RGB domain, in which intensity images, from the visual point of view, are similar across sensors and huge datasets that are composed of images acquired by different cameras are available.

More specifically, the use of depth maps in combination with deep learning methods presents the following issues:

- The difference between depth maps acquired with different devices is significant, in terms of visual appearance (holes, shadows, noise),

accuracy and detail preservation [185] (as it can be seen in Figure 3.1 and Figure 4.1).

- The same device is subject to environmental conditions, although the depth map should be independent of them; for example, it collects different data when facing direct sunlight or when the distance of the target from the device varies significantly. In the latter case, changes on the target distance affect not only the scale factor, but also the pixel values itself, the depth map quality and the level of noise.
- Mixed datasets, *i.e.* the dataset acquired with different types of depth devices, are still not publicly available. Moreover, the majority of the existing datasets are collected in a very limited number of acquisition settings, for instance using a single depth sensor for all of the the collected sequences. Thus, the generalization capabilities with respect to different devices and scenarios are often not analyzed in the literature.

Indeed, most of the available methods in the literature are task-tailored on a specific sensor, only performing intra-dataset tests, *i.e.* training and testing the proposed algorithms on the same data collection. Moreover, they usually use deep learning approaches to analyze depth maps that are represented as gray-level images, ignoring the intrinsic three-dimensional (3D) information that is embedded in depth data.

In this section, we study the use of depth maps and deep neural models for the face recognition task, in search of the depth map representation that maximizes the recognition accuracy and better generalizes on unseen data. In particular, we compare different representations of depth data (depth images, normal images, point clouds and voxels, as shown in Figure 4.1), pre-processing techniques (normalization, equalization, filtering and hole filling), sensor technology (SL and ToF) and face-to-camera distance, in a comprehensive analysis.

The proposed comparison mainly focuses on the output of the active depth devices, which have a limited and well-defined maximum range; other types of sensors, such as stereo cameras, 3D scanners and LiDARs are out of the scope of this work.

Summarizing, the main contributions of this work are the following:

- We provide the first rigorous extensive analysis of depth data representations for the face recognition task, testing the performance and

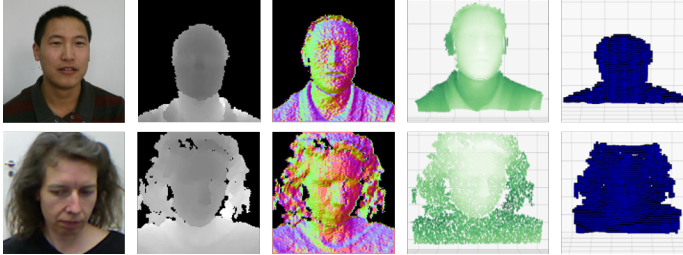


Figure 4.1: Sample images of different depth representation taken from Lock3DFace dataset [245] (Time-of-Flight, first row) and Biwi database [54] (Structured Light, second row). From the left, the RGB, depth and normal images, point clouds and voxels are reported.

generalization capabilities on four depth-based public datasets.

- We investigate the use of data pre-processing, such as filtering, equalization and hole filling, and depth image normalization, often exploited in the depth-based literature methods.
- We evaluate different sensor technologies, SL and ToF, and the impact of subject distance and device resolution using MultiSFace, our new dataset, presented in Section 3.1.1, that includes more than 11k depth maps captured with two different synchronized range sensors at different distances.

The experimental results suggest that normal images and point clouds that are computed from depth maps, even though rarely used in literature, are the best choice for achieving the highest accuracy and generalization in the face recognition task.

To the best of our knowledge, there are no existing works analyzing the use of different depth map representations and neural architectures for the face recognition task in the intra- and cross-dataset setting. Similar works [197] only address different representations of synthetic full 3D models of objects, in particular for object recognition and 6DoF pose estimation.

4.1.1 Methodology

In this work, we analyze the use of depth maps for deep face recognition. We aim at identifying the combination of data representation, pre-processing/normalization technique and deep learning model that obtains the highest recognition accuracy in both the intra- and the cross-dataset setting. In this section, we characterize this analysis, from the problem statement and the deep learning models to the datasets and pre-processing techniques.

Problem statement and experimental setting

We address the face recognition task as a face identification problem, where a single depth map of an unknown person, *i.e.* the probe, is compared to a gallery of known candidates in a closed-set scenario. In this setting, the recognition model compares the probe with each gallery identity, *i.e.* a one-to-many comparison, and then outputs a single label that represents the predicted identity of the probe. Given the predicted identity, we compare different approaches in terms of recognition accuracy (*i.e.* top-1 recognition rate) and compare different deep architectures in terms of computational complexity.

Within the different experimental settings (*i.e.* the different combinations of data representation, pre-processing and normalization steps and deep model), we employ the same training procedure. Each model is trained on the train split of the selected dataset for 50 epochs (that we empirically observe as a valid upper-bound limit), while using the Categorical Cross-Entropy (CCE) loss and the Adam optimizer. After every epoch, the validation accuracy is evaluated and, if higher than any validation accuracy obtained so far, the model parameters are saved (and later used for testing).

In the testing phase, we discard the last classification layer and compare the probe and gallery depth maps computing the cosine similarity between the deep features that were extracted by the networks [139]. For every probe, we select the predicted identity as the gallery candidate corresponding to the maximum similarity.

Deep Learning architectures

Well-known and representative deep learning-based models are selected for the evaluation part. For depth maps used as single-channel images, we

exploit the models VGG-16 [194], ResNet-18 [77] and Inception-v3 [201]. Voxels are used in combination with VoxNet [141], R3D and R(2+1)D [207], while PointNet [171] and PointNet⁺⁺ [172] are employed for point clouds.

Deep Networks are implemented in PyTorch and adapted for the specific task of face recognition from depth data, in terms of input channels and final classification layer. For instance, the first layer of the networks used to analyze the depth images is adapted to support a single-channel input, while the classification part of PointNet and PointNet⁺⁺ is used and the segmentation branch is discarded. For a fair comparison between models (image-based networks are often pre-trained on bigger datasets), all the networks are trained from scratch.

In all of the experiments on every dataset, we employ the same input format, as detailed in the following. Regarding the 2D CNNs, the input images are resized to the resolution of 128×128 pixels and the background behind the human face is filtered out, if present. The images are represented with single-channel images while using the 16-bit format. The depth values are expressed in mm. When considering the point clouds, we compute them from the depth maps, as detailed in Section 2.1.2. We consider, as valid, all of the points with a non-null depth value and feed them to the point cloud-based networks. The maximum number of points is set to 16,384. When using the 3D voxels, we obtain them from the point clouds of the human face. We centered the 3D volume at the point cloud center (computed as the mean of the coordinates of all the points) and set a cubic side $L = 400$ mm. The number of voxels per side m can be 32 or 64, as defined in the experimental results.

Datasets

Although the spread of depth sensors is still limited with respect to RGB ones, depth-based datasets containing faces are already available in the literature. Each of them has been acquired using a single depth sensor, *e.g.* Structured Light (SL) or Time of Flight (ToF).

Among them, we have selected two datasets that were acquired with the first version of the Microsoft Kinect sensor, based on the SL technology, and two datasets that were acquired with the second version of the same device, based on the ToF technology. We preferred to exclude other available datasets that contain a limited number of subjects (*e.g.* [134]), frames (*e.g.* [8]), unreliable depth data (*e.g.* [10]), or 3D facial models instead of depth

Table 4.1: Datasets selected for the proposed analysis. DT is the depth technology; #subjs, #frames, #cams are respectively the number of subjects, depth frames and used depth cameras. The chance (level) is the accuracy with random predictions. Settings corresponds to the position of the subject w.r.t. the acquisition device. Sessions is the number of different acquisitions per subject.

Dataset name	Year	DT	#subjs	#frames	chance (%)	#cams	Settings	Sessions
Biwi [54]	2011	SL	20	15k	5.0	1	1 (near)	1 or 2
CurtinFaces [118]	2013	SL	52	5k	2.9	1	1 (near)	17
Lock3DFace [245]	2016	ToF	509	300k	0.2	1	1 (near)	8 to 16
Pandora [16]	2017	ToF	22	125k	4.5	1	1 (near)	5
MultiSFace [166]	2020	ToF	31	11k	3.2	2	2 (near, far)	2

maps (*e.g.* [186]).

Table 4.1 reports an overview of the chosen datasets presenting, for each dataset, the sensor technology; the number of subjects, frames, cameras and sessions; the level of complexity when considering the face recognition task (expressed as chance level); the number of different acquisition settings. We split the data into train, validation and test sets using, whenever possible, different sessions/sequences for each subset. We aim at obtaining a fair subdivision, *i.e.*, the use of different sessions/sequences for each subset while including samples of each person in the training set. When the official splits conform to this policy, we used the official train, validation and test subsets. We also note that each employed dataset was acquired with a different procedure and thus requires a subdivision that is based on its structure, yielding a different number of recordings in different settings for each dataset. Table 4.2 reports the number of frames belonging to each split.

We also use the newly collected MultiSFace, a cross-device dataset for the evaluation of multi-device and multi-distance face recognition based on depth maps, which we presented in Section 3.1.1. The MultiSFace dataset allows for investigating the impact of using different depth sensors at varying distances on the face recognition accuracy. To the best of our knowledge, MultiSFace is the first publicly available dataset, in which each subject is acquired with different synchronized depth (and thermal) sensors.

For more information about the employed datasets, please refer to Section 3.1.1 and Section 3.2.1.

Table 4.2: Training, validation and testing splits adopted for each dataset. Frames are split following the procedures described in Section 4.1.1.

Dataset name	No. depth frames	Training	Validation	Testing
Biwi [54]	15k	6.6k	2.6k	3.9k
CurtinFaces [118]	5k	0.9k	0.4k	3.7k
Lock3DFace [245]	300k	12.2k	2.7k	17.8k
Pandora [16]	125k	9.3k	7.4k	9.5k
MultiSFace [166]	11k	-	-	3.5k

Pre-processing techniques

We select common image pre-processing techniques that are applied on depth images in the literature [245, 242, 249], such as filtering, histogram equalization and hole filling. We individually apply them on the depth image I_D , using OpenCV.

Filters are often applied to reduce the high level of noise caused, for instance, by external light sources and the use of an infrared emitter [185]. To this aim, in the tests we include a linear filter (Gaussian), a non-linear filter (Median) and a data-dependent, thus not shift-invariant, filter (Bilateral). In our experiments, we set the kernel size to 3, the sigma of the Gaussian filter to 0.8, the color sigma and the space sigma of the bilateral filter to 50.

Histogram equalization is applied to enhance the contrast in the intensity images and it can be used to stretch very similar values in depth facial images. Specifically, we consider the standard equalization, using 256 bins, and the Contrast-Limited Adaptive Histogram Equalization (CLAHE) [256] algorithm, setting the clip limit to 2 and the tile grid size to 8.

Depth maps often present pixels with missing or spurious depth values, due to specular or low albedo surfaces: typical parts with invalid values are hair and eye areas. Additionally, shadows, which are created by the disparity between the sensors and infrared emitter, contain missing values. Therefore, some works propose using hole filling (in-painting) techniques, replacing invalid data. In our work, we adopt the hole filling procedure that is proposed by Telea [204].

We report some visual results of these pre-processing techniques in Figure 4.2.

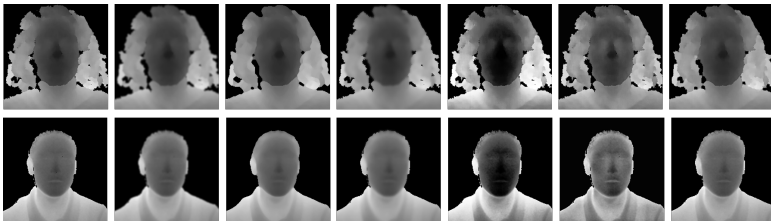


Figure 4.2: Sample images from Biwi [54] (first row) and Lock3d [245] (second row) showing the visual results of the pre-processing steps. From left to right: none (original depth image), gaussian blur, median blur, bilateral filtering, histogram equalization, CLAHE and hole filling. The images are converted to the 8-bit format for visualization.

Data normalization

Generally, data normalization is a key element during the training process of deep learning models with intensity images [108]. In our case, we test the following normalization procedures on depth data:

$$f_1(x) = x - \mu_x \quad (4.1)$$

$$f_2(x) = \frac{x - \mu_x}{\sigma_x} \quad (4.2)$$

$$f_3(x) = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4.3)$$

where μ_x and σ_x are the mean and the standard deviation of x . When applied to depth images, x is the set of valid pixel values (*i.e.* pixels that are not null, due to an invalid depth estimation or that do not exceed the maximum depth range of the device). Point clouds are normalized by applying the operation on each axis, independently. Equation (4.1) zero-centers the data/point coordinates, Equation (4.2) gives data/point coordinates with zero mean and unit variance, while Equation (4.3) outputs the values in the range $[0, 1]$.

Table 4.3: Intra-dataset results, in terms of recognition accuracy, using depth and normal images (I_D and I_N). We also report results applying pre-processing steps on depth images: filtering $F(I_D)$, hole filling $H(I_D)$, equalization $E(I_D)$, data normalization $N(I_D)$.

Model	Biwi [54]						CurtinFaces [118]					
	I_D	$F(I_D)$	$H(I_D)$	$E(I_D)$	$N(I_D)$	I_N	I_D	$F(I_D)$	$H(I_D)$	$E(I_D)$	$N(I_D)$	I_N
VGG [194]	32.5	33.6	32.9	29.9	26.6	43.1	60.5	57.7	57.4	63.4	57.5	66.5
Inception [201]	60.9	56.8	52.9	45.4	50.8	66.8	29.5	40.0	34.0	33.7	38.6	42.2
ResNet [77]	61.5	64.4	58.3	64.0	66.7	80.0	43.0	45.6	40.0	48.8	50.9	45.2

Model	Lock3DFace [245]						Pandora [16]					
	I_D	$F(I_D)$	$H(I_D)$	$E(I_D)$	$N(I_D)$	I_N	I_D	$F(I_D)$	$H(I_D)$	$E(I_D)$	$N(I_D)$	I_N
VGG [194]	54.6	53.4	55.2	61.3	54.9	62.1	51.6	51.2	47.2	54.0	51.3	57.4
Inception [201]	72.5	71.6	72.1	70.3	72.3	81.0	40.0	40.1	35.5	63.9	59.6	72.4
ResNet [77]	51.7	52.8	50.9	56.3	59.0	76.6	40.3	42.7	42.6	67.1	65.4	70.3

4.1.2 Experimental Evaluation

Intra-dataset experiments

Intra-dataset experiments are carried out on individual datasets, each split into training, validation and testing sets. Thus, models are trained and tested with data that were acquired by the same depth device and environment, then similar from a visual and quality point of view. These experiments are focused on the investigation regarding the use of depth data and deep architectures, in terms of accuracy in face recognition, without considering the generalization capabilities on different datasets and depth technologies. We report the results in terms of recognition accuracy, as described in the beginning of Section 4.1.1, while using depth and normal images, voxels and point clouds in Tables 4.3, 4.4 and 4.5.

We report the best performing pre-processing and normalization steps, which are individually applied, as described in Section 4.1.1. Specifically, for the depth images, we include the Gaussian filter for filtering (F), Equation (4.3) for data normalization (N), the histogram equalization (E) and the hole filling procedure (H). I_N represents the use of normal images as input data. For the point cloud, the data normalization referred as P_N is computed, as in Equation (4.1). For the voxels, two different sizes ($m = 32$ or $m = 64$) are evaluated.

Looking at the results of image-based methods (Table 4.3), in general the

Table 4.4: Intra-dataset results, in terms of recognition accuracy, using point clouds P . P_N represents the normalized point cloud computed while using Equation (4.1), as detailed in Section 4.1.1.

Model	Biwi		CurtinF.		Lock3D		Pandora	
	P	P_N	P	P_N	P	P_N	P	P_N
PointNet [171]	60.5	53.2	50.7	70.7	55.1	63.9	23.9	25.2
PointNet ⁺⁺ [172]	40.4	42.2	45.4	51.7	51.4	61.8	21.1	35.8

Table 4.5: Intra-dataset results, in terms of recognition accuracy, using voxels V . 32 and 64 specify the size m of the 3D volume (see Section 2.1.1).

Model	Biwi		CurtinF.		Lock3D		Pandora	
	V^{32}	V^{64}	V^{32}	V^{64}	V^{32}	V^{64}	V^{32}	V^{64}
VoxNet [141]	53.0	49.2	78.0	73.7	67.8	69.1	36.6	37.2
R3D [207]	64.4	63.3	69.5	71.4	71.0	70.1	30.0	31.9
R(2+1)D [207]	61.4	58.8	40.0	67.1	68.7	68.5	31.8	37.6

filtering, the equalization and the hole filling procedures do not introduce clear benefits, even if they are often exploited in literature, as highlighted in Section 2.2.1. Therefore, the additional computational load that is introduced by them is not justified by a corresponding increase of accuracy. Instead, data normalization generally maintains or improves the results, in particular on ToF data.

Nevertheless, the results show that normal images are the best data representation for recognizing faces while using CNNs in most cases. When compared to depth images, normal images do not contain the absolute distances of the target points, but they explicitly express 3D information that is related to the 3D shape of the captured scene. Thus, we hypothesize that the resulting representation is more suitable for the face recognition task while using depth devices.

Deep architectures based on point clouds and voxels generally achieve worse results than image-based approaches, as it can be seen in Tables 4.4 and 4.5. In the case of point clouds, the results show that data normalization is a key element to achieve a good level of accuracy (especially

Table 4.6: Cross-dataset results (same sensor technology), in terms of recognition accuracy. We report: the data type (I_D : Depth Maps, V : Voxels, P : Point Clouds), the dataset (C: Curtinfaces [118], B: Biwi [54], L: Lock3DFace [245], P: Pandora [16]), the depth sensor technology (SL: Structured Light, ToF: Time-of-Flight). $D_1 \rightarrow D_2$ means “trained on D_1 and tested on D_2 ”.

Model	Same Sensor Technology			
	SL		ToF	
	C→B	B→C	P→L	L→P
best (intra)	80.0	66.5	81.0	72.4
I_D	34.4	18.2	31.3	25.6
I_N	34.6	35.3	45.6	35.6
best (intra)	60.5	70.7	63.9	35.8
P	36.4	36.9	40.7	30.0
P_N	36.1	39.8	56.2	39.6
best (intra)	64.4	78.0	71.0	37.6
V^{32}	22.6	20.1	33.5	30.4
V^{64}	21.7	21.8	38.0	28.2

with PointNet⁺⁺), while experiments with voxels show that the attained accuracy is not dependent on the network architecture and voxel size. Even from a computational point of view, CNNs are usually the best choice in terms of memory usage and inference time.

Cross-dataset experiments

Cross-dataset experiments are carried out considering two datasets at a time, one for training the deep models and one for testing. Probe and gallery data are both extracted from the second dataset, as detailed in Section 3.1.1 and Section 3.2.1. Each experiment is referred in the form “ $D_1 \rightarrow D_2$ ”, which means that the model is trained on the dataset D_1 and tested on D_2 . Compared to the intra-dataset case, these tests are focused on the generalization capabilities of deep models, in particular when the two datasets have been acquired while using different sensor technologies or in different acquisition settings.

Table 4.7: Cross-dataset results (different sensor technology), in terms of recognition accuracy. We report: the data type (I_D : Depth Maps, V : Voxels, P : Point Clouds), the dataset (C: Curtinfaces [118], B: Biwi [54], L: Lock3DFace [245], P: Pandora [16]), the depth sensor technology (SL: Structured Light, ToF: Time-of-Flight). $D_1 \rightarrow D_2$ means “trained on D_1 and tested on D_2 ”.

Model	Different Sensor Technology							
	SL \rightarrow ToF				ToF \rightarrow SL			
	C \rightarrow L	C \rightarrow P	B \rightarrow L	B \rightarrow P	P \rightarrow B	P \rightarrow C	L \rightarrow B	L \rightarrow C
best (intra)	72.5	67.1	72.5	67.1	66.7	63.4	66.7	63.4
I_D	32.8	26.8	30.5	24.9	28.4	14.1	33.1	18.7
I_N	25.4	23.2	45.3	32.6	48.2	34.2	37.0	33.0
best (intra)	63.9	35.8	63.9	35.8	60.5	70.7	60.5	70.7
P	30.2	12.3	37.4	26.3	43.5	35.6	37.4	34.7
P_N	58.1	39.1	54.0	34.0	37.5	46.5	35.2	43.3
best (intra)	71.0	37.6	71.0	37.6	64.4	78.0	64.4	78.0
V^{32}	41.3	23.0	36.9	21.3	18.3	15.6	27.1	33.7
V^{64}	40.4	23.7	35.8	22.3	22.7	21.4	21.1	33.7

In Table 4.6 and Table 4.7, we report the most significant results of the cross-dataset evaluation, obtained with ResNet, R3D and PointNet⁺⁺ for depth images, normal images, voxels and point clouds. As in the intra-dataset setting, the results are expressed in terms of recognition accuracy, as described at the beginning of Section 4.1.1. Table 4.6 contains results obtained using train and test datasets that were acquired with the same sensor technology, while Table 4.7 contains experiments in which the sensor technology of the test dataset is different from the one of the training dataset. For the sake of comparison, the best results that were obtained in the corresponding intra-dataset experiment are reported as "best (intra)". The reference values included in the table are the ones obtained using D_2 for both the training and testing and collected from the previous section.

First of all, we note that point cloud-based methods are the best choice in the cross-dataset setting, even if point clouds that are computed from depth maps are rarely used in the literature for the face recognition task. They

achieve the best accuracy with both same and different sensor technologies, as confirmed by both the absolute accuracy and the minor performance drop when compared with the intra-dataset references, as shown in Tables 4.6 and 4.7. This finding confirms that this data representation is more independent from the acquisition sensor and that the point cloud-based models are less prone to overfit on the training dataset. Therefore, point clouds should be used when the testing data are acquired with different or unknown depth sensors. We believe that the performance discrepancy between the intra-dataset setting and cross-dataset one reveals a potential difficulty in assessing the quality of point cloud-based methods. In fact, most of the experiments that are reported in the literature do not deal with cross-dataset tests and may only observe unsatisfactory results in the intra-dataset setting.

Regarding the other depth map representations, normal images analyzed with CNNs obtain higher accuracy when compared to depth images and voxels, thus confirming that surface normals are an informative and invariant representation of depth maps for the face recognition task.

As it can be noted, the architectures trained on Pandora achieve better results than the ones trained on Lock3DFace whether tested on Biwi or CurtinFaces, in particular when considering normal images and point clouds. Because the main differences between Pandora and Lock3D are the number of frames with different poses (higher in the former) and the number of subjects (higher in the latter), we hypothesize that, for the face recognition on 3D representations of depth data, the head pose variability of the training set is more crucial than the number of different identities.

Cross-device and cross-distance experiments

The proposed dataset MultiSFace contains data that were acquired from diversified positions by two different depth sensors. Therefore, it could be used to run an additional set of challenging experiments. In fact, it can be employed to evaluate the recognition accuracy when the gallery set and the probe data are collected by different devices or at different sensor-subject distances.

We run this set of experiments employing architectures that were trained on the Lock3DFace dataset (we used ResNet for I_D , PointNet⁺⁺ for P_N and $R3D$ for V^{32}). We evaluate the recognition accuracy using two ToF sensors (having different resolutions), labelled as High Resolution (HR) and

Table 4.8: Results on MultiSFace, in terms of recognition accuracy. Tests are carried using different gallery and probe data. In the left part, cross-distance tests (**N**ear and **F**ar distance) are reported keeping the sensor fixed. In the right part, cross-device tests (**H**igh and **L**ow resolution) are reported keeping the distance fixed.

Cross-distance				Cross-device			
Setting	Model	N → F	F → N	Setting	Model	H → L	L → H
H	I_N	6.6	4.9	F	I_N	3.4	5.2
	P_N	16.7	13.9		P_N	9.2	7.5
	V^{32}	9.0	7.5		V^{32}	3.1	5.4
L	I_N	4.6	4.4	N	I_N	6.4	2.7
	P_N	8.6	7.2		P_N	3.1	6.0
	V^{32}	4.4	5.0		V^{32}	10.8	8.0

Low Resolution (LR), and two different sensor-subject distances, labelled as Near (N) and Far (F). It should be recalled that, since depth maps are acquired by depth devices, the sensor-subject distance directly affects their quality, in terms of noise and point density. Therefore, even if some data representations are distance-invariant (*e.g.* depth normals, voxels and point clouds), the depth data acquired by the sensors are not.

Table 4.8 reports the results in terms of recognition accuracy. The better generalization capabilities of the point cloud representation and PointNet⁺⁺ are highlighted. However, the tested approaches do not reach satisfactory recognition accuracy in these challenging cases. Image-based methods achieve results around 4–6%, which are only slightly higher than the chance level, while the voxel representation can be suitable for the cross-device scenarios. In fact, the voxel quantization filters out the differences in the resolution and quality between the sensors. This holds at the Near (N) distance, where both of the sensors acquire sufficiently precise depth maps, while it does not hold at the Far (F) distance, due to the noisy sparse data acquired by the sensors, especially the low-resolution one.

Computational complexity

The recognition accuracy is not the only element to be taken into account during the development of real-world face recognition systems. Therefore,

Table 4.9: A comparison of the computational complexity of different methods. We report the number of parameters, the amount of memory (RAM) and the inference time required by the models, implemented in PyTorch.

Model	Parameters (M)	RAM (GB)	Inference (ms)
VGG-16	117.5	2.63	1.4 ± 0.2
ResNet-18	11.2	0.76	2.2 ± 0.1
Inception-v3	21.8	0.91	8.2 ± 0.3
VoxNet	0.92	0.58	0.5 ± 0.1
R3D	33.1	1.11	2.1 ± 0.2
R(2+1)D	31.3	1.09	3.3 ± 0.2
PointNet	0.95	0.74	4.8 ± 0.1
PointNet ⁺⁺	0.81	1.17	226.5 ± 5.5

in this section, we report an analysis of the computational complexity of the investigated approaches. In particular, we report the number of parameters, the memory consumption and the inference speed of each method in Table 4.9. All of the deep models are implemented using the PyTorch framework [161] and tested on a computer equipped with an Intel(R) Core(TM) i7-7700K and a Nvidia GeForce GTX 1080 Ti.

The first three rows of Table 4.9 report CNNs relying on 2D input images, then voxel-based approaches are reported in the central rows and the last two rows contains the point cloud-based models. As expected, the number of parameters of 2D CNNs is correlated with the memory occupation: in this context, the VGG-16 model has the highest number of parameters and the highest RAM occupation. Nevertheless, its inference time is remarkably low, which is probably thanks to the level of optimization for the convolutional operations in the PyTorch framework [12]. The same analysis also holds for voxel-based methods. When considering PointNet and PointNet⁺⁺, the former requires a little amount of memory and a sufficiently low inference time while the latter represents an exception having a very high inference time. We believe that this is caused by the several clustering operations, still not optimized on GPUs, needed by the architecture.

4.1.3 Discussion

In this section, we summarize the main considerations that follow from the intra- and cross-dataset experiments, from the additional analysis obtained on the MultiSFace dataset and from the evaluation of the computational complexity.

First of all, we observe that, in general, approaches that rely on depth images and CNNs are limited in terms of the generalization capabilities. That is, a substantial performance drop occurs when these models are tested with depth data that differ from the training one (such as data acquired by the same depth sensor in a different setting or another sensor with the same or a different building technology). On the other hand, normal images represent the best choice in order to obtain a high level of accuracy in a cross-dataset scenario while using CNNs. However, they are employed in a minor part of literature work. Moreover, the results clearly show that point cloud-based representations and architectures are the best option in terms of generalization capabilities when the training and testing data do not belong to the same dataset (*i.e.* the data are collected in different acquisition setups). Because similar experiments are not available in the literature, the reported results can be considered as baselines for future investigation in this research field.

When considering the intra-dataset setting, the results show that the face recognition task can be carried out while using depth maps, even if they only contain geometrical information (in contrast to intensity images that contain shapes, colors and textures). However, the generalization capabilities of these architectures have still not been tested on more challenging settings, *i.e.* when the probes and the gallery set are acquired with different depth devices or in different scenarios. These types of experiment cannot be carried out using existing datasets since intra-dataset experiments contain data that are captured by a single depth sensor, while cross-dataset experiments are not possible (because the same subjects are not included in different datasets).

To this end, we collected the proposed MultiSFace dataset and reported the results obtained on it using probes and gallery sets acquired by different depth sensors and at different sensor-subject distances. These results confirm that 2D representations of depth maps, which are processed with CNNs, are not a suitable solution for cross-device and cross-distance settings. They also show, in line with previous findings, that point cloud-based

representations and architectures are the optimal solution in the majority of the tested settings. However, we want to highlight that the accuracy on the MultiSFace dataset is, without any doubt, too low, showing the challenging nature of the recognition task in these scenarios, made possible by this particular dataset. In contrast to the high recognition accuracy obtained in the single-sensor single-dataset scenario, the face recognition task carried out in the wild using several depth sensors in different acquisition settings is far from being solved. We believe that this dataset can inspire and be an interesting benchmark for future investigations regarding face recognition with depth maps that are focused on generalization capabilities over depth sensors and data.

From a computational point of view, we observe that a depth-based face verification system based on any of the analyzed architectures can easily obtain real-time speed on a GPU-equipped workstation. Moreover, the RAM usage is rather variable among different architectures, but considerably low in general when compared to the typical memory size of commercial GPUs.

4.2 Dynamic Hand Gesture Recognition

The recent introduction of affordable RGB-D devices, which couple RGB cameras with active depth sensors, has attracted the interest of the research community towards Natural User Interfaces (NUIs), in which the interaction is conveyed through the body of the user [183, 127] instead of traditional tools, like keyboards and mouse. In this context, the ability to recognize dynamic hand gestures, *i.e.* a combination of static hand poses and motion, without the use of contact-based sensors, is an enabling and crucial task.

The dynamic hand gesture recognition task is commonly tackled through the use of RNNs [82, 115], such as LSTMs [246, 27], which are able to model the temporal and sequential nature of dynamic gestures. Alternatively, other authors proposed to classify temporal sequences using 3D CNNs [254, 133], standard CNNs [50, 45] or other machine learning methods, like HMMs [113, 14] or HOG and SVM [175, 56]. The recent spread of attentive models, which are characterized by the use of the self-attention mechanism, has come with the introduction of new approaches, such as the Transformer [215], that can replace traditional recurrent modules. However, these approaches have not yet been deeply explored for the analysis of visual data and, in particular, for the dynamic hand gesture recognition task.

In this section, we propose a method to classify dynamic hand gestures based on the Transformer architecture, which was originally developed for the machine translation and language modeling tasks. We propose the use of RGB-D or active depth devices and, in particular, we show that the use of depth maps and the surface normals estimated from them leads to state-of-the-art results. In addition, we investigate the adoption of the other data streams usually provided by RGB-D sensors, *i.e.* infrared amplitude and color images, and derived data, such as optical flow. The use of lighting-invariant data sources – depth and infrared images – guarantees the applicability of the proposed method in Human-Computer Interaction systems that are able to work even in dark conditions or in presence of severe and fast lighting changes, as often occurs in the automotive context [21, 165]. Indeed, the presence of tunnels and trees or bad weather conditions can strongly influence the quality of the acquired data in this scenario. Moreover, the use of inexpensive and compact cameras, which can be easily integrated in the car cockpit, is an ideal choice to avoid obstructions to the driver’s movements or gaze. It is shown [229, 47] that the presence of a NUI-based system for the interaction with the infotainment system of a car can

significantly reduce the driver’s manual and visual distraction [15, 21] often responsible for fatal road crashes.

For these reasons, we test the proposed system on datasets collected in an automotive setting. We use two publicly released datasets, namely Nvidia Dynamic Hand Gesture [144] and Briareo, presented in Section 3.1.2. Both datasets are acquired in a realistic car simulator through several sensors placed in different position inside the car cockpit. When tested on these datasets, the proposed transformer-based architecture achieves state-of-the-art results, overcoming existing literature competitors. Moreover, the proposed method is flexible since it can be adapted to the available data types and is able to run in real-time on a dedicated graphics card. The proposed architecture is implemented in PyTorch 1.5 and the code is made available online ¹.

4.2.1 Proposed Method

In this section, we present the mathematical formulation and the transformer-based implementation of our method. The proposed model can process an input sequence of variable length and outputs the gesture classification. An overview of the architecture is shown in Figure 4.3.

Formulation

The proposed gesture recognition architecture can be defined as a function

$$\Gamma : \mathbb{R}^{m \times w \times h \times c} \rightarrow \mathbb{R}^n \quad (4.4)$$

that predicts a probability distribution over n classes from a set $S_t \in \mathbb{R}^{m \times w \times h \times c}$ of m sequential frames I , with size $w \times h$ and c channels, acquired in a time range t . In other words, the function Γ takes a sequence clip and predicts a class distribution over the considered hand gestures. The function can be decomposed in the following three components.

The first operation corresponds to a feature extraction function F applied at frame level:

$$f_t = F(S_t) \text{ where } F : \mathbb{R}^{m \times w \times h \times c} \rightarrow \mathbb{R}^{m \times k} \quad (4.5)$$

¹<https://aimagelab.ing.unimore.it/go/gesture-recognition-automotive>

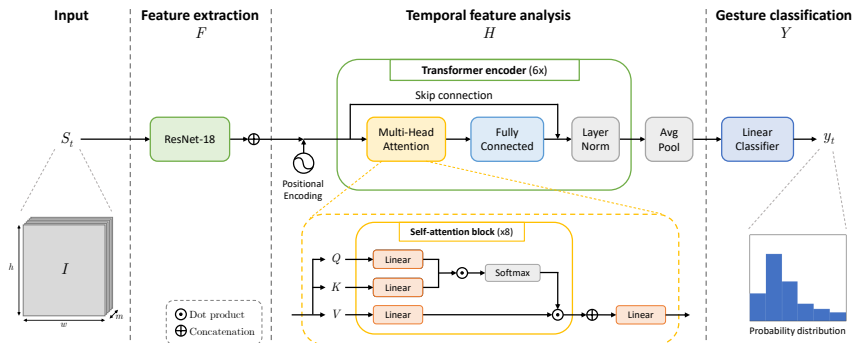


Figure 4.3: Overview of the proposed method. The temporal feature analysis module, applied on the features extracted by ResNet-18, shows the architecture of the transformer encoder and the self-attention block. Q , K , V denote the queries, keys and values of the attention mechanism.

Here, the extracted features f_t consist of m independent visual features of size k . Therefore, the function F can be defined as the concatenation of the results of a frame-level feature extractor:

$$F(S_t) = f_t^0 \oplus f_t^1 \oplus \dots \oplus f_t^m \quad \text{where } f_t^j = G(S_t^j) \quad (4.6)$$

where $G: \mathbb{R}^{w \times h \times c} \rightarrow \mathbb{R}^k$ is a function that extracts visual features from a single frame j of the sequence set S_t . \oplus denotes the concatenation operator.

The second operation is a temporal combination and analysis of the visual features extracted through F . This process can be defined as

$$h_t = H(f_t) = H(F(S_t)) \quad \text{where } H: \mathbb{R}^{m \times k} \rightarrow \mathbb{R}^l \quad (4.7)$$

where H is a temporal function that processes m feature maps of size k and outputs an aggregated feature map of size l which encodes the temporal information of S_t .

Finally, the last operation is a mapping between the extracted temporal features h_t and the n gesture classes:

$$y_t = Y(h_t) = Y(H(F(S_t))) \quad \text{where } Y: \mathbb{R}^l \rightarrow \mathbb{R}^n \quad (4.8)$$

The resulting y_t , being a probability distribution over n classes, is a vector of size n so that $\sum_{i=1}^n y_{t,i} = 1$ and $y_{t,i} \in [0, 1]$.

Implementation

In our implementation, the function Γ is a combination of multiple neural networks, defined as following.

The function F is the concatenation of the frame-level features extracted by the function G , which is implemented as ResNet-18 [77], taken from the first layer up to the last convolutional and average pooling layer. The network is designed for color images, but we adapt the first layer to work with inputs having a lower number of channels c as proposed by Molchanov et al. [144]. In practice, the convolutional kernels of the first layer are adapted to 1-channel images by summing their channels. In a similar way, they are adapted to 2-channel images by removing the third channel and rescaling the first two with a factor of 1.5.

The function H , which has to temporally combine the frames of the clip S_t , corresponds to a slightly modified Transformer module [215] followed by an average pooling at frame level. The model can handle sequences of any length by design. Formally, the module can be defined as:

$$H(x) = \text{AvgPool}(\text{Encoders}(x + PE)) \quad (4.9)$$

where $\text{AvgPool}(\cdot)$ denotes the average pooling operation over the m frames, while $\text{Encoders}(\cdot)$ corresponds to a sequence of 6 transformer encoders E , defined in the following. As proposed by Vaswani et al. [215], we add positional encodings PE to the input data as a way of including temporal information about the order of the frames into the model, which does not contain any recurrent module nor implicit definition of temporal order. Among the several positional encodings [61], we employ the proposal of Vaswani et al. [215].

Each transformer encoder can be defined as

$$E(x) = \text{Norm}(x + \text{FC}(\text{mhAtt}(x))) \quad (4.10)$$

where $\text{Norm}(\cdot)$ is a normalization layer, $\text{FC}(\cdot)$ is a sequence of two fully connected layers with 1024 units, followed by drop out (drop probability 0.1) and divided by a ReLU activation function. The multi-head attention block mhAtt is a self-attention layer that can be defined as

$$\text{mhAtt}(x) = (\text{Att}_1(x) \oplus \dots \oplus \text{Att}_8(x)) W^O \quad (4.11)$$

where

$$\text{Att}_i(x) = \text{softmax} \left(\frac{Q_i K_i}{\sqrt{d_k}} \right) V_i \quad (4.12)$$

Here, $Q_i = xW_i^Q$, $K_i = xW_i^K$, $V_i = xW_i^V$ are independent linear projections of x into a 64-d feature space, $d_k = 64$ is a scaling factor corresponding to the feature size of K_i , \oplus is the concatenation operator and W^O is a linear projection from and to a 512-d feature space.

Finally, the function Y is implemented as a fully connected layer with n hidden units followed by a softmax layer, resulting in a probability distribution over the n classes. The predicted gesture corresponds to the class with the highest probability.

We note that the proposed approach is supposed to receive a sequence of frames containing the whole gesture or be applied with a sliding-window approach. The temporal segmentation, *i.e.* the detection of the beginning and the end of each gesture, and the gesture detection, *i.e.* the distinction between gesture and no-gesture sequences, are out of the scope of this work.

Data representation

As mentioned above, we focus our investigation on the use of data produced by active depth sensors, *i.e.* depth data and infrared (amplitude) images. We include also RGB data since several depth devices available in the market consist of a combination of infrared and intensity sensors, like the Microsoft Kinect or Intel RealSense families.

In addition, we propose the use of surface normals, in which each pixel encodes the three components of the estimated surface normal in that point. From depth maps we obtain a representation containing an estimation of the surface normals, as introduced by previous works [11, 151, 9] and detailed in Section 2.1.2. Normals computed from depth maps are not frequently used in the literature, especially in the case of the hand gesture recognition task with neural architectures. Preliminary works investigate the use of surface normals for hand pose estimation [219] or human activity recognition [243, 155]. We show in the following that this representation is complementary to the common depth images and that greatly improves the overall accuracy when used in combination with the original depth data. Samples of the estimated surface normals are shown in Figure 4.4 and Figure 4.5.

In order to compare our work with literature competitors, we also compute the optical flow from consecutive RGB frames following the implementation of Farneback [55]. It is a well-known data representation that is often used to improve the performance of the proposed systems, even

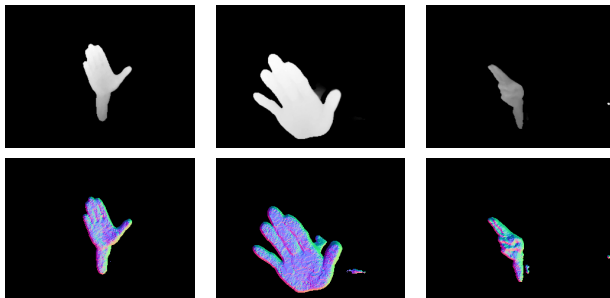


Figure 4.4: Sample depth maps (first row) and estimated surface normals (second row) from NVGestures. As shown, cameras are placed in a frontal position with respect to the driver and the noise level is low. In most of the frames, only the hand is visible.

in the hand recognition task [144, 1], thanks to its ability to provide an estimation of the magnitude and the direction of the motion of the objects (the hands in our case).

Multimodal integration

Multimodal architectures are becoming increasingly common in the literature, for a variety of different tasks. Since several input types are available from RGB-D sensors, we adopt a neural network architecture that can be easily adapted to work with a single input type or a multimodal combination of them. Specifically, the proposed architecture is able to efficiently work in a unimodal way, *i.e.* with a single input modality (color, depth, infrared, normals or optical flow). Moreover, two or more unimodal networks can be used at the same time through a late fusion approach [195] in which the predicted probability distributions of the single models are merged into a final classification score. Late fusion strategies are reported to present comparable or even better results with respect to the state-of-the-art in many computer vision tasks [240, 51]. In our case, we adopt a late fusion strategy based on the average of the intermediate scores to predict the final classification, as follows:

$$y_t = \frac{1}{N} \cdot \sum_i Y(H(F(S_{t,i}))) \quad (4.13)$$

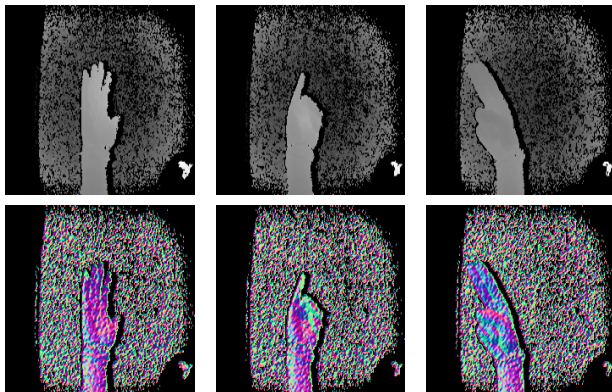


Figure 4.5: Sample depth maps (first row) and estimated surface normals (second row) from Briereo. Differently from NVGestures, this dataset is acquired placing the camera looking upwards. Moreover, a strong noise signal is present in the depth maps and, consequently, in the surface normals.

where N is the total number of tested classifiers, $S_{t,i}$ is the set of sequential frames of the i -th input type and F, H are the functions defined in Section 4.2.1. Then, $Y(H(F(S_{t,i})))$ is the probability distribution of a classifier trained and tested on a specific input type.

4.2.2 Experimental Evaluation

In this section, we present the experimental setting and the results obtained on two public datasets. Then, we compare with literature methods and discuss the obtained results. Since surface normals can be considered as a different representation of depth maps, we include competitors relying on RGB-D data. In addition to the core tests with depth images and estimated surface normals, we test on color and other modalities to compare with existing literature methods.

Datasets

Being interested in the usage of depth or RGB-D sensors and in the automotive environment, in which the lighting invariance is a key factor, we test our approach on two datasets that contain color, infrared and depth

data and were collected in a car simulator: the Nvidia Dynamic Hand Gesture dataset (or simply NVGestures) [144] and the Briareo dataset, presented in Section 3.1.2. Visual samples, which include depth maps and estimated surface normals, are respectively shown in Figure 4.4 and Figure 4.5 for NVGestures and Briareo. For more information about the employed datasets, please refer to Section 3.1.2 and Section 3.2.2.

Model training

We train and test the model with fixed-length clips of 40 frames extracted from the dataset sequences around the center of the gesture. We empirically set this input size, but the proposed model can potentially analyze sequences of any length thanks to its flexible design. For the NVGestures dataset, we extract the 80 central frames around the gesture and sample them to obtain 40 equidistant frames. For the Briareo dataset, which has a lower frame rate, we select the 40 frames containing the gesture movement.

Each input data is normalized individually to obtain zero mean and unit variance input, with the exception of the surface normals that are normalized to have unit-magnitude and are contained in the range $[-1, 1]$. Then, frames are cropped to 224×224 pixels as required by the chosen frame-level feature extractor, *i.e.* ResNet-18. As data augmentation, we apply random rescale, with rescale factor in the range $[0.8, 1.2]$, random crop and $\pm 15^\circ$ random rotation in order to avoid overfitting.

The ResNet-18 architecture is initialized with weights pre-trained on ImageNet [40] while the remaining of the architecture is trained from scratch. The architecture is trained end-to-end using the Adam optimizer [102], minimizing the categorical cross entropy loss. We use a mini-batch size of 8 video samples, learning rate $1e^{-4}$, weight decay $1e^{-4}$ and random dropout. We apply the early stopping based on the accuracy on the validation set, following the official dataset splits.

A different model is trained for each modality and multiple modalities are combined at prediction level with the late fusion approach presented in Section 4.2.1. Empirically, we find that other types of fusion, *e.g.* mid and early fusion, results in overfitting on the training set, in line with what found by Molchanov et al. [144].

Table 4.10: Unimodal results on NVGestures [144]. Previous results are taken from the respective papers and from [144, 1]. † indicates models pre-trained on Kinetics [99], in addition to ImageNet [40].

Method	Modality	Accuracy
color	Spat. st. CNN [193]	54.6%
	iDT-HOG [221]	59.1%
	Res3ATN [46]	62.7%
	C3D [206]	69.3%
	R3D-CNN [144]	74.1%
	Ours	76.5%
depth	I3D [29]†	78.4%
	SNV [238]	70.7%
	C3D [206]	78.8%
	R3D-CNN [144]	80.3%
	Ours	83.0%
infrared	I3D [29]†	82.3%
	Ours	83.0%
infrared	R3D-CNN [144]	63.5%
	Ours	64.7%
flow	iDT-HOF [221]	61.8%
	Temp. st. CNN [193]	68.0%
	Ours	72.0%
	iDT-MBH [221]	76.8%
	R3D-CNN [144]	77.8%
	I3D [29]†	83.4%
normals	Ours	82.4%
color	<i>Human</i> [144]	88.4%

Results on NVGestures

We analyze here the performance on the NVGestures dataset.

Table 4.10 compares our method to the literature in the unimodal case, *i.e.* when a single input is fed into the model. Focusing on depth data, the proposed approach achieves state-of-the-art results when depth maps are the only used input. A similar high accuracy is also achieved using surface normals as input, revealing that normals are a discriminative representation for the hand gesture recognition task, even though no competitors are currently available. Similarly, the infrared modality overcomes the

Table 4.11: Multimodal results on NVGestures [144] using several combinations of modalities. # refers to the number of used modalities.

#	Modality	Accuracy
1	infrared (ir)	64.7%
	color	76.5%
	normals	82.4%
	depth	83.0%
2	color + ir	79.0%
	depth + ir	81.7%
	normals + ir	82.8%
	color + depth	84.6%
	color + normals	84.6%
	depth + normals	87.3%
3	color + ir + depth	85.3%
	color + ir + normals	85.3%
	color + depth + normals	86.1%
	depth + normals + ir	87.1%
4	color + depth + normals + ir	87.6%

competitor, even if the absolute accuracy is lower than other modalities. On the other remaining modalities, *i.e.* color and optical flow, our method achieves comparable accuracy to the I3D method [29, 1]. However, we note this method is pre-trained on ImageNet [40] (as our feature extractor) and on Kinetics [99], which is a large dataset of action recognition in videos. We hypothesize that the slight gap between this and our method can be due to this pre-training step, which was not available for the other types of the exploited data. *Human* denotes the recognition accuracy obtained by humans looking at color videos [144].

Moving from the unimodal to the multimodal case, we show in Table 4.11 a thorough analysis of the possible multimodal combinations, following the late-fusion approach reported in Section 4.2.1. The results are grouped by number of employed modalities and ordered by accuracy. It can be seen that, in general, the proposed approach benefits from the multimodal integration. Moreover, the best performing methods in each group are those using a combination of depth and surface normals as input data, confirming that the partial 3D data obtained by the depth sensors contains

Table 4.12: Multimodal results on NVGestures [144], comparison with competitors. Previous results are taken from the respective papers and from [144, 1]. † indicates models pre-trained on Kinetics [99], in addition to ImageNet [40], while * shows models pre-trained on the Jester gesture dataset [140].

Method	Modality	Accuracy
Two-st. CNNs [193]	color + flow	65.6%
iDT [221]	color + flow	73.4%
R3D-CNN [144]	color + flow	79.3%
R3D-CNN [144]	color + depth + flow	81.5%
R3D-CNN [144]	color + depth + ir	82.0%
R3D-CNN [144]	depth + flow	82.4%
R3D-CNN [144]	all	83.8%
8-MFFs-3f1c [105]*	color + flow	84.7%
I3D [29]†	color + depth	83.8%
I3D [29]†	color + flow	84.4%
I3D [29]†	color + depth + flow	85.7%
MTUT _{RGB-D} [1]†	color + depth	85.5%
MTUT _{RGB-D+flow} [1]†	color + depth	86.1%
MTUT _{RGB-D+flow} [1]†	color + depth + flow	86.9%
Ours	depth + normals	87.3%
Ours	color+depth+normals+ir	87.6%
<i>Human [144]</i>	<i>color</i>	<i>88.4%</i>

discriminative information for the gesture recognition task.

We highlight that the combination of depth images and surface normals leads to a remarkable accuracy of 87.3%. This result confirms that these two modalities are complementary and their combination greatly improves the overall accuracy compared to the usage of a single modality (which scores 83.0% for the depth and 82.4% for the surface normals). Combining additional modalities (color and infrared) the accuracy is slightly improved, reaching 87.6%.

We also compare our method in the multimodal setting with state-of-the-art approaches, as shown in Table 4.12. Among other methods that exploit several data types, our approach obtains state-of-the-art accuracy (87.3%)

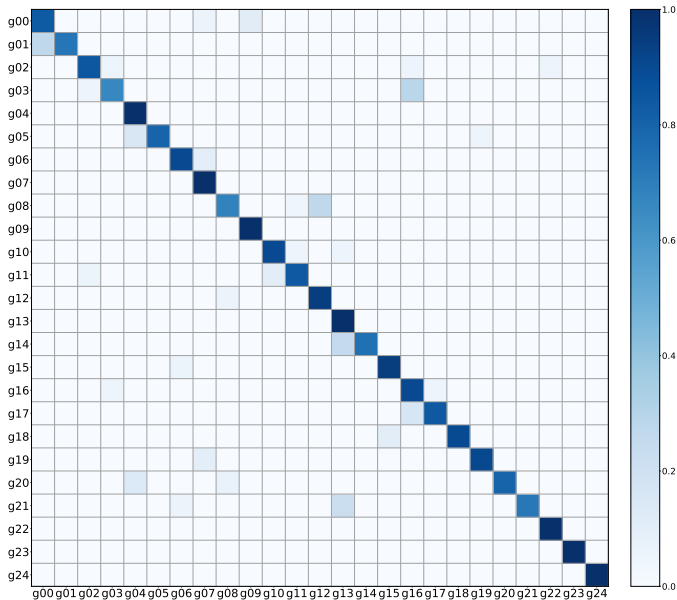


Figure 4.6: Confusion matrix for the best performing multimodal combination (fusion of color, depth, normals, ir) on NVGestures. Best viewed in color.

using only depth data and surface normals, which derive from a single depth sensor. Therefore, the whole system can depend on a single depth or RGB-D device and can run in real time, as will be shown in Section 4.2.2. In addition, our method, combining a broader set of modalities (*i.e.* color, depth, surface normals, infrared), slightly improves the overall accuracy, reaching a 87.6% recognition rate. A wide set of other methods make use of the optical flow, but still perform worse than our method. However, we note that the computation of the optical flow on the whole sequence of frames heavily affects speed performance, hindering the achievement of real-time computation.

Finally, we show the confusion matrix for the best performing multimodal combination (*i.e.* color + depth + normals + ir) in Figure 4.6. Most of the gestures are correctly classified, but some errors caused by

Table 4.13: Unimodal and multimodal results obtained on Briareo. # refers to the number of used modalities.

#	Modality	Accuracy
1	color	90.6%
	depth	92.4%
	ir	95.1%
	normals	95.8%
2	color + depth	94.1%
	depth + ir	95.1%
	color + ir	95.5%
	depth + normals	96.2%
	color + normals	96.5%
	ir + normals	97.2%
3	color + depth + ir	95.1%
	color + depth + normals	95.8%
	color + ir + normals	96.9%
	depth + ir + normals	97.2%
4	color + depth + ir + normals	96.2%

confusion between pairs of gestures are also visible. As expected, the model sometimes swaps similar – in terms of hand poses or motion – gestures, such as “move hand/fingers left/right”, “opening” and “shaking” hand and “push hand down/towards the camera” .

Results on Briareo

Table 4.13 presents the results of the unimodal and the multimodal setting on the Briareo dataset. The results are grouped by number of employed modalities and ordered by accuracy.

Considering the unimodal case, the surface normals obtains the highest accuracy, reaching 95.8%, outperforming the results using other modalities. This confirms that surface normals estimated from depth are an informative and discriminative representation for the hand gesture recognition task. Similarly, the infrared source achieves a high accuracy, probably due to the position of the infrared sensor, close to the hand. In point of fact, differently from the previous dataset, the infrared data in Briareo corresponds to the

Table 4.14: Comparison with the state-of-the-art methods tested on Briareo.

Method	Modality	Accuracy
C3D-HG [133]	color	72.2%
C3D-HG [133]	depth	76.0%
C3D-HG [133]	ir	87.5%
LSTM-HG [133]	3D joint features	94.4%
Ours	normals	95.8%
Ours	depth + normals	96.2%
Ours	ir + normals	97.2%

infrared amplitude collected by the depth sensor. Thus, depth maps and infrared images share the same point and field of view.

The combination of multiple modalities, with the late fusion approach presented in Section 4.2.1, slightly improves the overall results. The fusion of infrared and normals results in an overall accuracy of 97.2% which is the highest result. While the combination of surface normals with infrared and depth increases the combined accuracy, the usage of color data does not provide significant gains.

Table 4.14 compares our method in the multimodal setting with state-of-the-art approaches. The proposed approach obtains state-of-the-art accuracy using only infrared data and surface normals, which derive from a single active depth sensor. Even with the usage of a single modality, *e.g.* surface normals, our method outperforms the literature competitors by a clear margin. Indeed, it performs better than methods based on recurrent networks (LSTMs) and 3D joint features (computed by the Leap Motion SDK), which require additional computation. Also in this case, the whole system requires a single active depth device and can run in real time, as shown in the next section.

Computational complexity

We assess the computational requirements of our and other architectures in terms of number of parameters, inference time on a single GPU and required RAM on the graphics card. We test them on a workstation with an Intel Core i7-7700K and a Nvidia GeForce GTX 1080 Ti. As shown in Table 4.15, our method has fewer parameters, faster inference speed and

Table 4.15: Performance analysis of the proposed method. Specifically, we report the number of parameters, the inference time and the amount of memory (RAM) needed to run the system.

Model	Parameters (M)	Inference (ms)	RAM (GB)
R3D-CNN [144]	38.0	30	1.3
C3D-HG [133]	26.7	55	1.0
Ours (1 modality)	24.3	26.7	1.8
Ours (2 modalities)	48.6	61.7	3.0
Ours (4 modalities)	97.2	108.3	5.3

comparable memory usage when used with a single modality. When applied on multiple modalities, running in parallel on the same hardware, the proposed approach still maintains real time speed and acceptable memory usage, both in case of 2 modalities and in case of 4 modalities.

4.2.3 Discussion

Results presented in the previous section confirm that the proposed architecture, composed of a low-level feature extractor and the temporal aggregation module based on the Transformer, is a successful method to tackle the recognition of dynamic hand gestures. In addition, the approach is flexible in terms of both sequence length and input modality. Indeed, multiple modalities can be combined with a trivial but effective late-fusion approach, yielding to higher scores compared to using a single modality. Moreover, the experimental evaluation shows that the proposed method is suitable for being used in a real-world in-car infotainment system. In fact, it can be employed with illumination-invariant data streams, such as depth maps and infrared images; it can work using a single depth sensor (that can provide multiple data streams); the inference can be computed in real time on dedicated hardware (such as the Nvidia Jetson boards). We identified, however, that pairs of "symmetric" gestures are occasionally confused, despite the temporal flow is explicitly encoded into the transformer-based module. This fact shows that the analysis of the temporal progression of the gesture can still be improved.

4.3 3D Human Pose Estimation and Refinement

In this section, we firstly present a method for the 2D human pose estimation from depth maps. Then, we propose a modular framework for the 3D human pose refinement from an initial 2D human pose estimation and a depth map. Finally, we discuss the obtained results and their implications.

4.3.1 2D Human Pose Estimation from Depth Maps

In recent years, the task of estimating the human pose has been widely explored in the computer vision community. Many deep learning-based algorithms that tackle the 2D human pose estimation have been proposed [226, 26, 234, 198] along with a comprehensive set of annotated datasets, collected in real world [5, 123, 69] or in simulations [214, 31]. However, the majority of these works and data collections are based on standard intensity images (*i.e.* RGB or gray-level data) while datasets and algorithms based only on depth maps have been seldom explored, even though this kind of data contains fine 3D information and can be successfully used in particular settings, such as the automotive one [216, 19].

Leveraging the fine annotations collected in the Watch-R-Patch dataset, presented in Section 3.1.3, and the groundbreaking work of Cao et al. [26], we present a deep learning-based architecture that estimates the human pose directly on depth images. The model is trained combining the original Watch-n-Patch dataset with the manually refined annotations of Watch-R-Patch, obtaining remarkable results. Similar to Shotton et al. [189], the proposed system can run in real time, at more than 180 fps.

Proposed method

In the development of the deep architecture, we focus on both accuracy, in terms of mean Average Precision (mAP), and speed, in terms of frames per second (fps). To guarantee high precision, we develop a deep neural network derived from the work of Cao et al. [26] while we do not include the Part Affinity Fields (PAF) module to guarantee high fps, even on cheap/low-power hardware (for details about PAF, see [26]).

Network architecture. An overview of the proposed architecture, which is implemented using Pytorch [162], is shown in Figure 4.7.

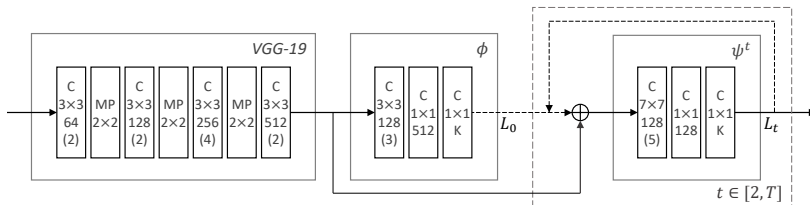


Figure 4.7: Overview of the proposed method. Each block contains its type (C: convolution, MP: max pool), kernel size, no. of feature maps and no. of repetitions (if higher than 1). In our experiments, $K = 21$ and $T = 6$.

The first part of the model is a VGG-like feature extractor which comprises the first 10 layers of VGG-19 [194] and two layers that gradually reduce the number of feature maps to the desired value. In contrast to previous work [26], we do not use ImageNet [40] pre-trained weights and we train these layers from scratch along with the rest of the architecture.

The feature extraction module is followed by a convolutional block that produces an initial coarse prediction of human body joints analyzing the image features extracted by the previous block. The output of this module can be expressed as

$$\mathbf{P}^1 = \phi(\mathbf{F}, \theta^1) \quad (4.14)$$

where \mathbf{F} are the feature maps computed by the feature extraction module and ϕ is a parametric function that represents the first convolutional block of the architecture with parameters θ^1 . Here, $\mathbf{P}^1 \in \mathbb{R}^{k \times w \times h}$.

Then, a multi-stage architecture is employed. A common convolutional block is sequentially repeated $T - 1$ times in order to gradually refine the body joint prediction. At each stage, this block analyzes the concatenation of the features extracted by the feature extraction module and the output of the previous stage, refining the previous prediction. The output at each step can be defined as

$$\mathbf{P}^t = \psi^t(\mathbf{F} \oplus \mathbf{P}^{t-1}, \theta^t) \quad \forall t \in [2, T] \quad (4.15)$$

where \mathbf{F} are the feature maps computed by the feature extraction module, \mathbf{P}^{t-1} is the prediction of the previous block, \oplus is the concatenation operation and ψ^t is a parametric function that represents the repeated convolutional block of the architecture with parameters θ^t . As in the previous case, $\mathbf{P}^t \in \mathbb{R}^{k \times w \times h}$.

Training procedure. The architecture is trained in an end-to-end manner applying the following objective function

$$L^t = \sum_{k=1}^K \alpha_k \cdot \sum_{\mathbf{p}} \|\mathbf{P}_k^t(\mathbf{p}) - \mathbf{H}_k(\mathbf{p})\|_2^2, \quad (4.16)$$

where K is the number of considered body joints, α_k is a binary mask with $\alpha_k = 0$ if the annotation of joint k is missing, t is the current stage and $\mathbf{p} \in \mathbb{R}^2$ is the spatial location. Here, $\mathbf{P}_k^t(\mathbf{p})$ represents the prediction at location \mathbf{p} for joint k while $\mathbf{H}_k \in \mathbb{R}^{w \times h}$ is the ground-truth heatmap for joint k , defined as

$$\mathbf{H}_k(\mathbf{p}) = e^{-\|\mathbf{p} - \mathbf{x}_k\|_2^2 \cdot \sigma^{-2}} \quad (4.17)$$

where $\mathbf{p} \in \mathbb{R}^2$ is the location in the heatmap, $\mathbf{x}_k \in \mathbb{R}^2$ is the location of joint k and σ is a parameter to control the Gaussian spread. In our experiments, we set $\sigma = 7$. The overall objective function can be expressed as $L = \sum_{t=1}^T L^t$ where T is the number of stages. In our experiments, $T = 6$. As outlined by Cao et al. [26], applying the supervision at every stage of the network mitigates the vanishing gradient problem and, along with the sequential refining of the body joint prediction, leads to a faster and more effective training of the whole architecture.

The network is trained in two steps. In the first stage, the original body joint annotations of Watch-n-Patch are employed to train the whole architecture from scratch. It is worth noting that the Watch-n-Patch body joints are inferred by the Microsoft Kinect SDK, which makes use of a random forest-based algorithm [189]. In the second stage, the network is fine-tuned using the training set of Watch-R-Patch. During this phase, we test different procedures. In the first tested procedure, the whole architecture is fine-tuned, in the second one the feature extraction block is frozen and not updated, while in the last procedure all the blocks but the last one are frozen and not updated.

During both training and fine-tuning, we apply data augmentation techniques and dropout regularization to increase the generalization of the model. In particular, we apply random horizontal flip, crop (extracting a portion of 488×400 pixels from the original image having size 512×424 pixels), resize (to the crop dimension) and rotation ($\pm 4^\circ$). Dropout is applied between the first convolutional block and each repeated block.

In our experiments, we employ the Adam optimizer [102] with $\alpha = 0.9$, $\beta = 0.999$ and weight decay set to $1 \cdot 10^{-4}$. During the training phase

Table 4.16: Comparison of the mAP reached by different methods on the Watch-R-Patch dataset.

Metric	Shotton et al. [189]	Ours _{orig}	Ours _{last}	Ours _{blk}	Ours
AP ^{OKS=0.50}	0.669	0.845	0.834	0.894	0.901
AP ^{OKS=0.75}	0.618	0.763	0.758	0.837	0.839
mAP	0.610	0.729	0.726	0.792	0.797

and the fine-tuning step, we use a learning rate of $1 \cdot 10^{-4}$ and apply the dropout regularization with 0.5 as drop probability.

Experimental results

Evaluation procedure. We adopt an evaluation procedure that follows the one of the Microsoft COCO Keypoints Challenge [148].

In details, we employ the mean Average Precision (mAP) to assess the quality of the human pose estimations compared to the ground-truth positions. The mAP is defined as the mean of 10 Average Precision calculated with different Object Keypoint Similarity (OKS) thresholds:

$$\text{mAP} = \frac{1}{10} \sum_{i=1}^{10} \text{AP}^{\text{OKS}=0.45+0.05i} \quad (4.18)$$

The OKS is defined as

$$\text{OKS} = \frac{\sum_i^K [\delta(v_i > 0) \cdot \exp \frac{-d_i^2}{2s^2k_i^2}]}{\sum_i^K [\delta(v_i > 0)]} \quad (4.19)$$

where d_i is the Euclidean distance between the ground-truth and the predicted location of the keypoint i , s is the area containing all the keypoints and k_i is defined as $k_i = 2\sigma_i$. Finally, v_i is a visibility flag: $v_i = 0$ means that keypoint i is not labeled while $v_i = 1$ means that keypoint i is labeled. The values of σ depend on the dimension of each joint of the human body. In particular, we use the following values: $\sigma_i = 0.107$ for the spine, the neck, the head and the hip joints; $\sigma_i = 0.089$ for the ankle and the foot joints; $\sigma_i = 0.087$ for the knee joints; $\sigma_i = 0.079$ for the shoulder joints; $\sigma_i = 0.072$ for the elbow joints; $\sigma_i = 0.062$ for the wrist and the hand joints.

Table 4.17: Comparison of the per-joint mAP on the Watch-R-Patch dataset.

Joint	Shotton et al. [189]	Ours _{orig}	Ours
SpineBase	0.832	0.841	0.905
SpineMid	0.931	0.911	0.935
Neck	0.981	0.975	0.978
Head	0.971	0.961	0.962
ShoulderLeft	0.663	0.673	0.819
ElbowLeft	0.490	0.635	0.772
WristLeft	0.456	0.625	0.677
HandLeft	0.406	0.599	0.680
ShoulderRight	0.538	0.547	0.782
ElbowRight	0.454	0.618	0.748
WristRight	0.435	0.642	0.727
HandRight	0.412	0.641	0.712
HipLeft	0.646	0.766	0.824
KneeLeft	0.494	0.743	0.788
AnkleLeft	0.543	0.771	0.800
FootLeft	0.497	0.743	0.801
HipRight	0.696	0.778	0.860
KneeRight	0.493	0.670	0.763
AnkleRight	0.508	0.630	0.648
FootRight	0.388	0.605	0.605
SpineShoulder	0.969	0.942	0.955

Results. Following this procedure, we perform experimental evaluations to assess the quality of the proposed method and the Watch-R-Patch dataset. Table 4.16 shows the results of the proposed method under different training settings in terms of mean Average Precision, using the Watch-R-Patch dataset. In particular, Ours_{orig} identifies the accuracy obtained by our architecture after training on the original Watch-n-Patch dataset. As expected, when trained on the Kinect annotations, our model is capable of learning to predict human body joints accordingly to the method of Shotton et al. [189], reaching a remarkable mAP of 0.777 on the Watch-n-Patch testing set. Evaluating the performance of our method

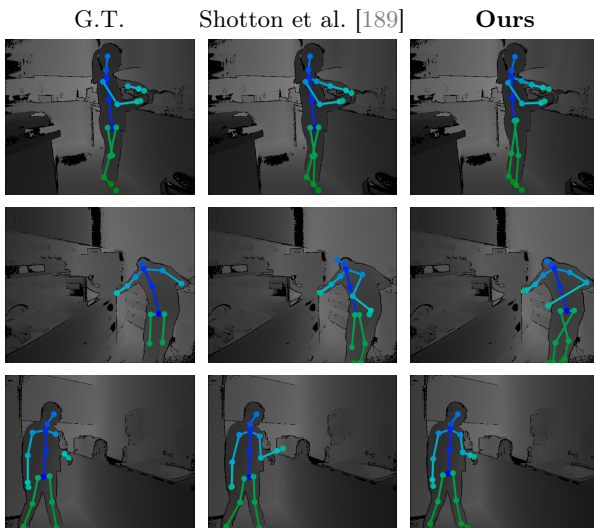


Figure 4.8: Qualitative results on the Watch-R-Patch dataset (kitchen).

on Watch-R-Patch, the model reaches a mAP of 0.729, outperforming the Shotton et al.’s method with an absolute margin of 0.119. It is worth noting that our method is trained on the Kinect annotations only, but the overall performance on the manually annotated sequences is considerably higher than the one obtained by the method of Shotton et al. [189]. We argue that the proposed architecture has better generalization capabilities than the method proposed in [189], even if it has been trained on its predictions. This is supported by the higher mAP when tested on scenes with fine body joint annotations.

We also report the results obtained applying different fine-tuning procedures in the same Table. In particular, we firstly train the proposed network on the original Watch-n-Patch annotations. Then, we fine-tune the model with the Watch-R-Patch annotations updating different parts of the architecture. Ours_{last} correspond to the experiment where we freeze the parameters of all but the last repeated block, which means updating only the parameters θ^6 of the last convolutional block ψ^6 . In Ours_{blk} , we freeze the parameters of the feature extraction block, *i.e.* only the parameters

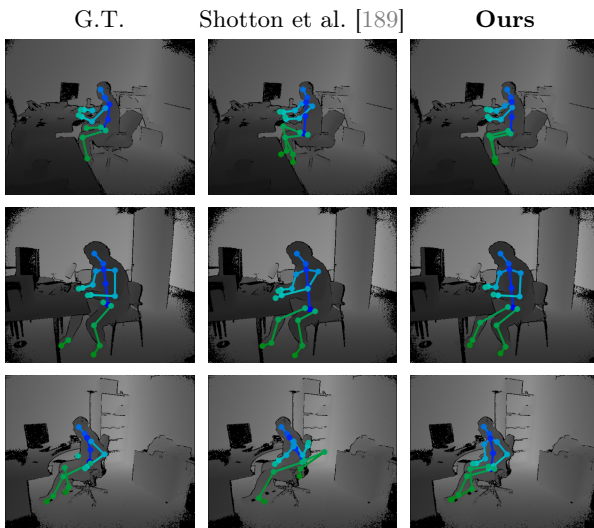


Figure 4.9: Qualitative results on the Watch-R-Patch dataset (office).

θ^t of ϕ and ψ^t are updated. We also fine-tune the whole network in the experiment Ours. As shown in Table 4.16, fine-tuning the whole architecture leads to the highest scores. The proposed model, trained on the original Watch-n-Patch dataset and fine-tuned on the presented annotations, reaches a remarkable mAP of 0.797, outperforming previous methods with an absolute gain of 0.187.

Finally, we report the per-joint mAP scores in Table 4.17. As it can be observed, the proposed method outperforms the competitor and the baseline in nearly every joint prediction, confirming the effectiveness of the model and the employed training procedure. Qualitative results are reported in Figure 4.8 and Figure 4.9. The model is able to run in real time (5.37ms per inference, corresponding to roughly 186fps) on a workstation equipped with an Intel Core i7-6850K and a GPU Nvidia GeForce GTX 1080 Ti.

4.3.2 3D Human Pose Refinement from Depth Maps

As discussed in the previous section, the Human Pose Estimation (HPE) from images is a crucial and enabling task in many vision-based applications, like Action Recognition [169, 14] and People Tracking [28]. Recently, many methods based on deep learning architectures [26, 198, 252] have in turn improved the accuracy in joint detection and localization on intensity images, achieving stunning results. Encouraged by the seminal work of Shotton et al. [189] developed for depth images, the research on marker-less human pose estimation is now more focused on RGB images. The combination of effective deep learning approaches (*e.g.* Convolutional Neural Networks) and huge datasets of RGB images (*e.g.* COCO [123]) have led to impressive performance, in terms of accuracy, computational load and generalization capabilities. Nowadays, it is possible to obtain a reliable localization of body joints even in presence of challenging situations such as occlusions, cluttered backgrounds, low-quality images. The pose is usually provided in 2D image coordinates, thus without the third coordinate (referred as the depth or z -value) and lacking any metric information.

In this section, we focus on applications that require an extremely precise estimation of the 3D position of each joint. Taking into account, for instance, the automotive field [133, 21], the configuration of some car parameters could be set depending on anthropometric measures of the driver and the passenger. A vision-based automatic system could be an excellent solution in this regard, as shown in Section 4.4.

Some preliminary works [142, 35] proposed methods to recover a complete 3D pose from RGB images with promising results. However, even though these methods predict a good estimation of the pose, they fail to recover the correct positioning in the camera space as well as the real scale of the body [35]. Thus, the errors will affect the computation of the corresponding measures of body parts and limbs (*e.g.* the exact height of person or the length of arms and legs). In these cases, depth sensors are a valid solution in place of traditional cameras. Indeed, depth cameras are more and more widespread, miniaturized and cheap; they have been recently integrated in some embedded and mobile devices; and, in particular, they capture 3D information of the scene. Thus, depth sensors can be a suitable and effective solution in place of or in addition to RGB cameras.

With this in mind, we propose to combine off-the-shelf 2D Human Pose Estimation methods with the 3D information provided by depth cameras

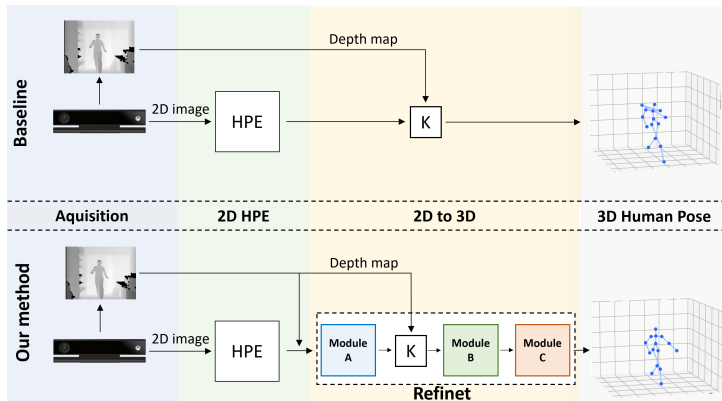


Figure 4.10: A baseline method for the 3D human pose estimation from depth maps compared to the proposed one, called RefiNet. K is the mapping operation between 2D and 3D coordinates, requiring camera calibration parameters and depth values.

in the form of depth maps. Aware that recent 2D pose estimation methods [26, 198, 252] achieve remarkable accuracy and real-time performance, we propose *RefiNet*, a modular framework that recovers an accurate 3D human pose estimation in camera space from a coarse 2D human pose and a depth map. In particular, RefiNet is a multi-stage system that regresses a precise 3D human pose through a sequential refinement of an approximate 2D estimation on a depth map. It is composed of three independent modules, each one specialized in a particular type of refinement and data representation, that can be independently enabled or disabled. Thanks to the adopted training procedure, the method does not rely on any specific 2D HPE model. Thus, the initial 2D pose can be obtained exploiting any existing 2D human pose estimation system.

The source code of RefiNet is publicly released².

Proposed method

RefiNet is a multi-stage modular framework composed of three different modules that, given as input a depth image and a set of 2D image coordinates

²<https://aimagelab.ing.unimore.it/go/3d-human-pose-refinement>

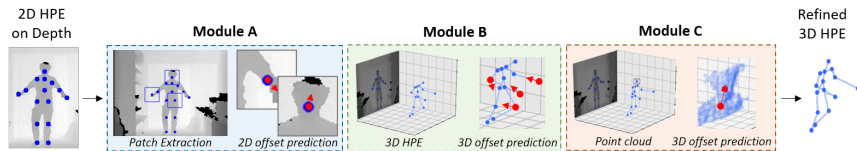


Figure 4.11: Overview of the modules that compose the RefiNet framework. Module A analyzes 2D depth patches, extracted from depth maps. Module B works directly on the 3D skeleton while module C processes point clouds computed around individual joints.

of the body joints, outputs a refined and accurate 3D human pose in camera space, *i.e.* in the absolute 3D camera coordinate system. Figure 4.10 shows a visual overview of RefiNet, compared to a conventional baseline model that directly outputs a 3D pose sampling the z coordinate from the depth map, without any refinement procedure. For the sake of clarity and ease of comprehension, the reported schema includes the generation of an initial 2D estimation, discussed in the following. Regardless of the method used for the initial estimation, the RefiNet framework refines the predicted joints and outputs an accurate 3D human pose, expressed as a set of 3D joints in the camera space, *i.e.* in the absolute 3D camera coordinate system.

RefiNet is composed of three independent modules, here referred as Module A, B, C and detailed in the following. The refinement pipeline is also depicted in Figure 4.11. During the training phase, each module is trained independently, *i.e.* the output of the previous module is not required to train the following one. As an alternative, Gaussian noise is added to the ground-truth annotations and these noisy joints are used as training data. In this way, each module is able to refine the noisy pose given as input during the testing phase without being dependent from the previous module.

Initial 2D pose estimation. As mentioned above, RefiNet requires an initial 2D human pose estimation on a depth image. This initial body pose can be computed using any off-the-shelf pose estimator applied on a 2D image, represented, for instance, by an RGB image, a depth map (encoded as grey-level image) or an IR amplitude map. Supposing the use of well-known human pose estimators trained on RGB datasets (such as COCO [123] and MPII Human Pose [5]), the RGB images would ensure

the best results, but not all the sensors and datasets provide it along with the depth channel. Moreover, coordinate translation and parallax issues between the RGB channel and the depth one should be taken into account. On the other hand, the depth and the IR amplitude channel are aligned by definition, but the pose estimation methods may perform worse or even not work on these kinds of data. Therefore, in our experiments we train 2D state-of-the-art methods on depth images from scratch, similarly to what has been presented in the previous Section 4.3.1, and use them as providers of the initial 2D pose for the proposed framework.

Once the 2D estimation is computed and mapped in the depth map space, each 2D joint location is associated to a specific depth value. Thus, joints can be converted into the camera-space 3D coordinates using the 2D joint location, the depth value and the camera calibration parameters. However, the 3D pose obtained with this approach is always an approximation: even when the 2D estimation is correct, the resulting 3D joints would lie on the body surface rather than inside the human body and may be affected by errors due to occlusions and noise. This baseline approach is depicted in Figure 4.10 (top). To overcome these limitations, we propose the use of RefiNet, shown in Figure 4.10 (bottom).

General training procedure. All the modules of RefiNet are trained following a similar approach. Each module is individually trained and is completely independent of the others. In fact, it requires only ground-truth body poses to be trained. Errors, by means of Gaussian noise, are added to the annotated joints, that are then used as input. This technique simulates the presence of errors during the pose prediction procedure. As a result, each module learns to remove noise from joint locations and to regress the original accurate human pose. This approach is especially important for Module A, which is indeed independent of any specific 2D human pose estimation used to regress the initial pose.

We adopt the same loss function L_o for the training of every module, *i.e.* the mean squared error between the predicted and the ground truth offset for each body joint. In addition, a mask is applied in order to ignore non-visible joints in the loss computation:

$$L_o = \frac{1}{n} \sum_{i=1}^n W^i \cdot \|\delta^i - t^i\|_2^2 \quad (4.20)$$

where n is the number of joints of the skeleton and $W^i \in \{0, 1\}$ is the

visibility mask for the i -th joint (which is 1 if it is visible, 0 otherwise). For each joint, δ^i and t^i are the predicted and the ground truth displacements, whose form differs for each specific module.

All models are trained with Adam [102] as optimizer and the learning rate set to 0.001, along with batch normalization and dropout.

Module A: 2D patch-based refinement. The first module of the framework refines the 2D human pose exploiting visual cues computed on depth maps. The input of the module are a set of body joints, expressed in (x, y) coordinates and the corresponding depth image. A depth map patch is cropped around each body joint and used as input of the deep network described below, which outputs a 2D offset with respect to the input coordinates. The offset is represented as a displacement vector $\delta = (\delta_x, \delta_y)$ which denotes the displacement of each joint with respect to its initial position. Indeed, considering the input coordinates (x, y) , the refined coordinates of each joint are simply computed as $x + \delta_x, y + \delta_y$, *i.e.* summing the input coordinates and the predicted offset. In this way, Module A is able to correct small errors in the 2D joint predictions. It is worth note that a small error in terms of 2D coordinates on the depth image could highly influence the sampled z -value, resulting in an extremely inaccurate 3D skeleton. A visual example of this case is shown in Figure 4.15: the z -value of the left elbow after Module A (Figure 4.15b) is more accurate than the initial one (Figure 4.15a).

The deep network of Module A is based on 3 different blocks. The first block takes the depth-image patches as input and extracts features through a single 7×7 convolutional layer with 64 feature maps. Then, the spatial dimension is reduced by applying a max pooling layer with stride $s = 2$. The second block is composed of 2 residual layers [77] with 64 and 128 feature maps and stride $s = 2$. An average pooling layer is then used to aggregate the feature maps. Finally, a series of fully connected layers with 256, 256, 2 hidden units regresses the 2D joint displacement from the averaged deep features. From a general point of view, Module A learns to predict 2D coordinate displacements for each patch independently.

Given an input joint in (x, y) coordinates, a squared bounding box (patch) or 40×40 pixels and centered in (x, y) is extracted from the depth map. If needed, patches are padded accordingly to the joint location and the width and height of the depth image. Each patch is then normalized to obtain a zero-mean unit-variance tensor that is fed to the network. During training, we directly apply random Gaussian noise to the input 2D

joints. We investigate the influence of the standard deviation σ of the error distribution in the experimental section.

Module B: skeleton-based refinement. The second module of the framework converts the 2D joint coordinates into the 3D camera space, *i.e.* the 3D real-world camera coordinates, and refines the 3D human pose using only the 3D skeleton. The input of Module B are the 2D (x, y) joint coordinates predicted in the depth map space while the output are the 3D coordinates in camera space. As first step, the bidimensional (x, y) input is converted in real-world coordinates (x_C, y_C, z_C) using the camera calibration parameters $K = \{f_x, f_y, c_x, c_y\}$, :

$$\begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} = \begin{bmatrix} (x - c_x) \cdot \frac{z}{f_x} \\ (y - c_y) \cdot \frac{z}{f_y} \\ z \end{bmatrix} \quad (4.21)$$

where z is the value of the depth map sampled in (x, y) , f_x and f_y are the focal lengths, c_x and c_y the coordinates of the optical center. To mitigate the effect of noise and missing depth data, the sampling of z is performed by calculating the median value within a 3×3 neighborhood centered in (x, y) . Then, the 3D human skeleton, expressed as the set of body joints in camera space, is fed to the deep model described below. Similarly to Module A, an offset is regressed for each body joint, in order to move the joints from the incorrect location to the most plausible one. Each predicted offset is a three-dimensional displacement vector $\delta = (\delta_x, \delta_y, \delta_z)$ between the location (x_C, y_C, z_C) of each input joint, expressed in camera-space coordinates, and the refined position.

The network, inspired by the successful work of Martinez et al. [138], is composed of a sequence of 4 blocks. The input block is a fully connected layer with 1024 units. The layer is followed by two residual blocks, each containing 2 fully connected layers with 1024 units. The output block corresponds to a fully connected layer with $n \times 3$ units, where n is the number of joints of the skeleton. Each fully connected layer consists of a linear layer, a batch normalization layer and ReLU activation.

As previously mentioned, we perturb the ground-truth input annotations through the use of a random Gaussian noise during the training procedure. Thanks to this operation, the module is forced to refine the input coordinates to the ground truth 3D annotations. The noise is applied on the (x, y) coordinates, before retrieving the z -value and converting them into 3D

camera coordinates, in order to simulate the error of a 2D human pose estimator.

Module C: point cloud-based refinement. The third module of the framework firstly converts the depth map into a point cloud (using the camera calibration parameters K). Then, it refines the body joints exploiting the 3D information of the point cloud by sampling the neighboring points of each joint location. To this aim, we exploit PointNet [171] as deep model since it is specifically designed to analyze point clouds. We extract small point clouds sampling a squared 3D space around each skeleton joint instead of considering the whole point cloud, ranging from the head to the feet of the subject, in order to reduce the computational load and the required GPU memory. In particular, we start from a small 3D volume size and expand it progressively until it contains a minimum number of points or reaches a predefined maximum size, set to 150 mm higher than the initial one. As minimum and maximum number of points, we respectively use 32 and 512, which we empirically select as the best trade-off between accuracy and inference speed. If the number of points in the volume is higher than the maximum, we randomly drop the exceeding points.

Similarly to Module B, Module C predicts independent offsets for each body joint. Each regressed offset is expressed as the displacement vector $\delta = (\delta_x, \delta_y, \delta_z)$ between the input locations of the (x_C, y_C, z_C) joint coordinates in the camera space and the refined ones.

The model architecture derives from the work of Qi et al. [171] and consists of two different blocks: the first is responsible for the feature extraction while the second one for the 3D offset regression. In details, the first block computes single-point features with a series of fully connected layers. Then, single-point features are aggregated through a max-pooling layer. For further details, see [171]. The second block computes the joint offset from the point-cloud features through a fully connected layer with 128 units and ReLU activation and an output layer with 3 units (corresponding to the 3D displacement vector).

As in the previous modules, random Gaussian noise is added to the 3D ground-truth annotations available in the train dataset to create the input data. In this case, since the module works in the 3D camera space, the noise is added to the (x_C, y_C, z_C) coordinates of each joint before the crop of the point cloud.

Experimental evaluation

Dataset. The main drawback of using depth maps for the Human Pose Estimation task is the lack of large-scale and, specifically, manually annotated datasets. Some datasets include annotations on the body surface [43] while other datasets are not always reliable due to the use of marker-based systems, such as the Mocap. Indeed, the visual appearance and 3D shape are altered by the markers and the locations of the body joints usually correspond to the position of these markers, which are placed on the body surface, instead of the physical center of the joint. For this reason, we evaluate the proposed approach on the ITOP dataset [74], that contains the 2D and 3D coordinates of 15 body joints. Using two orthogonal points of view, the human poses are semi-automatically annotated and manually refined to lie inside the body of the subject, *i.e.* at the 3D center of the physical joint. In this work, we focus on the “side-view” part of the dataset, which contains recordings from the common frontal point of view. For further details, please refer to Section 3.2.3.

Experimental setting. RefiNet performs a refinement of 2D body joints on a depth map in order to obtain accurate 3D pose coordinates in the real world camera space. Thanks to the adopted training procedure, the framework is independent from the source of the initial 2D coordinates. As outlined in Section 4.3.2, the independence from the method that predicts the 2D body joints allows the use of pre-trained 2D human pose estimators, such as OpenPose [26] and HRNet [198], on RGB or IR images. The predicted 2D coordinates need to be mapped to the depth image then RefiNet can be applied to improve the 3D prediction. However, the ITOP dataset contains only depth images. Therefore, we train from scratch OpenPose and HRNet on the training set of ITOP using the Adam optimizer [102], learning rate 0.001 and weight decay 0.0001. In the following experiments, we use these two methods as 2D pose estimators. Since our method is independent from the 2D model, we expect to obtain similar results with both the architectures.

In order to assess the quality of the predictions, we adopt two common evaluation metrics: the mean Average Precision (mAP), as proposed by Haque et al. [74], and the mean Distance Error (mDE). The mAP is the percentage of predicted joints whose 3D distance from the ground truth is lower than a threshold τ ; the mDE is the average distance between the

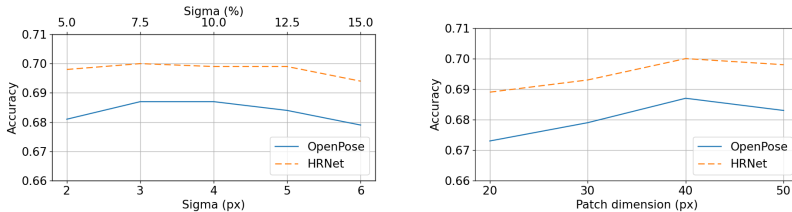


Figure 4.12: Effects of Gaussian noise σ (left) and 2D patch size (right) on mAP of Module A. OpenPose and HRNet refer to the initial set of joints.

predicted joints and the ground truth. They are defined as

$$\text{mAP} = \frac{1}{N} \sum_N (\|\mathbf{v} - \mathbf{w}\|_2 < \tau) \quad [\%] \quad (4.22)$$

$$\text{mDE} = \frac{1}{N} \sum_N \|\mathbf{v} - \mathbf{w}\|_2 \quad [\text{cm}] \quad (4.23)$$

where N is the overall number of joints, \mathbf{v} is the predicted joint while \mathbf{w} is the ground truth joint. In our experiments, we set the threshold $\tau = 10$ cm, as done by competitors [74].

Ablation study. Each module of RefiNet presents a small set of hyper-parameters. In this section, we evaluate their impact to the refinement accuracy.

Module A has two key hyper-parameters: the standard deviation of the Gaussian noise and the 2D patch size. Both the parameters are expressed in pixels. As shown in Figure 4.12 (left), the standard deviation of the random Gaussian noise added to the ground-truth joints has a limited impact, confirming the ability of Module A to correctly refine the original pose. We observe an accuracy peak at $\sigma = 3$ (corresponding to about 7.5% of the patch size), which is the value that is used in the experiments reported in the following sections. On the other hand, the 2D patch size has a higher impact on performance, as shown in Figure 4.12 (right). We use a patch size of 40×40 pixels in the rest of the experiments.

For Module B, we evaluate one main hyper-parameter: the standard deviation of the added Gaussian noise. Also in this case, the parameter is expressed in pixels since the noise is added to the 2D human pose

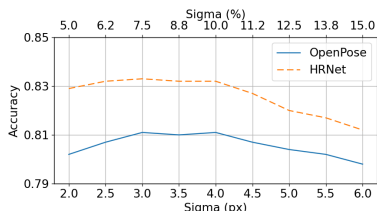


Figure 4.13: Effects of Gaussian noise σ on mAP of Module B. OpenPose and HRNet refer to the initial set of joints.

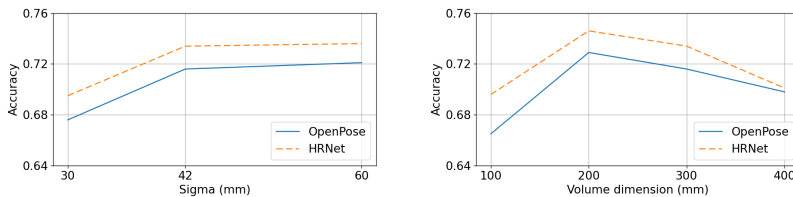


Figure 4.14: Effects of Gaussian noise σ (left) and 3D patch size (right) on mAP of Module C. OpenPose and HRNet refer to the initial set of joints.

(to simulate the error of an inaccurate 2D human pose estimator). As shown in Figure 4.13, the hyper-parameter has a substantial impact on the performance of this module. The higher accuracy is obtained using $\sigma \in [3.0, 4.0]$. Thus, we use $\sigma = 3.0$ in the following experiments.

For Module C, we consider two hyper-parameters: the standard deviation of the added Gaussian noise and the size of the considered 3D volume. Both the parameters are expressed in millimeters. As shown in Figure 4.14 (left), the standard deviation of the random Gaussian noise, which is added to the 3D ground-truth joints, has a limited impact, with an accuracy peak in the range $\sigma \in [42, 60]$. In the following experiments, we set $\sigma = 42$. As in Module A, the initial size of the considered 3D volume has a higher impact on performance, as shown in Figure 4.14 (bottom). In this case, we set a volume size of $200 \times 200 \times 200$ mm in the rest of the experiments.

Results. Experimental results obtained on the ITOP dataset are reported

Table 4.18: Results in terms of mAP and mDE obtained on ITOP dataset. Mod. A, Mod. B and Mod. C refer to the three modules of RefiNet. Improvements are computed w.r.t. poses obtained with the initial 2D pose estimators. The ✓ symbol indicates which RefiNet moduled are enabled and used for the refinement.

Refinement Method	Mod. A	Mod. B	Mod. C	OpenPose [26]				HRNet [198]			
				mAP ↑	Improv.	mDE ↓	Improv.	mAP ↑	Improv.	mDE ↓	Improv.
None				0.646	-	12.634	-	0.670	-	10.711	-
[44]	✓			0.687	6.35%	10.442	17.4%	0.699	4.32%	10.060	6.08%
		✓		0.775	20.0%	8.463	33.0%	0.787	17.5%	8.185	23.6%
			✓	0.719	11.3%	11.834	6.33%	0.734	9.55%	10.693	0.17%
	✓	✓	✓	0.818	26.6%	7.646	39.5%	0.824	23.0%	7.447	30.5%
Ours	✓			0.687	6.35%	10.415	17.6%	0.700	4.48%	9.994	6.69%
		✓		0.811	25.5%	8.258	34.6%	0.833	24.3%	8.335	22.2%
			✓	0.735	13.8%	11.630	7.95%	0.752	12.2%	10.436	2.57%
	✓	✓	✓	0.833	28.9%	7.347	41.8%	0.842	25.7%	7.217	32.6%

in Table 4.18. We compare them with a previous version of our work [44] (which is not presented in this thesis) and with the 2D-to-3D baseline (see Figure 4.10 (top) and Section 4.3.2). Moreover, as an additional ablation study, we report the results obtained by using only one module at a time, indicated with a ✓ symbol, during the testing phase.

Results are expressed in terms of absolute mAP and mDE and in terms of relative improvement. It is worth noting that RefiNet framework leads to better results with an overall improvement of about 27% over mAP and about 37% over mDE compared to the baseline approach. As expected, refining the output of OpenPose and HRNet leads to similar results, confirming that RefiNet is invariant to different off-the-shelf 2D predictors.

Visual results are reported in Figure 4.15. As shown, Module A is able to refine the 2D position of the body joints. However, depth values can be still inaccurate due to local occlusions that influence the sampling of the z value from the depth map, as visible in the example for the left arm. At this point, Module B refines the 3D joints fixing errors caused by occlusions and obtaining a plausible 3D skeleton in terms of, for instance, limb lengths. Finally, Module C refines the 3D prediction of each joint by looking at the 3D points around each skeleton joint.

We also compare the proposed framework with literature methods in Table 4.19. Following the literature convention [74], we present the mAP

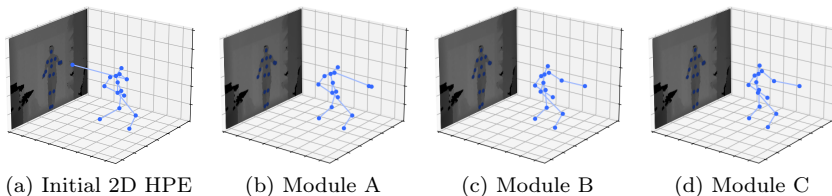


Figure 4.15: Visual examples of the output of each module of the RefiNet framework. From the left, the initial 2D HPE and the depth map are the input of the framework. Then, Module A refines the skeleton through 2D patches, while Module B and Module C work on the 3D skeleton and the point cloud, respectively.

metric divided into the upper and lower body parts in addition to the full body. Specifically, we report the results from the work of Haque et al. [74], our previous version of RefiNet [44], the recent method proposed by Zhang et al. [250] and the 2D-to-3D baseline approach. Experimental results show that RefiNet achieves comparable accuracy with the respect to methods designed to directly work on depth images. Specifically, the proposed framework effectively improves the predictions obtained from off-the-shelf Human Pose Estimation methods originally developed to work on the 2D domain. The method of Zhang et al. [250] confirms that point clouds are an effective information source for this task and that the adoption of an adversarial loss can improve the final accuracy. We observe that both methods are able to predict reasonable 3D human poses in real time.

Finally, we analyze the computational requirements of RefiNet in terms of the number of learnable parameters, the required memory and the inference time (running on CPU or GPU). We evaluate these measures running the framework on a workstation equipped with an Intel i7-7700K and a GPU Nvidia GeForce GTX 1080 Ti and report the results in Table 4.20. As it can be seen, RefiNet is able to run in real time and the three modules introduce a limited overhead in terms of parameters, memory usage and inference time with respect to the off-the-shelf 2D human pose estimation methods.

Table 4.19: Comparison between 3D HPE methods [241, 74, 250], the baseline approach (based on HRNet [198]), our previous version [44] and the proposed method.

Method	ITOP side view		
	Upper Body	Lower Body	Full Body
Baseline	71.2	62.3	67.0
Yub Jung et al. [241]	84.8	72.5	80.5
Haque et al. [74]	84.0	67.3	77.4
Zhang et al. [250]	88.8	94.1	89.6
D'Eusanio et al. [44]	77.9	85.7	81.8
Ours	80.8	88.1	84.2

4.3.3 Discussion

As a first attempt, we investigated Human Pose Estimation using depth data by adapting existing architectures that were designed for 2D HPE on color images. Results reported in Section 4.3.1 show that 2D CNNs can be successfully applied to obtain fairly accurate 2D body joints from depth images. They also show that a small set of data with curated annotations can greatly increase the accuracy of the proposed model.

However, we soon realized that depth maps and accurate 2D body joints are not enough to retrieve a precise 3D human pose in the camera coordinate frame. Leveraging on off-the-shelf 2D human pose estimators, again adapted to work with depth maps, we moved our focus on the refinement of the 3D pose given an initial 2D estimation. Results reported in Section 4.3.2 show that the proposed framework, called RefiNet, is able to regress an accurate 3D pose in camera coordinates given an initial 2D human pose mapped on a depth map. The framework is composed of three independent modules that can be individually enabled or disabled depending on the required precision and the available computational power. While the experimental evaluation proves the effectiveness of the proposed architecture, the performance analysis shows that its overhead is negligible when compared with the requirements of a baseline 2D human pose estimator. Thanks to

Table 4.20: Performance analysis in terms of number of learnable parameters, RAM and inference time required by the system.

Model	Params (M)	RAM (GB)	Infer. CPU (ms)	Infer. GPU (ms)
OpenPose	52.311	1.175	285.377	44.859
HRNet	28.536	1.107	175.757	43.385
Module A	0.828	0.669	6.033	1.872
Module B	4.302	0.665	0.897	0.824
Module C	2.935	1.681	72.607	5.542
RefiNet	8.064	1.705	77.815	7.543

its properties, RefiNet can be employed in settings where there is a need for accurate 3D location of human body joints in absolute camera coordinates, such as in collaborative robotics or for the estimation of anthropometric measurements.

4.4 Estimation of Anthropometric Measurements

The ability to estimate anthropometric measurements – *e.g.* body height, shoulder span, arm length – is a key element in many real world applications and academic research fields, such as soft-biometrics [36], medical health diagnosis [212], person (re)-identification [4], ergonomics [37] and human computer interaction [164].

Usually, accurate anthropometric measurements are collected by qualified personnel (*e.g.* medical staff) relying on time-consuming contact-based measuring methods. Some methods and commercial software that automatically gather anthropometric measurements are available, but they are generally based on high-quality and expensive 3D scanners. In both cases, the measurement accuracy relies on complex acquisition procedures. Recently, the spread of cheap but accurate depth sensors has introduced the possibility to affordably estimate anthropometric measurements using range and visual data. However, a significant issue is represented by the lack of real world and released-for-free datasets containing accurate anthropometric measurements and depth data. To fill this gap, we introduced *Baracca*, a new challenging and multimodal dataset collected for the contact-free estimation of anthropometric measurements from visual data, presented in Section 3.1.4. The dataset consists of more than 9k frames collected by synchronized depth, infrared, thermal and RGB cameras capturing people inside and outside a car.

In this section, we present several approaches for the anthropometric measurement estimation in order to assess the challenges of the proposed dataset and provide useful baselines for future investigations. To the best of our knowledge, this is the first work that presents a thorough evaluation of the efficacy of multiple approaches, *i.e.* a geometric-based approach and multiple machine learning and deep learning methods, in estimating anthropometric measurements from visual cues. Using the proposed *Baracca* dataset, our results show that several anthropometric measurements and soft-biometric traits can be precisely estimated from different contact-free data modalities. We believe that the evaluation of the proposed approaches, in terms of quantitative error metrics and computational requirements, provide a valuable starting point to other academic and industrial researchers in the field.

4.4.1 Proposed Baselines

We present here methods that make use of Baracca for the estimation of anthropometric measurement. In this way, we evaluate the dataset complexity and provide useful baselines for future work. The methods are trained to predict the anthropometric measurements available in the dataset, *i.e.* *height*, *shoulder width*, *forearm length*, *arm length*, *torso width*, *leg length* and *eye height from the ground*. Moreover, we further train a deep model, detailed in the following, to predict the annotated soft-biometric traits, *i.e.* *age*, *weight* and *BMI*. All the experiments are carried out using the official cross-subject train and test splits.

Geometric approach

We propose a geometric method that estimates the distance between the head of the person and the ground and between the eyes and the ground. It works only on the depth data of the outside-view sequences.

The input is a depth map and the camera calibration parameters, that are used to compute the corresponding 3D point cloud. Then, the RANSAC algorithm is used to estimate the plane corresponding to the ground (*i.e.* the plane that fits the elements of the point cloud which belong to the ground). Finally, a trivial point-to-plane distance is calculated to retrieve the height and the eye height of the subject. This method does not require any training, but requires the location of the upper part of the head and of the eyes. To obtain them, we used the joints provided by HRNet [198]. We report results obtained using the entire point cloud (“Geom. (100%)”) and using only 1% of the points (“Geom. (1%)”).

Machine Learning approaches

We propose machine learning methods that do not directly exploit the images of the dataset, but use the body skeleton, *i.e.* a set of human joints, calculated in every frame with HRNet [198]. After the skeleton estimation, the following set of distances is calculated over it and used as input of the learning models: head-neck, neck-shoulder, shoulder-elbow, elbow-hand, neck-hip, hip-knee. Only the first 3 distances are used for the in-car view since the lower body, elbows and hands are often not visible. When possible, these measures are calculated as the mean of the left and the right side of the body.

We evaluate three well-known machine learning methods: Linear Regression, Random Forests and AdaBoost.

- The Linear Regression method simply attempts to fit a linear model to the training data as the least-squares solution.
- Random decision trees and forests [23] consist of an ensemble of regression decision trees, which are independently trained as in the bagging technique. When testing, the estimation of each tree is averaged to obtain the final result.
- Adaptive Boosting [59, 48] (AdaBoost) consists of multiple weak regressors, sequentially trained weighting the training samples based on the errors of the previous weak regressors. The single predictions are combined with a weighted sum to obtain the final estimation.

Deep Learning approach

We propose a deep learning-based method that directly estimates the anthropometric measures from visual data. The deep model is composed of ResNet-18 [77], without the last fully connected layer, pre-trained on ImageNet [40] and used as feature extractor. It is followed by a fully connected layer with 128 units, batch normalization, ReLU activation and dropout (drop probability $p = 0.5$). Finally, a linear layer with size k regresses the k anthropometric measures. The network is trained optimizing the robust Huber loss function [86] through the Adam optimizer [102]. The training is executed for 70 epochs with a batch size of 32 and a learning rate of 0.001, which is reduced by a factor of 10 after 50 and 65 epochs. The input image size is 128×128 .

4.4.2 Experimental Evaluation

The results presented in this section are obtained training and testing the proposed baselines on the official training and testing set of the Baracca dataset. We further split the dataset in the “Outside view” split, which contains external sequences (at 1, 1.5 and 2 meters), and in the “In-car view”, which contains the in-car sequences.

Baseline results, obtained predicting anthropometric measures from depth, IR, RGB and thermal data, are respectively reported in Tables 4.21,

Table 4.21: Results on the depth domain, in terms of MAE \pm std (cm). The ML approaches employ 3D joints in this setting.

Outside view								
Method	Height	Eye Height	Forearm	Arm	Torso	Leg	Shoulders	Average
Geom. (100%)	5.576 \pm 4.6	4.393 \pm 5.0	-	-	-	-	-	4.985 \pm 4.8
Geom. (1%)	5.686 \pm 4.9	4.570 \pm 5.2	-	-	-	-	-	5.128 \pm 5.0
LR	3.853 \pm 1.9	1.115 \pm 0.3	1.740 \pm 0.4	2.151 \pm 0.3	4.538 \pm 0.5	2.597 \pm 1.3	2.317 \pm 1.0	2.616 \pm 0.8
RandomForest	3.238 \pm 2.9	1.187 \pm 0.9	1.745 \pm 1.1	2.593 \pm 1.2	3.912 \pm 1.3	3.049 \pm 2.3	2.235 \pm 1.2	2.566 \pm 1.6
Adaboost	3.523 \pm 2.3	0.814 \pm 0.5	1.382 \pm 0.6	2.548 \pm 1.1	3.993 \pm 1.1	2.310 \pm 2.0	2.393 \pm 1.1	2.423 \pm 1.2
Deep Model	5.724 \pm 3.1	5.201 \pm 3.0	0.840 \pm 0.5	2.014 \pm 0.6	2.482 \pm 0.7	3.613 \pm 1.9	3.040 \pm 0.9	3.273 \pm 1.5

In-car view								
Method	Height	Eye Height	Forearm	Arm	Torso	Leg	Shoulders	Average
LR	3.667 \pm 1.5	1.031 \pm 0.2	1.937 \pm 0.3	2.604 \pm 0.3	4.408 \pm 0.3	3.474 \pm 1.4	2.774 \pm 0.8	2.842 \pm 0.7
RandomForest	4.185 \pm 3.0	1.035 \pm 0.8	1.890 \pm 1.0	2.686 \pm 1.6	4.412 \pm 2.1	3.211 \pm 2.4	2.494 \pm 1.1	2.845 \pm 1.7
Adaboost	3.973 \pm 1.9	0.939 \pm 0.3	1.500 \pm 0.3	2.283 \pm 0.9	4.627 \pm 0.8	3.210 \pm 1.4	2.441 \pm 0.7	2.710 \pm 0.9
Deep Model	7.082 \pm 5.9	6.316 \pm 5.5	1.016 \pm 0.9	2.072 \pm 0.9	2.874 \pm 1.3	4.734 \pm 3.4	3.370 \pm 1.4	3.923 \pm 2.8

Table 4.22: Results on the IR domain, in terms of MAE \pm std (cm). The ML approaches employ 3D joints in the “known distance” setting, 2D joints otherwise.

Outside view - known distance								
Method	Height	Eye Height	Forearm	Arm	Torso	Leg	Shoulders	Average
LR	3.524 \pm 2.6	1.020 \pm 0.5	1.415 \pm 0.6	2.096 \pm 0.4	4.108 \pm 0.6	2.156 \pm 2.2	1.946 \pm 1.1	2.324 \pm 1.1
RandomForest	3.530 \pm 3.0	1.014 \pm 0.8	1.973 \pm 1.0	2.389 \pm 1.1	4.345 \pm 1.4	2.649 \pm 2.2	2.212 \pm 1.2	2.587 \pm 1.5
Adaboost	3.998 \pm 2.5	0.894 \pm 0.4	1.440 \pm 0.4	2.317 \pm 1.0	4.271 \pm 0.8	2.144 \pm 1.9	2.147 \pm 1.0	2.459 \pm 1.1

Outside view								
Method	Height	Eye Height	Forearm	Arm	Torso	Leg	Shoulders	Average
LR	5.823 \pm 2.1	1.150 \pm 0.4	1.629 \pm 0.4	2.228 \pm 0.3	4.409 \pm 0.3	3.412 \pm 1.9	2.589 \pm 1.4	3.034 \pm 1.0
RandomForest	3.916 \pm 3.1	1.097 \pm 1.1	1.856 \pm 0.9	2.737 \pm 1.4	4.653 \pm 1.3	2.973 \pm 2.7	2.250 \pm 1.3	2.783 \pm 1.7
Adaboost	4.542 \pm 1.8	1.011 \pm 0.3	1.253 \pm 0.4	2.328 \pm 0.6	4.480 \pm 0.9	3.117 \pm 1.6	2.253 \pm 1.0	2.712 \pm 0.9
Deep Model	6.641 \pm 1.9	5.914 \pm 1.8	1.109 \pm 0.3	1.965 \pm 0.3	2.579 \pm 0.4	4.113 \pm 1.1	3.011 \pm 0.4	3.619 \pm 0.9

In-car view								
Method	Height	Eye Height	Forearm	Arm	Torso	Leg	Shoulders	Average
LR	4.975 \pm 1.0	0.964 \pm 0.3	1.837 \pm 0.4	2.527 \pm 0.5	4.412 \pm 0.6	3.509 \pm 1.1	2.885 \pm 0.8	3.016 \pm 0.6
RandomForest	6.515 \pm 3.5	1.151 \pm 1.1	2.056 \pm 1.3	2.493 \pm 1.7	4.396 \pm 1.8	3.822 \pm 2.5	2.809 \pm 1.4	3.320 \pm 1.9
Adaboost	4.924 \pm 1.3	1.018 \pm 0.2	1.395 \pm 0.4	2.408 \pm 0.5	4.395 \pm 0.6	4.206 \pm 2.0	3.015 \pm 0.7	3.052 \pm 0.8
Deep Model	6.555 \pm 3.6	6.488 \pm 3.4	1.130 \pm 0.6	2.034 \pm 0.7	1.895 \pm 0.9	3.952 \pm 2.2	3.022 \pm 1.0	3.582 \pm 1.8

4.22, 4.23 and 4.24. We report the Mean Absolute Error (MAE) and the standard deviation calculated between the predicted value and the ground

Table 4.23: Results on the RGB domain, in terms of MAE \pm std (cm). The ML approaches employ 3D joints in the “known distance” setting, 2D joints otherwise.

Outside view - known distance								
Method	Height	Eye Height	Forearm	Arm	Torso	Leg	Shoulders	Average
LR	3.633 \pm 2.0	1.089 \pm 0.5	1.647 \pm 0.3	1.981 \pm 0.4	4.624 \pm 0.6	2.002 \pm 1.5	1.587 \pm 1.1	2.366 \pm 0.9
RandomForest	3.844 \pm 2.2	1.147 \pm 0.9	1.888 \pm 0.8	1.982 \pm 1.0	4.740 \pm 1.2	2.734 \pm 1.9	1.882 \pm 1.1	2.602 \pm 1.3
Adaboost	2.877 \pm 1.8	0.955 \pm 0.5	1.455 \pm 0.4	1.995 \pm 0.7	4.610 \pm 0.5	2.442 \pm 1.8	1.931 \pm 0.7	2.324 \pm 0.9

Outside view								
Method	Height	Eye Height	Forearm	Arm	Torso	Leg	Shoulders	Average
LR	5.260 \pm 1.8	1.007 \pm 0.4	1.866 \pm 0.4	2.201 \pm 0.5	4.859 \pm 0.5	3.409 \pm 1.4	2.510 \pm 1.5	3.016 \pm 0.9
RandomForest	4.321 \pm 2.8	1.082 \pm 1.0	1.784 \pm 0.9	1.946 \pm 1.1	4.801 \pm 1.2	2.891 \pm 2.6	2.189 \pm 1.2	2.716 \pm 1.5
Adaboost	4.771 \pm 1.4	1.004 \pm 0.2	1.280 \pm 0.3	2.192 \pm 0.7	4.716 \pm 0.6	3.033 \pm 1.2	2.235 \pm 0.8	2.747 \pm 0.7
Deep Model	10.124 \pm 3.4	9.373 \pm 3.2	1.564 \pm 0.5	1.898 \pm 0.5	2.355 \pm 0.7	6.392 \pm 2.0	3.348 \pm 0.9	5.008 \pm 1.6

In-car view								
Method	Height	Eye Height	Forearm	Arm	Torso	Leg	Shoulders	Average
LR	5.224 \pm 1.3	0.955 \pm 0.3	1.775 \pm 0.3	2.477 \pm 0.4	4.365 \pm 0.5	3.770 \pm 1.1	2.811 \pm 0.8	3.054 \pm 0.7
RandomForest	6.481 \pm 3.7	1.307 \pm 1.0	2.240 \pm 1.2	2.724 \pm 1.7	4.278 \pm 1.3	4.937 \pm 3.1	3.466 \pm 1.6	3.633 \pm 1.9
Adaboost	6.014 \pm 1.5	0.912 \pm 0.3	1.541 \pm 0.4	2.279 \pm 0.6	4.279 \pm 0.3	4.378 \pm 1.3	3.093 \pm 1.2	3.214 \pm 0.8
Deep Model	7.898 \pm 3.5	7.810 \pm 3.3	1.520 \pm 0.6	2.115 \pm 0.6	2.314 \pm 0.7	4.973 \pm 1.9	2.776 \pm 0.8	4.201 \pm 1.6

Table 4.24: Results on the thermal domain, in terms of MAE \pm std (cm). The ML approaches employ 2D joints.

Outside view								
Method	Height	Eye Height	Forearm	Arm	Torso	Leg	Shoulders	Average
LR	4.877 \pm 1.4	1.182 \pm 0.4	1.496 \pm 0.5	2.316 \pm 0.5	4.117 \pm 0.7	3.347 \pm 1.2	2.607 \pm 0.9	2.849 \pm 0.8
RandomForest	5.278 \pm 3.5	1.351 \pm 1.0	1.568 \pm 0.8	2.268 \pm 1.3	3.861 \pm 1.1	3.767 \pm 2.8	2.294 \pm 1.3	2.912 \pm 1.7
Adaboost	5.064 \pm 1.9	1.131 \pm 0.4	1.349 \pm 0.4	2.250 \pm 0.7	4.291 \pm 0.5	3.203 \pm 1.8	2.442 \pm 1.0	2.819 \pm 1.0
Deep Model	5.267 \pm 3.1	4.939 \pm 2.9	0.955 \pm 0.4	2.220 \pm 0.5	2.458 \pm 0.7	3.659 \pm 1.8	2.930 \pm 0.8	3.204 \pm 1.4

In-car view								
Method	Height	Eye Height	Forearm	Arm	Torso	Leg	Shoulders	Average
LR	4.823 \pm 1.2	1.087 \pm 0.2	1.611 \pm 0.1	2.251 \pm 0.3	4.409 \pm 0.5	3.112 \pm 1.0	2.443 \pm 0.5	2.819 \pm 0.6
RandomForest	5.038 \pm 3.7	1.402 \pm 1.0	1.826 \pm 0.9	2.233 \pm 1.2	4.678 \pm 1.3	3.365 \pm 2.9	3.035 \pm 1.6	3.082 \pm 1.8
Adaboost	4.856 \pm 2.0	1.172 \pm 0.2	1.792 \pm 0.5	2.295 \pm 0.6	4.587 \pm 0.5	3.507 \pm 1.3	2.805 \pm 1.1	3.002 \pm 0.9
Deep Model	6.632 \pm 2.7	6.320 \pm 2.7	0.945 \pm 0.4	2.317 \pm 0.4	2.441 \pm 0.7	4.542 \pm 1.5	3.479 \pm 0.7	3.811 \pm 1.3

truth in centimeters, aggregated for each subject and then on the whole test set. Considering the depth domain, we report the geometrical approach (“Geom.”), which exploits the point clouds; the machine learning approaches, which employ the 3D distances between joints (in camera space); and the

Table 4.25: Age, weight and BMI estimated by the Deep Learning Approach using different data domains, in terms of MAE \pm std.

Outside view			
Domain	Age	Weight	BMI
Depth	3.863 \pm 0.8	10.749 \pm 5.0	3.247 \pm 1.3
IR	3.824 \pm 0.6	5.689 \pm 3.4	2.278 \pm 0.9
RGB	3.530 \pm 0.6	16.537 \pm 4.8	4.098 \pm 1.3
Thermal	4.040 \pm 0.8	9.926 \pm 3.8	2.386 \pm 1.0
In-Car view			
Domain	Age	Weight	BMI
Depth	3.819 \pm 0.8	9.561 \pm 6.4	2.603 \pm 1.5
IR	4.914 \pm 1.6	7.410 \pm 3.2	2.235 \pm 0.8
RGB	4.135 \pm 1.0	11.959 \pm 5.4	2.992 \pm 1.4
Thermal	3.550 \pm 0.9	11.012 \pm 5.7	2.620 \pm 1.5

Table 4.26: Inference time of the tested approaches, in ms \pm std.

Method	Inference time (ms)	
	CPU	GPU
Geom. (100%)	741.9 \pm 138.3	-
Geom. (1%)	66.81 \pm 1.992	-
HRNet	591.8 \pm 134.2	61.81 \pm 24.44
+ LR	0.047 \pm 0.006	-
+ RandomForest	0.540 \pm 0.013	-
+ Adaboost	1.527 \pm 0.693	-
Deep Model	23.07 \pm 0.430	4.619 \pm 0.289

deep learning method, which analyzes the depth images. The 3D joints are obtained from the 2D image coordinates using the depth values and the camera calibration parameters. In the other cases (IR, RGB and thermal), we report the machine learning approaches, which exploit the 2D distances between joints (in image coordinates), and the deep method, which employs normalized images. Moreover, in the IR and RGB case, we further report results obtained using the 3D distances between joints in the “Outside view”.

We exploit the known distance (1, 1.5, 2 meters) as depth approximation and the camera calibration parameters to convert the 2D joints (in image coordinates) to the 3D ones (in camera space). In addition, Table 4.25 contains the results obtained by the deep model trained for the estimation of soft-biometric traits. Finally, Table 4.26 presents the inference time of the proposed approaches.

4.4.3 Discussion

In this section, we demonstrated that several anthropometric measurements can be successfully estimated using any of the visual data included in the Baracca dataset, *i.e.* depth, infrared, RGB and thermal data. As can be seen from Tables 4.21, 4.22, 4.23, 4.24, the machine learning approaches, which exploit the accurate joint predictions of HRNet [198], are the methods that ensures the lowest average error.

Considering the IR and RGB domain (Table 4.22 and Table 4.23), the use of approximate 3D joints further improves the accuracy of these methods, confirming that 3D data, independent from the camera intrinsics, are the most suitable data for the anthropometric estimation. However, this kind of sensors require additional constraints and assumptions. Indeed, the 2D to 3D conversion is possible only if the distance between the subject and the camera is known and if all the subject's joints can be assumed to lie on a plane at the same distance from the camera.

For that reason, the most adequate sensor for anthropometric estimation is the depth one, which naturally gather the 3D information of the scene. Using range data, even a simple geometrical approach can be employed, obtaining acceptable, but less accurate results (see Table 4.21). It is worth to note that this approach still obtains low MAE even with extremely small point clouds (1% of the original one, consisting in just 1k-2k points). This result shows that cheap low-resolution depth sensors can be successfully used to estimate anthropometric measurements in real-world settings.

Regarding the inference time, the ML approaches are extremely fast, but require the subject body joints (calculated, for instance, with HRNet [198]) increasing the overall inference time, as can be seen in Table 4.26. Being end-to-end, the deep method is the fastest approach, regardless of running on CPU or GPU. Moreover, this method can estimate additional soft-biometric traits with a relatively low average error from any data type, as shown in Table 4.25.

4.5 3D Reconstruction of Vehicles

In recent years, the inference of 3D object shapes from 2D images has shown astonishing progress in the computer vision community. By addressing the task as an inverse graphics problem, *i.e.* considering the 2D image as the rendering of a 3D model, several methods [92, 66, 210] have shown that deep models are capable of restoring the shape, pose and texture of the portrayed object. While previous methods rely on direct 3D supervision [34, 63, 222, 236] or multiple views [203, 71, 208, 122], recent approaches only require segmentation masks, object keypoints and coarse camera poses [92, 66, 210]. In the last couple of years, some methods have lessened the dependency on keypoints [210] and even on the camera viewpoint [66]. All these methods share the same underlying approach: a deep model learns a mean 3D shape, called *meanshape*, for the object category during training; then, instance-specific deformation, texture and camera pose are predicted and applied to the learned meanshape to regress the 3D model of the object.

A major limitation of existing methods is that they are category-specific: they must be trained and evaluated on image collections of a single object category. This choice has been motivated by the need of category-specific priors in order to recover the 3D shape from 2D images, which is indeed an ill-posed problem unless additional constraints are taken into account. Moreover, most of the approaches [92, 66, 210] initialize the learnable meanshape with a category-specific representative 3D model. To the best of our knowledge, there have been no attempts to extend these methods to scenarios where image collections of multiple categories are available both in training and at inference time.

In this section, we present a multi-category approach that learns to infer the 3D mesh of an object from a single RGB image. As illustrated in Figure 4.16, the method learns a series of deformable 3D models and predicts a set of instance-specific deformation, pose and texture based on the input image. Differently from previous approaches, the proposed framework is trained with images of multiple object categories using only foreground masks and rough camera poses as supervision. While rough camera poses could depend on the object category, this is not strictly needed for classes that share semantic keypoints. The method learns several 3D models in an unsupervised manner, *i.e.* without explicit category supervision, starting from a set of spheres and automatically selects the proper one during inference. Moreover, the instance-specific deformation is inferred by a

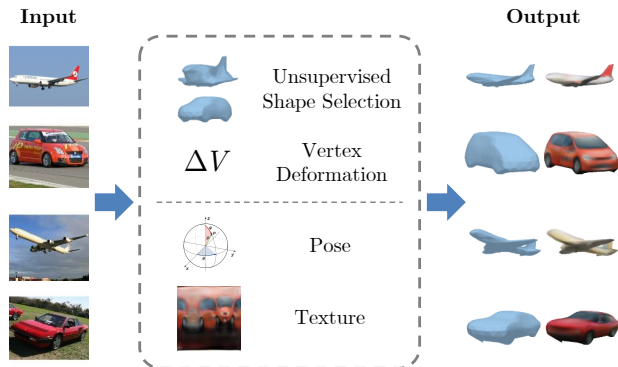


Figure 4.16: Overview of the proposed approach. The method predicts realistic 3D textured shapes of objects of different categories and their 3D pose from a single RGB image.

network that independently predicts the displacement of each vertex of the learned 3D mesh, given the 3D position of the vertex and conditioned on the selected shape and the visual features extracted from the input image. The predicted deformation is naturally smooth and the number of vertices and triangles of the 3D mesh can be dynamically changed during training, with either a global or a local subdivision.

To showcase the quality of the proposed method, we present a variety of experiments in different settings on two datasets, namely Pascal3D+ [233] and CUB [218], and run several ablation studies. For instance, we test the method on multiple object categories related to the automotive environment of the Pascal3D+ dataset (*i.e.* bicycle, bus, car and motorbike) and on the entire set of Pascal3D+ categories. Qualitative and quantitative results confirm the quality of the proposed approach and show that the model is able to learn category-specific shape priors without direct supervision.

To sum up, our main contributions are as follows:

- We present an approach that recovers the 3D shape, pose and texture of an object from a 2D image. The method is trained using image collections with foreground masks and coarse camera poses, but no explicit category nor 3D supervision.
- Our multi-category framework learns to distinguish between different

object categories and produces meaningful meanshapes starting from a set of 3D spheres.

- Our approach predicts single vertex deformations, resulting in smooth 3D surfaces and enabling the dynamic subdivision of the learned meshes.
- We publicly release our code and trained models ³.

4.5.1 Proposed Method

In this section, we present the components of our method, from the input image to the reconstructed 3D textured mesh. The architecture is illustrated in Figure 4.17.

Preliminary definitions

Shape. As other approaches in the literature [92, 95, 66, 121], we use the triangle mesh as 3D shape representation, which is defined by a set of vertices $V = \{v_j = [x, y, z], j = [1, \dots, k]\}$ and a set of triangle faces F . The faces determine the connectivity between vertices, but are also related to the texture mapping. In our approach, we leverage this connectivity property and dynamically change, during training, the number of vertices and faces of the 3D shape aiming for smoothness and better textures. We refer to this technique as *dynamic mesh subdivision*.

Texture. The triangle mesh texture is represented by a texture image I_{tex} and a color map UV which maps between the 2D coordinate space of I_{tex} and the 3D coordinate space of the mesh surface of a sphere. Thus, the UV mapping is defined by spherical coordinates.

Pose. We use a weak-perspective camera projection to define the 3D object pose, as commonly done in literature. This geometric projection is a simplified version of the standard perspective projection. Thus, the object pose is parametrized by a scale factor $s \in \mathbb{R}$, a translation $t = (x, y)$ in image coordinates and a quaternion rotation q obtained by a rotation matrix computed from Euler angles (*i.e.* azimuth, elevation and roll). We define $\pi = (s, t, q)$ as the weak-perspective camera projection.

³ <https://github.com/aimagelab/mcmr>

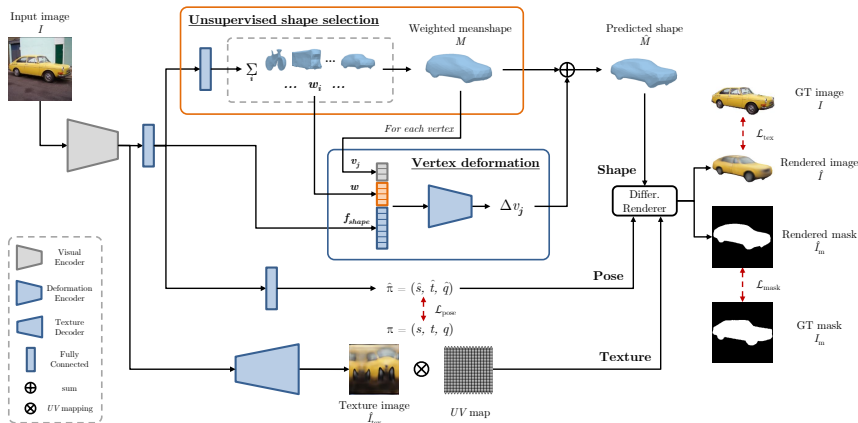


Figure 4.17: Detailed overview of the proposed method. The *unsupervised shape selection* module predicts the category meanshape while the *vertex deformation* module infers the instance-specific deformation, obtaining the predicted shape. In parallel, pose and texture are estimated and then provided, along with the shape, to a differentiable renderer that renders the textured image.

Rendering. In order to render a 3D shape with its texture, we rely on the differentiable renderer Soft Rasterizer [125]. It takes a triangle mesh, a texture image I_{tex} and an object pose π as input and outputs the rendering of the textured object as the RGB image \hat{I} and the foreground mask \hat{I}_{m} .

Multi-category mesh reconstruction

Our goal is the recovery of the 3D shape of an object from a single image. In the literature, this task has been often addressed by splitting it in two parts: on the one hand, the definition or learning of a category-specific base shape, named *meanshape*; on the other hand, the prediction of an instance-specific deformation of the learned shape. Differently from the majority of previous works (see Table 2.3), we do not need a category-specific initialization of these shapes and propose the joint and unsupervised training of shapes for multiple object categories. In the following, we provide the details of our approach.

Feature extraction. Given an RGB image $I \in \mathbb{R}^{3 \times w \times h}$ as input, the first step of our framework is the extraction of visual features with a convolutional encoder (*e.g.* ResNet-18 [77] in our experiments). These features are defined as f_{tex} and used to estimate the 3D object texture with a specific decoder. The same features are flattened and mapped into a compact version f_{shape} , used to recover the shape and its viewpoint.

Unsupervised shape selection. In contrast to current literature approaches, which are category specific, we propose an unsupervised technique that automatically learns to distinguish between different object categories. Instead of a single meanshape, we define a set of N deformable spheres and use a network to select the instance-specific meanshape according to the input image. The features f_{shape} are passed through a set of fully connected layers and a softmax function. Then, the resulting scores are used to compute a weighted sum of the mesh vertices and obtain a single mesh, approximating the argmax function over the N meanshapes. While the meanshapes are initially defined as spheres, they are updated during the training process and progressively specialize in different object categories. Formally, let $M_i = (V_i, F)$ be one of the N meanshapes, composed of a set of vertices V_i and faces F , and $\mathbf{w} = [w_1, \dots, w_N]$ be the output of the network. The weighted meanshape M is computed as:

$$M = (V, F) = \left(\sum_{i=1}^N w_i V_i, F \right) \quad (4.24)$$

This mesh M will be deformed according to the object depicted in the input image I , as explained in the following.

Vertex deformation. Inspired by previous works [67, 159], we develop a lightweight network which deforms the meanshape M taking as input the features f_{shape} and the 3D coordinates of a single meanshape vertex v_j at a time. We further condition the output on the selected meanshape giving the weighting scores \mathbf{w} produced by the previous module as additional input. In this way, we enforce the connection between the weighted meanshape M and the predicted deformation. The module outputs a 3D displacement or deformation Δv_j of the vertex v_j in the 3D space. This approach makes the architecture independent of the number of vertices of the mesh, enabling us to predict the deformation of meshes of variable sizes. Given a set of deformations ΔV for each vertex of a meanshape M , the predicted shape can be defined as $\hat{M} = M + \Delta V = (V + \Delta V, F)$.

Dynamic mesh subdivision. In order to improve the smoothness of the predicted deformed shape, we apply during training a dynamic subdivision of the triangle mesh. In particular, we use a global subdivision that divides each triangle of a mesh M in 4 equal parts. Other methods that make use of mesh subdivision (*e.g.* [222, 119]) need architectural changes that drastically increase the required memory and the inference time. On the contrary, our method is not heavily affected by the mesh subdivision operation and does not require any architectural changes, thanks to the per-vertex prediction of the deformation network.

3D pose regression. We further predict the object viewpoint with a supervised regression technique using two fully connected layers which take as input the features f_{shape} and output a 3D weak-perspective pose $\hat{\pi} = (\hat{s}, \hat{t}, \hat{q})$.

Texture prediction. In order to produce a realistic 3D shape, we finally predict the texture that the differentiable renderer applies to the predicted deformed mesh \hat{M} . Similarly to the work of Goel et al. [66], we use a convolutional decoder that takes as input the visual features f_{tex} , which preserve the spatiality, and directly outputs an RGB image \hat{I}_{tex} . The texture is mapped onto the UV space of the shape, which is homeomorphic to a sphere, so that it can be exploited by the renderer to produce the final image \hat{I} .

Losses and priors

The shape prediction is supervised only by two annotated information that are the binary object mask I_m and the 3D camera pose π .

We first handle the shape deformation applying a mask loss $\mathcal{L}_{\text{mask}} = \|I_m - \hat{I}_m\|_2^2$ where \hat{I}_m is the binary object mask produced by the renderer using the ground truth pose π . In addition to this loss, we also use some priors in order to maintain a certain smoothness of the object surface. The first prior is a laplacian smoothing loss $\mathcal{L}_{\text{smooth}} = \|LV\|_2$ where the Laplace-Beltrami operator [196] minimizes the mean curvature; we apply this smoothing prior both to the predicted deformations ΔV and the vertices of the deformed shape \hat{M} . The second prior is a regularization term $\mathcal{L}_{\text{def}} = \|\Delta V\|_2$ which prevents the network from learning large deformations and helps to produce more realistic meanshapes. Our final shape loss is represented by:

$$\mathcal{L}_{\text{shape}} = \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{def}} \quad (4.25)$$

For the pose regression module, we use a loss defined as:

$$\mathcal{L}_{\text{pose}} = \|\hat{s} - s\|_2^2 + \|\hat{t} - t\|_2^2 + (1 - |q * (\hat{q} \odot -\hat{q})|) \quad (4.26)$$

where the first two terms consist of the mean squared error for scale and translation and the last term is the geodesic quaternion loss. The operator $*$ is the Hamilton product and \odot the concatenation between the original quaternion and its version rotated by 360 degrees, representing the same rotation. Moreover, following the approach proposed by Pavllo et al. [163], we further regularize the quaternion prediction with the penalty term $\mathcal{L}_{\text{pose_reg}} = w^2 + x^2 + y^2 + z^2 - 1^2$ that forces the quaternion to have unit length and thus representing a valid rotation. The overall camera loss is set as:

$$\mathcal{L}_{\text{cam}} = \mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{pose_reg}} \quad (4.27)$$

In order to produce realistic colors and details for the object texture, we convert the rendered RGB image and the masked input image to the LAB color space and apply the following losses: a color loss $\mathcal{L}_{\text{color}} = \|\hat{I}_{ab} - (I \cdot I_m)_{ab}\|_2^2$ on the AB channels for more faithful texture details and a style loss $\mathcal{L}_{\text{style}} = \|\hat{I}_L - (I \cdot I_m)_L\|_2^2$ on the L channel for sharper high-frequency details. Moreover, we apply a perceptual loss $\mathcal{L}_{\text{percept}} = F_{\text{dist}}(\hat{I}, I \cdot I_m)$ where F_{dist} is the metric defined by Zhang et al. [247] using a VGG16 backbone as feature extractor. The final texture loss is defined by:

$$\mathcal{L}_{\text{tex}} = \mathcal{L}_{\text{color}} + \mathcal{L}_{\text{style}} + \mathcal{L}_{\text{percept}} \quad (4.28)$$

The overall objective applied during training is a weighted sum of the shape, camera and texture losses, obtaining a balanced learning of the different network modules. For more details about the loss weights, please refer to the supplementary material of our paper [192].

4.5.2 Experimental Evaluation

In this section, we firstly present the employed datasets and the experimental setting. Then, we present quantitative and qualitative evaluations of our approach in comparison with literature methods. Finally, we report an ablation study on the key elements of the proposed approach.

Datasets

Two common datasets, namely Pascal3D+ [233] and CUB-200-2011 [218], are used to evaluate the proposed approach on a diverse set of object categories and, at the same time, to obtain a comparison with the current state-of-the-art methods. As done in previous works [92, 66], 2D image collections, foreground masks and coarse camera/object poses (manually or automatically annotated) are used for training. We do not take advantage of annotated keypoint positions nor coarse 3D model correspondences. For further information about Pascal3D+ and CUB-200-2011, please refer to Section 3.2.

Network architecture

Our model is composed of 5 modules: (i) a visual encoder, defined as a pre-trained ResNet-18, with an additional convolutional layer, (ii) an unsupervised shape selection module composed of two fully connected layers and a softmax activation function, (iii) a vertex deformation network with four 512-dimensional fully connected layers with random dropout and a tanh activation function, (iv) a camera pose regressor with two fully connected layers and random dropout and (v) a texture decoder that follows the implementation of the SPADE architecture [160] with 6 upsampling steps. Additional details are available in the supplementary material of our paper [192].

Training procedure

We train our network on both datasets for 500 epochs with an initial learning rate of $1e^{-4}$. The meanshapes are initialized as icospheres with 162 vertices and 320 faces (corresponding to the subdivision level 3). After 350 epochs, we apply the dynamic subdivision to the 3D shapes (roughly obtaining the subdivision level 4) and reduce the learning rate to $1e^{-5}$. Our final 3D shape has roughly the same number of vertices and faces as the competitor approaches [92, 66] which use a deformable template with subdivision level fixed to 4.

All input images are cropped using the object bounding box and resized to a dimension of 256×256 and the model predicts a texture image of the same size. As data augmentation, we apply standard random jittering on the bounding box size and location and random horizontal image flipping. In

Table 4.27: 3D IoU on Pascal3D+. Our method is trained on aeroplanes and cars independently using N meanshapes (one for each subclass) or on aeroplanes and cars jointly with 2 meanshapes.

Approach	Training	Aeroplane	Car	Avg
CSDM [94]	indep.	0.400	0.600	0.500
DRC [208]	indep.	0.420	0.670	0.545
CMR [92]	indep.	0.460	0.640	0.550
IMR [210]	indep.	0.440	0.660	0.550
U-CMR [66]	indep.	-	0.646	-
Ours (N meanshapes)	indep.	0.460	0.684	0.572
Ours (2 meanshapes)	joint	0.448	0.686	0.567

addition, instead of forcing the shape to be symmetric with post-processing steps (as done in other works, *e.g.* [92, 66, 121]), we force the network to predict symmetric shapes with the following approach, similar to what is done in the work of Wu et al. [231]. During training, the predicted shape (*i.e.* its pose) is randomly rotated by 180 degrees around the vertical axis and compared with the flipped versions of the ground truth image and mask. In this way, the network is forced to predict symmetric shapes (along the vertical axis) and thus to consistently minimize the losses without computational overhead.

We use a batch size of 16 and Adam [102] as optimizer with a momentum of 0.9. The code is developed using PyTorch [162].

Results on Pascal3D+

We show the results of our method compared to the state of the art on the Pascal3D+ dataset in Table 4.27, using the 3D IoU metric as proposed by Tulsiani et al. [208]. We present two different versions of our method. Firstly, we employ the same approach used by competitors: train a different model for each class of Pascal3D+ (experiments marked as “independent training”). In this case, we set the number of meanshapes equal to the number of subclasses of Pascal3D+, *i.e.* $N = 8$ for the aeroplane class, $N = 10$ for the car class. As reported in the second-to-last row of Table 4.27, our method can leverage the use of multiple meanshapes and the dynamic

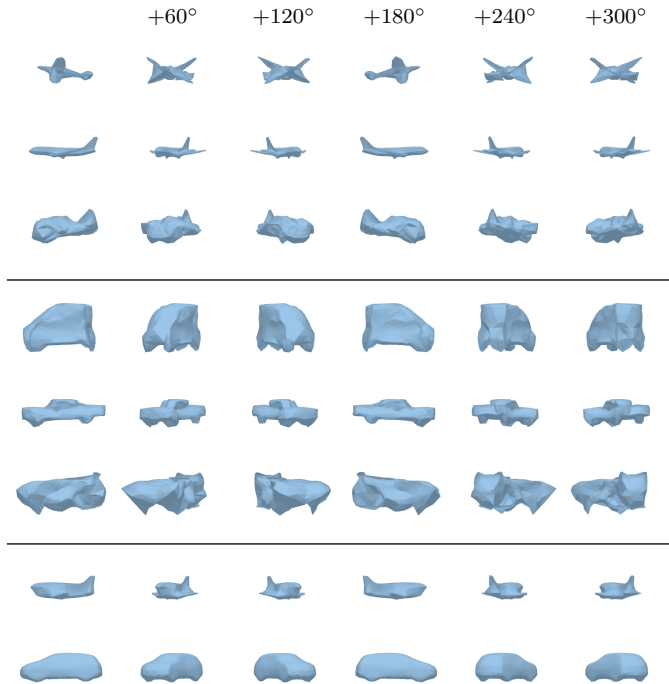


Figure 4.18: Some of the meanshapes learned training on Pascal3D+. First group: aeroplane class (8 meanshapes); second group: car class (10 meanshapes); third group: aeroplane and car classes (2 meanshapes).

subdivision obtaining state-of-the-art results on this dataset. In addition, we jointly train our method on both the aeroplane and the car classes, using 2 meanshapes and letting the network distinguish between the two classes. Even in this more complex scenario, we obtain comparable or state-of-the-art scores on both classes (see last row of Table 4.27). The learned meanshapes for these three experiments, *i.e.* training on aeroplanes, on cars and on aeroplanes and cars jointly, are shown in Figure 4.18. We observe that the set of meanshapes on the single classes contains both recognizable and less explainable shapes (Figure 4.18, top and middle). On the other hand, the two meanshapes learned in an unsupervised manner using images

Table 4.28: Mask IoU and texture metrics on CUB. Our method is trained using 1 or 14 learnable meanshapes.

Approach	Mask IoU \uparrow		Texture metrics		
	Pred cam	GT cam	SSIM \uparrow	L1 \downarrow	FID \downarrow
CMR [92]	0.706	0.734	0.718	0.063	290.32
DIB-R [32]	-	0.757	-	-	-
U-CMR [66]	0.637	-	0.689	0.077	190.35
Ours (1 meanshape)	0.658	0.721	0.717	0.064	227.24
Ours (14 meanshapes)	0.642	0.723	0.715	0.065	231.95

of aeroplanes and cars correspond to these two classes (Figure 4.18, bottom). We show qualitative results of the joint setting on aeroplanes and cars in Figure 4.21 (middle).

Results on CUB

We also evaluate our method on the CUB dataset. Results in terms of foreground mask IoU and texture metrics (SSIM [225], L1 and FID [80, 131]) are reported in Table 4.28. Differently from the previous case, the CUB dataset does not have a clear subdivision in classes and literature approaches have only tested on the whole dataset. Thus, we test our method in two different settings. On the one hand, we evaluate the use of a single meanshape (as done by competitors). On the other hand, we test our method initializing N deformable meanshapes, as done in previous experiments. We empirically set $N = 14$, which is equal to the number of different values of the annotated categorical attribute “has_shape”. As shown, even if this dataset does contain objects of the same class “bird”, our method obtains comparable results with respect to literature approaches, on both shape and texture metrics. Even if the experiment with multiple shapes does not seem to increase the overall scores, it produces a set of insightful meanshapes learned in an unsupervised manner, as shown in Figure 4.19. Qualitative results are reported in Figure 4.21 (top).

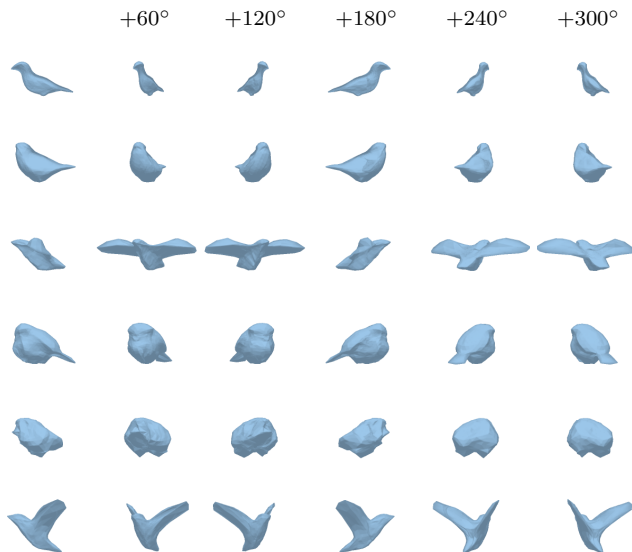


Figure 4.19: Some of the meanshapes learned training on CUB using our method, initialized with 14 spherical meanshapes.

Ablation study

In this section, we investigate the impact of using one or multiple meanshapes. In addition, we evaluate the influence of the dynamic subdivision approach compared to the static one. In these experiments, we use the Pascal3D+ dataset and extract precise foreground masks with PointRend [103]. Additional ablation studies and qualitative results are available in the supplementary material of our paper [192].

Unsupervised shape selection. As our first analysis, we evaluate the impact of the proposed unsupervised shape selection, which enables the training with multiple meanshapes and classes. We test three different training settings using the following object categories: (i) *aeroplane, car*, (ii) *bicycle, bus, car, motorbike*, (iii) all the 12 Pascal3D+ classes. Each setting has been tested using both a single meanshape or a set of N meanshapes, in order to verify the contribution of the usage of multiple learnable shapes and their unsupervised selection. The obtained results are reported in

Table 4.29: Ablation study comparing the usage of several meanshapes (our proposal) against a single meanshape (as a baseline) on Pascal3D+ using segmentation masks obtained with PointRend [103].

Training classes	Number of meanshapes	3D IoU \uparrow	Mask IoU \uparrow		Texture metrics		
			Pred cam	GT cam	SSIM \uparrow	L1 \downarrow	FID \downarrow
aeroplane, car	1	0.532	0.592	0.689	0.736	0.066	365.01
aeroplane, car	2	0.552	0.671	0.702	0.737	0.062	344.80
bicycle, bus, car, motorbike	1	0.517	0.665	0.751	0.601	0.100	390.41
bicycle, bus, car, motorbike	4	0.543	0.711	0.759	0.607	0.094	380.15
12 Pascal3D+ classes	1	0.409	0.602	0.670	0.660	0.088	357.51
12 Pascal3D+ classes	12	0.425	0.620	0.685	0.665	0.086	345.90

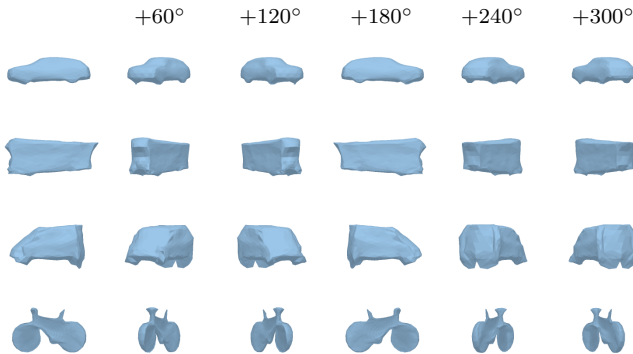


Figure 4.20: Meanshapes learned during training on the classes bicycle, bus, car, motorbike of Pascal3D+.

Table 4.29 in terms of 3D IoU, foreground mask IoU and texture metrics. Our approach with multiple meanshapes provide the best results in all the experimental settings. Furthermore, the meanshapes learned with the four-category setting are depicted in Figure 4.20. Even if the meanshapes do not exactly correspond to the four classes (*e.g.* the motorbike is missing), the meanshapes are meaningful and represent different object categories. Qualitative results are shown in Figure 4.21 for settings (i) and (ii) and in Figure 4.22 for setting (iii).

Dynamic mesh subdivision. We evaluate the contribution of the dynamic mesh subdivision during the training process using the four automotive

Table 4.30: Ablation study comparing different subdivision levels on Pascal3D+. Model trained on 4 classes (bicycle, bus, car, motorbike) using 4 meanshapes.

Subdivision level	Mask IoU \uparrow		Texture metrics		
	Pred cam	GT cam	SSIM \uparrow	L1 \downarrow	FID \downarrow
3	0.701	0.759	0.600	0.096	395.96
4	0.685	0.756	0.593	0.101	385.68
3 \rightarrow 4	0.711	0.759	0.607	0.094	380.15

classes. We compare three different settings of the 3D mesh connectivity, in terms of icosphere subdivision level: (i) level set to 3, (ii) level set to 4 and (iii) dynamic subdivision starting from level 3 and going up to level 4. Results are reported in Table 4.30. As shown, the method can converge to good results even using a fixed subdivision level. However, a higher level does not always lead to better scores, as in the case of fixed subdivision level 4. On the contrary, increasing the subdivision level during training leads to higher results in terms of both mask IoU and texture metrics. Indeed, dynamic subdivision allows to take advantage of low subdivision levels during the initial training phase, optimizing the shape smoothness in a faster and easier way, and at the same time leveraging the higher number of faces of high subdivision levels in the second part of the training, improving the finer details and the quality of the texture.

4.5.3 Discussion

Results reported in the previous section confirm that the 3D mesh reconstruction of objects can be learned jointly on multiple classes using only foreground masks and coarse camera poses as supervision. The accuracy is on-par or higher than the one of literature competitors while learning multiple meanshapes and predicting multiple classes at once with a single model.

Looking at the learned meanshapes, which are shown in Figures 4.18, 4.19 and 4.20, it is clear the method learns meaningful meanshapes in all the tested settings. From an inspection of the meanshapes during the training process, it seems that the method distinguishes different object





























































Class	Input image I	Weighted meanshape M	Predicted shape \hat{M}	Predicted shape with texture $\hat{M} + \hat{I}_{\text{tex}}$		
Birds						
						
						
						
Aeroplane						
Car						
Bicycle						
Bus						
Car						
Motorbike						

Figure 4.21: Qualitative results on different settings: CUB (birds) and Pascal3D+ (aeroplane and car, 4 automotive classes). We show the input image I , the output M of the unsupervised shape selection module, the predicted shape \hat{M} and the predicted textured shape $\hat{M} + \hat{I}_{\text{tex}}$ under several 3D rotations over the vertical axis of the predicted pose $\hat{\pi}$.

categories within the first few epochs and then progressively optimize each meanshape accordingly. Nevertheless, there are seldom a few meanshapes that do not correspond to a clear object category. Evaluating their impact according to the weight given to them by the unsupervised shape selection module throughout the test set, we discovered that these meanshapes have a marginal impact on the weighted meanshape. On the contrary, the most representative meanshapes have, on average, a major contribution on the weighted one.

















































Class	Input image I	Weighted meanshape M	Predicted shape \hat{M}	Predicted shape with texture $\hat{M} + \hat{I}_{\text{tex}}$
Aeroplane				
Bicycle				
Boat				
Bottle				
Bus				
Car				
Chair				
Table				
Motorbike				
Sofa				
Train				
TV Monitor				

Figure 4.22: Qualitative results on all the 12 classes of Pascal3D+. We show the input image I , the output M of the unsupervised shape selection module, the predicted shape \hat{M} and the predicted textured shape $\hat{M} + \hat{I}_{\text{tex}}$ under several 3D rotations over the vertical axis of the predicted pose $\hat{\pi}$.

Finally, it is worth looking at the qualitative results reported in Figure 4.21 and Figure 4.22. As it can be seen, the proposed method predicts meaningful weighted meanshapes, which are further refined applying the predicted vertex deformation. The object pose is remarkably precise too, arguably thanks to the direct pose supervision. Looking at the appearance,

the textured shapes resemble the objects and animals shown in the input images and they are consistent regardless of the rendered point of view. In general, however, the textures lack in fine details.

Chapter 5

Conclusions and Future Work

In this thesis, we aimed at improving the current human-vehicle interaction means by investigating a new vision-based approach to the field. We studied computer vision techniques using mainly non-RGB data, such as depth maps and infrared images, and applied them on several complementary tasks, trying to cover the most common interactions between the car and its users. Given the scarcity of public databases designed for the depth map-based or infrared-based human-vehicle interaction, we collected several datasets, presented them in Chapter 3, and released them to the public. Moreover, we developed several new methods and architectures, evaluated them on newly collected and public datasets and compared the results against the literature competitors, obtaining on-par or state-of-the-art results, as shown in Chapter 4. During the design of the datasets and the proposed methods, as well as during their evaluation, we took into account the strict requirements posed by the automotive industry. In particular, we considered the illumination invariance, the non-invasiveness, and the low latency, as discussed in Chapter 2. Overall, we believe that the datasets and methods presented in this thesis can serve as the foundations of a new vision-based approach to the human-vehicle interaction. In the following, we draw conclusions from the achieved results and look at possible future works and their industrial applications.

5.1 Depth Map Representations for Face Recognition

In Section 4.1, we conducted an extensive comparison on the use of depth maps and deep learning-based approaches. We investigated how data representations, network architectures, pre-processing and normalization techniques affect the accuracy of the face recognition task using depth maps. We presented the results that were obtained on four public datasets with multiple intra- and cross-dataset tests that suggest that depth maps should not be represented and treated as standard images. The results show that pre-processing and data normalization techniques, applied in combination with convolutional networks, reduce the 3D content of the depth data, making the corresponding systems less capable of generalizing and transferring to other depth domains, *e.g.* different sensors and acquisition setups. Representations that are based on normal images and, in particular, point clouds alleviate this problem and result in models with better generalization capabilities. In Section 3.1.1, we also presented a new challenging dataset, called MultiSFace, which contains facial data that were acquired using different synchronized sensors and in different conditions, *i.e.* at different sensor-subject distances. The results obtained on this dataset reveal the need for a proper face recognition method that is invariant to the acquisition sensor and setting and, in general, capable of fully exploiting the 3D content of depth maps.

Future work will study face recognition methods based on depth maps in order to make them invariant to the acquisition setting and to the recording sensor. Recognition systems that can handle different depth map qualities, resolutions and noise patterns are needed in real-world scenarios where sensors having different characteristics are used when registering into the system and when verifying the user’s identity. Future work will also review the performance and the computational requirements of different methods and ways to improve their optimization, with the aim of finding an efficient and effective method that runs on power- or computational-limited embedded boards.

5.2 Dynamic Hand Gesture Recognition

We presented a new dataset of dynamic hand gestures, that we recorded in a car simulator, in Section 3.1.2. We designed the performed gestures for the control of an in-vehicle infotainment system and released the data at the disposal of the research community. In Section 4.2, we further proposed a transformer-based architecture for the dynamic hand gesture recognition task. Through an extensive evaluation we showed how the frame-level feature extraction and the temporal aggregation computed by the transformer, starting from depth and surface normals combined through a late fusion approach, achieves state-of-the-art results. Moreover, we investigated the use of other data types usually provided by RGB-D sensors, such as color and infrared images. We evaluated our method on the proposed dataset, *i.e.* Briareo, and on the Nvidia Dynamic Hand Gesture dataset. Experimental results confirm the effectiveness of the proposed method in the automotive context, where the illumination invariance is extremely important, as discussed in Section 2.1. In addition, the computational performance analysis showed that the framework is able to run in real time and requires a limited amount of memory, making it suitable for an in-car infotainment system.

Even though the temporal flow is explicitly encoded into the transformer-based architecture, there are subsets of temporally symmetric gestures that are occasionally confused. The main challenge that future work will address is the encoding of the temporal progression of the gesture and the correct classification of symmetric and similar gestures. Future work will also investigate different strategies of multimodal fusion and may collect further data, either real or virtual, if it emerges that is necessary to avoid overfitting.

5.3 Human Pose Estimation and Refinement from Depth Maps

Regarding the 2D human pose estimation from depth maps, in Section 3.1.3 we presented both an annotation refinement tool and a novel set of fine joint annotations for a representative subset of the Watch-n-Patch dataset, calling it Watch-R-Patch. Moreover, we investigated methods for the 2D human pose estimation from depth maps and its 3D conversion and refinement in Section 4.3. We proposed a deep model that performs the

human pose estimation by means of body joints, reaching state-of-the-art results on the challenging fine annotations of the Watch-R-Patch dataset. Then, addressing the 3D refinement of human poses from depth maps, we proposed RefiNet, a multi-stage and modular refinement framework which provides an accurate 3D human pose starting from a depth map and a coarse 2D pose. While the first module improves the 2D position of joints on the depth map, the second one converts and improves their 3D representation and the last one enhances the 3D absolute locations using point clouds. Experimental results on the ITOP dataset confirm that RefiNet steadily improves the baseline approach and its results are comparable to the ones of pure 3D models.

Future work will investigate the introduction of an adversarial loss, that could force the predicted poses to be human-like, thus correcting potential errors in the initial 2D estimation. Furthermore, we plan to evaluate the system on different public datasets and test it in a real-world setting, in combination with estimators of anthropometric measurements.

5.4 Estimation of Anthropometric Measurements

We presented a new dataset for the contact-free estimation of anthropometric measurements from visual data in Section 3.1.4. The dataset is specifically designed for the automotive context and includes both in-car and outside views. To the best of our knowledge, this is the first publicly available dataset of this kind. In Section 4.4, we proposed a set of baselines for the estimation of the anthropometric measurements and evaluate them against different settings and data modalities. Results show that the estimation of body measures and self-biometric traits from visual data is feasible and can be achieved using geometric techniques, machine learning approaches and deep neural networks.

Future work will study multimodal and point cloud-based algorithms for anthropometric measurements, going beyond the unimodal approaches presented in this thesis. In addition, future work will investigate the radiometric thermal data that are included in the Baracca dataset. They could be used, for instance, to estimate the thermal comfort of the car passengers.

5.5 3D Reconstruction of Vehicles

In Section 4.5, we proposed a new method for the 3D object reconstruction, in terms of shape, pose and texture, from a single 2D image. We showed how the 3D mesh reconstruction of objects can be learned jointly on multiple classes using only foreground masks and coarse camera poses as supervision. The proposed approach is able to discern between different object categories and to learn meaningful category-level meanshapes in an unsupervised manner. In addition, we introduced a novel approach to predict the instance-specific deformation at vertex level, obtaining smooth deformations and the ability to dynamically subdivide the mesh during the training process. Quantitative and qualitative results on two public datasets, *i.e.* CUB and Pascal3D+, show the effectiveness of the proposed method, that obtains state-of-the-art or competitive results while learning multiple meanshapes in an unsupervised manner.

Future work will address the main drawback of this work, *i.e.* the dependency on coarse camera poses. Indeed, the method is trained supervising the camera poses to avoid the model collapse or degenerate solutions. However, recent methods have shown that the camera supervision can be successfully removed in the single-category setting. Thus, we aim at applying similar approaches also to the multi-category scenario. Future work will also study the development of improved texture prediction modules and investigate the use of the latest differentiable renderers.

Acknowledgements

Extraordinary people have accompanied me throughout my PhD. In this limited space, I do my best to acknowledge all of them and to say to them: please keep in touch and don't be a stranger!

First of all, I'd like to thank my PhD supervisors, Prof. Rita Cucchiara and Prof. Roberto Vezzani, for their guidance and the insightful discussions. I am grateful for their continuous support, regardless of my sometimes questionable behaviour. I also acknowledge the Ferrari S.p.A., which supported the RedVision Laboratory during my PhD and gave me the opportunity to experience what it means to work with a renowned automotive company. Besides, I'd like to thank the reviewers of my thesis, Bob Fisher and Matteo Poggi, for the insightful comments they provided.

I would also like to thank my dear friends Luca and Carmen, and every colleague from AImageLab. I acknowledge Federico, Matteo, Angelo and Federico, that shared these years pursuing a PhD with me. I thank my friends and colleagues Guido, Andrea and Alessandro, that managed to work with me for so long. I acknowledge Riccardo, Fabio and Matteo, time at AImageLab would have been much more boring (and quiet) without them.

Then, I'd like to thank the organizers, the speakers and the attendees of the 2018 Cornell, Maryland, Max Planck Pre-doctoral Research School. Attending the school was an amazing and unforgettable experience. It further persuaded me to pursue a PhD in computer science and gave me the chance to meet so many wonderful people from all over the world. Among those people, I met Alberto, possibly the kindest friend I could ever imagine. I want to truly thank him for sharing his time with me; he has taught me so much without even noticing.

I'd also like to acknowledge my invaluable longtime friends, who still stand my company after so many years. Listing them in alphabetical order, I'd like to thank Alessandra, Andrea, Di, Elena, Giacomo, Gianluca, Giuliana, Ilaria, Lorenzo, Luca, Matteo, Riccardo, Sara, Sofia, Stefano.

Last but not least, I would like to express my gratitude to my family. They've constantly taken care of me and supported me way beyond my academic career. I hope I'll be able to repay them for their love.

Appendix A

List of publications

In this Appendix, we report the research papers published during the PhD in chronological order. The articles that are discussed in this thesis present the list of related sections after the publication details.

Content and experimental results published in some of these papers have been included, even *verbatim*, in the previous chapters.

- (i) Stefano Pini, Marcella Cornia, Federico Bolelli, Lorenzo Baraldi, Rita Cucchiara. M-VAD Names: a dataset for video captioning with naming. In *Multimedia Tools and Applications*, 78(10), pp. 14007-14027. Springer, 2019.
- (ii) Andrea D'Eusanio, Stefano Pini, Guido Borghi, Roberto Vezzani, Rita Cucchiara. Manual Annotations on Depth Maps for Human Pose Estimation. In *20th International Conference on Image Analysis and Processing (ICIAP)*. Springer, 2019. Sections 3.1.3, 4.3.1.
- (iii) Fabio Manganaro, Stefano Pini, Guido Borghi, Roberto Vezzani, Rita Cucchiara. Hand Gestures for the Human-Car Interaction: the Briareo dataset. In *20th International Conference on Image Analysis and Processing (ICIAP)*. Springer, 2019. Section 3.1.2.
- (iv) Stefano Pini, Guido Borghi, Roberto Vezzani, Rita Cucchiara. Video synthesis from Intensity and Event Frames. In *20th International Conference on Image Analysis and Processing (ICIAP)*. Springer, 2019.

- (v) F. M. Caputo, S. Burato, G. Pavan, T. Voillemin, H. Wannous, J. P. Vandeborre, M. Maghoumi, E. M. Taranta, A. Razmjoo, J. J. LaViola Jr., F. Manganaro, S. Pini, G. Borghi, R. Vezzani, R. Cucchiara, H. Nguyen, M. T. Tran, A. Giachetti. Online Gesture Recognition. In *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019.
- (vi) Guido Borghi, Stefano Pini, Roberto Vezzani, Rita Cucchiara. Driver Face Verification with Depth Maps. In *Sensors*, 19(15), 3361. MDPI, 2019.
- (vii) Stefano Pini, Guido Borghi, Roberto Vezzani. Learn to See by Events: Color Frame Synthesis from Event and RGB Cameras. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020)*, Volume 4: VISAPP, pp. 37-47. Scitepress, 2020.
- (viii) Guido Borghi, Stefano Pini, Roberto Vezzani, Rita Cucchiara. Mercury: a vision-based framework for Driver Monitoring. In *International Conference on Intelligent Human Systems Integration*, pp. 104-110. Springer, 2020.
- (ix) Stefano Pini, Andrea D'Eusanio, Guido Borghi, Roberto Vezzani, Rita Cucchiara. Baracca: a Multimodal Dataset for Anthropometric Measurements in Automotive. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2020. *Sections 3.1.4, 4.4*.
- (x) Riccardo Gasparini, Stefano Pini, Guido Borghi, Giuseppe Scaglione, Simone Calderara, Eugenio Fedeli, Rita Cucchiara. Anomaly Detection for Vision-Based Railway Inspection. In *European Dependable Computing Conference Workshops*, pp. 56-67. Springer, 2020.
- (xi) Andrea D'Eusanio, Alessandro Simoni, Stefano Pini, Guido Borghi, Roberto Vezzani, Rita Cucchiara. Multimodal Hand Gesture Classification for the Human–Car Interaction. In *Informatics*, 7(3), 31. MDPI, 2020.
- (xii) Andrea D'Eusanio, Alessandro Simoni, Stefano Pini, Guido Borghi, Roberto Vezzani, Rita Cucchiara. A Transformer-Based Network for Dynamic Hand Gesture Recognition. In *2020 International Conference on 3D Vision (3DV)*, pp. 623-632. IEEE, 2020. *Section 4.2*.

- (xiii) Andrea D'Eusanio, Stefano Pini, Guido Borghi, Roberto Vezzani, Rita Cucchiara. RefiNet: 3D Human Pose Refinement with Depth Maps. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 2320-2327. IEEE, 2021. *Section 4.3.2.*
- (xiv) Riccardo Gasparini, Andrea D'Eusanio, Guido Borghi, Stefano Pini, Giuseppe Scaglione, Simone Calderara, Eugenio Fedeli, Rita Cucchiara. Anomaly Detection, Localization and Classification for Railway Inspection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 3419-3426. IEEE, 2021.
- (xv) Alessandro Simoni, Andrea D'Eusanio, Stefano Pini, Guido Borghi, Roberto Vezzani. Improving Car Model Classification through Vehicle Keypoint Localization. In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021)*, Volume 5: VISAPP, pp. 354-361. Scitepress, 2021.
- (xvi) Luca Bergamini, Stefano Pini, Alessandro Simoni, Roberto Vezzani, Simone Calderara, Rick B. D'Eath, Robert B. Fisher. Extracting Accurate Long-Term Behavior Changes from a Large Pig Dataset. In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021)*, Volume 5: VISAPP, pp. 524-533. Scitepress, 2021.
- (xvii) Stefano Pini, Guido Borghi, Roberto Vezzani, Davide Maltoni, Rita Cucchiara. A Systematic Comparison of Depth Map Representations for Face Recognition. In *Sensors*, 21(3), 944. MDPI, 2021. *Sections 3.1.1, 4.1.*
- (xviii) Stefano Pini, Guido Borghi, Roberto Vezzani. Video Frame Synthesis Combining Conventional and Event Cameras. In *International Journal of Pattern Recognition and Artificial Intelligence*. World Scientific, 2021.
- (xix) A. Caputo, A. Giachetti, S. Soso, D. Pintani, A. D'Eusanio, S. Pini, G. Borghi, A. Simoni, R. Vezzani, R. Cucchiara, A. Ranieri, F. Giannini, K. Lupinetti, M. Monti, M. Maghoumi, J. J. LaViola Jr, M. Le, H. Nguyen, M. T. Tran. SHREC 2021: Skeleton-based hand gesture

recognition in the wild. In *Computers & Graphics*, Volume 99, pp. 201-211. Elsevier, 2021.

- (xx) Alessandro Simoni, Stefano Pini, Roberto Vezzani, Rita Cucchiara. Multi-Category Mesh Reconstruction From Image Collections. In *2021 9th International Conference on 3D Vision (3DV)*. IEEE, 2021. *Section 4.5*.

Appendix B

Activities carried out during the PhD

In this Appendix, we briefly report the additional research and academic activities carried out during the 3 years of the PhD at the University of Modena and Reggio Emilia.

National and foreign collaborations and projects

- Member of RedVision Laboratory, a 3 year-long collaboration between the University of Modena and Reggio Emilia and Ferrari S.p.A. - Modena, Italy - 2018 to 2021
- Epidetect, a 6-month project on early detection of epidemic outbreaks in collaboration with MAPS S.p.A. - Modena, Italy - July 2020 to January 2021
- Research internship at Level 5 (formerly part of Lyft, now part of Woven Planet Holdings) - London, UK - April to October 2021

Teaching activities

- Teaching Assistant for the BSc course “Fundamentals of Computer Science” (C programming) - Academic Year 2019-2020

- Tutor for the course “Neural network computing AI, and machine learning for automotive”, co-supervisor of the project “Driver Fatigue Detection: Yawn Detection from Facial Landmarks” led by MSc students Di Pinto Valentina, Goldoni Daniele, Mattioli Alessandro - Academic Year 2020-2021
- Lecture titled “How to build a self-driving car: Perception, Motion Planning and Simulation for Autonomous Vehicles”, given to the MSc students of the course “Neural Network Computing, AI and Machine Learning for Automotive” - Academic Year 2021-2022

MSc thesis co-advisor

- Andrea D'Eusano, “Body Pose Estimation and Hand Detection for Driver Monitoring”, 2019
- Fabio Manganaro, “Gesture recognition for the human-vehicle interaction through infrared and 3D sensors”, 2019
- Vincenzo Maria Seritti, “Pose analyses and anthropometric measurements of people driving through Time-of-Flight cameras”, 2019
- Fabrizio Di Blasi, “3D human pose refinement through synthetic datasets”, 2020
- Francesco Suma, “3D pose estimation of collaborative robots from RGB-D images: algorithms and datasets”, 2020
- Giuseppe Bisaccia, “Generation of a synthetic automotive dataset and analysis of the main monocular depth estimation methods”, 2021

Other academic services

- Reviewer for the following conferences:
 - IEEE International Conference on Computer Vision, 2019
 - IEEE Winter Conference on Applications of Computer Vision, 2020
 - International Conference on Pattern Recognition, 2020

- CVPR Workshop on Autonomous Driving: Perception, Prediction and Planning (ADP3), 2021
- Reviewer for the following journals:
 - Transactions on Multimedia, IEEE
 - Multimedia Tools and Applications, Springer
 - The Visual Computer, Springer
 - Signal Processing: Image Communication, Elsevier
 - Transactions on Multimedia Computing Communications and Applications, ACM
 - Robotics and Automation Letters, IEEE

Attended conferences, schools, courses and seminars

Conferences

- 20th International Conference on Image Analysis and Processing (ICIAP) - Trento, Italy - 2019
- 15th International Conference on Computer Vision Theory and Applications (VISAPP) - Valletta, Malta - 2020
- 2020 IEEE International Joint Conference on Biometrics (IJCB) - Houston, TX, USA (Online) - 2020
- 8th International Conference on 3D Vision (3DV) - Fukuoka, Japan (Online) - 2020
- 25th International Conference on Pattern Recognition (ICPR) 2020 - Milan, Italy (Online) - 2021
- 16th International Conference on Computer Vision Theory and Applications (VISAPP) - Vienna, Austria (Online) - 2021
- 9th International Conference on 3D Vision (3DV) - London, United Kingdom (Online) - 2021

Schools and courses

- Academic English Workshop I, 2019
- Academic English Workshop II, 2019
- ICT Technology Commercialization and Business Development for Engineers, 2019
- International Computer Vision Summer School (ICVSS), 2019
- Advanced Course on Data Science and Machine Learning (ACDL), 2020
- Mediterranean Machine Learning Summer School (M2L), 2021

Seminars and lectures

- Computer Graphics for Cultural Heritage Applications, Roberto Scopigno, March 2019
- Deep Learning-based Automatic Speech Recognition at IIT - A multimodal approach, Leonardo Badino, March 2019
- Big Data in Neuroimaging, Prof. Mark Jenkinson, June 2019
- Computational Aspects of Deep Reinforcement Learning, Iuri Frosio, July 2019
- Transferring Knowledge Across Domains: an Introduction to Deep Domain Adaptation, Massimiliano Mancini and Pietro Morerio, September 2019
- Probabilistic and deep learning for regression in computer vision, Xavier Alameda-Pineda and Stéphane Lathuilière, September 2019
- Developing 3-D Imaging and LIDAR Sensors: Problems and Technologies, Prof. Silvano Donati, December 2019
- Cloudy with high chance of DBMS: A 10-year prediction for Enterprise-Grade ML (and what we are doing to get there), Matteo Interlandi, December 2019

APPENDIX B – Activities carried out during the PhD

- Bias from the Wild, Prof. Nello Cristianini, May 2020
- Learning Representations and Geometry from Unlabelled Videos, Prof. Andrea Vedaldi, June 2020
- About Time, Prof. Arnold Smeulders, September 2020

Bibliography

- [1] Mahdi Abavisani, Hamid Reza Vaezi Joze, and Vishal M Patel. Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1165–1174, 2019. 23, 75, 78, 79, 80
- [2] Byungtae Ahn, Jaesik Park, and In So Kweon. Real-time head orientation from a monocular camera using deep neural network. In *Asian conference on computer vision*, pages 82–96. Springer, 2014. 19
- [3] Timo Ahonen, Esa Rahtu, Ville Ojansivu, and Janne Heikkila. Recognition of blurred faces using local phase quantization. In *International Conference on Pattern Recognition (ICPR)*. IEEE, 2008. 18
- [4] Virginia Ortiz Andersson and Ricardo Matsumura Araujo. Person identification using anthropometric and gait data from kinect sensor. In *29th AAAI Conference on Artificial Intelligence*, 2015. 106
- [5] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 85, 94
- [6] S Anith, D Vaithyanathan, and R Seshasayanan. Face recognition system based on feature extraction. In *IEEE International Conference on Information Communication and Embedded Systems*, 2013. 18
- [7] Tristan Aumentado-Armstrong, Alex Levinstein, Stavros Tsogkas, Konstantinos G Derpanis, and Allan D Jepson. Cycle-consistent gener-

- ative rendering for 2d-3d modality translation. In *2020 International Conference on 3D Vision (3DV)*, pages 230–240, 2020. 29
- [8] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. 3d constrained local model for rigid and non-rigid facial tracking. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2610–2617. IEEE, 2012. 57
- [9] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1670–1687, 2014. 12, 74
- [10] Stefano Berretti, Pietro Pala, and Alberto Del Bimbo. Face recognition by super-resolved 3d models from consumer depth cameras. *IEEE transactions on information forensics and security*, 9(9):1436–1449, 2014. 57
- [11] Paul J Besl and Ramesh C Jain. Invariant surface characteristics for 3d object recognition in range images. *Computer vision, graphics, and image processing*, 33(1):33–80, 1986. 12, 74
- [12] Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277, 2018. 67
- [13] Didier Bieler, Semih Gunel, Pascal Fua, and Helge Rhodin. Gravity as a reference for estimating a person’s height from video. In *IEEE International Conference on Computer Vision*, pages 8569–8577, 2019. 28
- [14] Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Fast gesture recognition with multiple stream discrete hmms on 3d skeletons. In *International Conference on Pattern Recognition (ICPR)*, pages 997–1002. IEEE, 2016. 70, 92
- [15] Guido Borghi, Riccardo Gasparini, Roberto Vezzani, and Rita Cucchiara. Embedded recurrent network for head pose estimation in car. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1503–1508. IEEE, 2017. 71

- [16] Guido Borghi, Marco Venturelli, Roberto Vezzani, and Rita Cucchiara. Poseidon: Face-from-depth for driver pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4661–4670, 2017. 11, 19, 44, 53, 58, 59, 61, 63, 64
- [17] Guido Borghi, Matteo Fabbri, Roberto Vezzani, Rita Cucchiara, et al. Face-from-depth for head pose estimation on depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 19
- [18] Guido Borghi, Elia Frigieri, Roberto Vezzani, and Rita Cucchiara. Hands on the wheel: a dataset for driver hand detection and tracking. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 564–570. IEEE, 2018. 24, 25
- [19] Guido Borghi, Stefano Pini, Filippo Grazioli, Roberto Vezzani, and Rita Cucchiara. Face verification from depth using privileged information. In *British Machine Vision Conference (BMVC)*, page 303, 2018. 11, 19, 44, 85
- [20] Guido Borghi, Stefano Pini, Roberto Vezzani, and Rita Cucchiara. Driver face verification with depth maps. *Sensors*, 19(15):3361, 2019. 19, 44, 53
- [21] Guido Borghi, Stefano Pini, Roberto Vezzani, and Rita Cucchiara. Mercury: a vision-based framework for driver monitoring. In *International Conference on Intelligent Human Systems Integration*, pages 104–110. Springer, 2020. 70, 71, 92
- [22] Said Yacine Boulahia, Eric Anquetil, Franck Multon, and Richard Kulpa. Dynamic hand gesture recognition based on 3d pattern assembled trajectories. In *International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2017. 24, 25
- [23] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL <https://doi.org/10.1023/A:1010933404324>. 108
- [24] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision*, 2016. 26

- [25] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018. 18, 53
- [26] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 25, 85, 86, 87, 92, 93, 99, 102
- [27] Fabio Marco Caputo, S Burato, G Pavan, T Voillemin, Hazem Wanoun, Jean-Philippe Vandeborre, Mehran Maghoumi, EM Taranta, A Razmjoo, JJ LaViola Jr, et al. Online gesture recognition. In *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019. 70
- [28] Marco Carraro, Matteo Munaro, and Emanuele Menegatti. Skeleton estimation and tracking by means of depth data fusion from depth camera networks. *Robotics and Autonomous Systems*, 2018. 92
- [29] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 22, 23, 78, 79, 80
- [30] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 25, 27
- [31] Wenzheng Chen, Huan Wang, Yangyan Li, Hao Su, Zhenhua Wang, Changhe Tu, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Synthesizing training images for boosting human 3d pose estimation. In *2016 International Conference on 3D Vision (3DV)*, 2016. 85
- [32] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. *Advances in Neural Information Processing Systems*, 32:9609–9619, 2019. 29, 123

- [33] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 26
- [34] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*. Springer, 2016. 29, 113
- [35] R. Dabral, N. B. Gundavarapu, R. Mitra, A. Sharma, G. Ramakrishnan, and A. Jain. Multi-person 3d human pose estimation from monocular images. In *2019 International Conference on 3D Vision (3DV)*, 2019. 92
- [36] Antitza Dantcheva, Carmelo Velardo, Angela D’angelo, and Jean-Luc Dugelay. Bag of soft biometrics for person identification. *Multimedia Tools and Applications*, 51(2):739–777, 2011. 106
- [37] Biman Das and Arijit K Sengupta. Industrial workstation design: a systematic ergonomics approach. *Applied ergonomics*, 27(3):157–163, 1996. 106
- [38] Mathieu De Coster, Mieke Van Herreweghe, and Joni Dambre. Sign language recognition with transformer networks. In *12th Int. Conf. on Language Resources and Evaluation*, 2020. 23
- [39] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. Heterogeneous hand gesture recognition using 3d dynamic skeletal data. *Computer Vision and Image Understanding*, 181:60–72, 2019. 22
- [40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 47, 77, 78, 79, 80, 86, 108
- [41] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 53

- [42] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Ar-face: Additive angular margin loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 18
- [43] Andrea D’Eusanio, Stefano Pini, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Manual annotations on depth maps for human pose estimation. In *International Conference on Image Analysis and Processing*, pages 233–244. Springer, 2019. 99
- [44] Andrea D’Eusanio, Stefano Pini, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Refinet: 3d human pose refinement with depth maps. In *International Conference on Pattern Recognition (ICPR)*, 2020. 102, 103, 104
- [45] Andrea D’Eusanio, Alessandro Simoni, Stefano Pini, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Multimodal hand gesture classification for the human–car interaction. *Informatics*, 7(3):31, 2020. 53, 70
- [46] Naina Dhingra and Andreas Kunz. Res3ATN - deep 3d residual attention network for hand gesture recognition in videos. In *2019 International Conference on 3D Vision (3DV)*, pages 491–501. IEEE, 2019. 22, 78
- [47] Yanchao Dong, Zhencheng Hu, Keiichi Uchimura, and Nobuki Murayama. Driver inattention monitoring system for intelligent vehicles: A review. *IEEE transactions on intelligent transportation systems*, 12(2):596–614, 2011. 70
- [48] Harris Drucker. Improving regressors using boosting techniques. In *International Conference on Machine Learning*, volume 97, pages 107–115, 1997. 108
- [49] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2): 124, 1971. 43
- [50] Abdessamad Elboushaki, Rachida Hannane, Karim Afdel, and Lahcen Koutti. Multid-cnn: A multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in rgb-d

- image sequences. *Expert Systems with Applications*, 139:112829, 2020. 70
- [51] Hilal Ergun, Yusuf Caglar Akyuz, Mustafa Sert, and Jianquan Liu. Early and late level fusion of deep convolutional neural networks for visual concept recognition. *International Journal of Semantic Computing*, 10(03):379–397, 2016. 75
- [52] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 2015. 47
- [53] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 605–613, 2017. 29
- [54] Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real time head pose estimation with random regression forests. In *CVPR 2011*, pages 617–624. IEEE, 2011. 43, 55, 58, 59, 60, 61, 63, 64
- [55] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003. 45, 74
- [56] Kai-ping Feng and Fang Yuan. Static hand gesture recognition based on hog characters and support vector machines. In *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*, pages 936–938. IEEE, 2013. 70
- [57] Mihai Fieraru, Anna Khoreva, Leonid Pishchulin, and Bernt Schiele. Learning to refine human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018. 27
- [58] Sergi Foix, Guillem Alenya, and Carme Torras. Lock-in time-of-flight (tof) cameras: A survey. *IEEE Sensors Journal*, 11(9):1917–1926, 2011. 8

- [59] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995. 108
- [60] Rikke Gade and Thomas B Moeslund. Thermal cameras and applications: a survey. *Machine vision and applications*, 25(1):245–262, 2014. 14
- [61] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252, 2017. 73
- [62] Silvio Giancola, Matteo Valenti, and Remo Sala. *A survey on 3D cameras: Metrological comparison of time-of-flight, structured-light and active stereoscopy technologies*. Springer, 2018. 8
- [63] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016. 29, 113
- [64] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 23
- [65] Ross Girshick, Jamie Shotton, Pushmeet Kohli, Antonio Criminisi, and Andrew Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *IEEE International Conference on Computer Vision*, 2011. 26
- [66] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *European Conference on Computer Vision*, 2020. 29, 47, 113, 115, 118, 120, 121, 123
- [67] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 216–224, 2018. 117

- [68] Ye-Peng Guan et al. Unsupervised human height estimation from a single image. *Journal of Biomedical Science and Engineering*, 2(06): 425, 2009. 28
- [69] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 85
- [70] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pages 87–102. Springer, 2016. 18, 53
- [71] JunYoung Gwak, Christopher B Choy, Manmohan Chandraker, Animesh Garg, and Silvio Savarese. Weakly supervised 3d reconstruction with adversarial constraint. In *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017. 29, 113
- [72] Rostam Affendi Hamzah and Haidi Ibrahim. Literature survey on stereo vision disparity map algorithms. *Journal of Sensors*, 2016, 2016. 8
- [73] Miles Hansard, Seungkyu Lee, Ouk Choi, and Radu Patrice Horaud. *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media, 2012. 8
- [74] Albert Haque, Boya Peng, Zelun Luo, Alexandre Alahi, Serena Yeung, and Li Fei-Fei. Towards viewpoint invariant 3d human pose estimation. In *European Conference on Computer Vision*. Springer, 2016. 26, 46, 99, 100, 102, 103, 104
- [75] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *IEEE International Conference on Computer Vision*, pages 991–998, 2011. 47
- [76] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and H-P Seidel. A statistical model of human pose and body shape. In *Computer graphics forum*. Wiley Online Library, 2009. 27

- [77] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 18, 22, 53, 57, 61, 73, 96, 108, 117
- [78] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 47
- [79] Paul Henderson and Vittorio Ferrari. Learning to generate and reconstruct 3d meshes with only 2d supervision. In *British Machine Vision Conference (BMVC)*, 2018. 29
- [80] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 2017. 123
- [81] Radu Horaud, Miles Hansard, Georgios Evangelidis, and Clément Ménéier. An overview of depth cameras and range scanners based on time-of-flight technologies. *Machine vision and applications*, 27(7): 1005–1020, 2016. 8
- [82] Yu Hu, Yongkang Wong, Wentao Wei, Yu Du, Mohan Kankanhalli, and Weidong Geng. A novel attention-based hybrid cnn-rnn architecture for semg-based gesture recognition. *PloS one*, 13(10):e0206049, 2018. 70
- [83] Zhenguo Hu, Penghui Gui, Ziqing Feng, Qijun Zhao, Keren Fu, Feng Liu, and Zhengxi Liu. Boosting depth-based face recognition from a quality perspective. *Sensors*, 19(19):4124, 2019. 19
- [84] Chen Huang, Yining Li, Change Loy Chen, and Xiaoou Tang. Deep imbalanced learning for face recognition and attribute prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 18
- [85] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on Faces in*

- 'Real-Life' Images: Detection, Alignment, and Recognition, 2008. 18, 53
- [86] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992. 108
- [87] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in Neural Information Processing Systems*, pages 2807–2817, 2018. 29
- [88] Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. Moviereshape: Tracking and reshaping of humans in videos. *ACM Transactions on Graphics*, 29(6):1–10, 2010. 28
- [89] Bongjin Jun, Inho Choi, and Daijin Kim. Local transform features and hybridization for accurate face and human detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 18
- [90] Ho Yub Jung, Yumin Suh, Gyeongsik Moon, and Kyoung Mu Lee. A sequential approach to 3d human pose estimation: Separation of localization and identification of body joints. In *European Conference on Computer Vision*, 2016. 26
- [91] Ioannis A Kakadiaris, Georgios Passalis, George Toderici, Mohammed N Murtuza, Yunliang Lu, Nikos Karampatziakis, and Theoharis Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Transactions on pattern analysis and machine intelligence*, 29(4): 640–649, 2007. 20
- [92] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *European Conference on Computer Vision*, pages 371–386, 2018. 29, 47, 113, 115, 120, 121, 123
- [93] Juho Kannala and Esa Rahtu. Bsif: Binarized statistical image features. In *International Conference on Pattern Recognition (ICPR)*. IEEE, 2012. 18
- [94] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *IEEE*

- Conference on Computer Vision and Pattern Recognition*, pages 1966–1974, 2015. 29, 121
- [95] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 29, 115
- [96] Hiroharu Kato and Tatsuya Harada. Self-supervised learning of 3d objects from natural images. *preprint arXiv:1911.08850*, 2019. 29
- [97] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018. 29
- [98] Arie Kaufman, Daniel Cohen, and Roni Yagel. Volume graphics. *Computer*, 26(7):51–64, 1993. 13, 21
- [99] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 78, 79, 80
- [100] Güneş Kaym, Cihan Sarı, and Ceyhan Burak Akgül. Facial feature selection for gender recognition based on random decision forests. In *21st Signal Processing and Communications Applications Conference*. IEEE, 2013. 18
- [101] Donghyun Kim, Matthias Hernandez, Jongmoo Choi, and Gérard Medioni. Deep 3d face identification. *International Joint Conference on Biometrics (IJCB)*, 2017. 20
- [102] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 77, 87, 96, 99, 108, 121
- [103] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9799–9808, 2020. 47, 124, 125

- [104] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *IEEE International Conference on Computer Vision*, pages 863–872, 2017. 20
- [105] Okan Kopuklu, Neslihan Kose, and Gerhard Rigoll. Motion fused frames: Data level fusion strategy for hand gesture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2103–2111, 2018. 23, 80
- [106] Okan Köpüklü, Ahmet Gunduz, Neslihan Kose, and Gerhard Rigoll. Real-time hand gesture detection and classification using convolutional neural networks. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE, 2019. 22
- [107] Alexander Kozlov, Vadim Andronov, and Yana Gritsenko. Lightweight network architecture for real-time action recognition. In *35th Annual ACM Symposium on Applied Computing*, 2020. 23
- [108] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 60
- [109] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *IEEE International Conference on Computer Vision*, pages 2202–2211, 2019. 29, 30
- [110] Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 452–461, 2020. 29, 30
- [111] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 8
- [112] Truc Le and Ye Duan. Pointgrid: A deep network for 3d shape understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9204–9214, 2018. 21

- [113] Hyeon-Kyu Lee and Jin-Hyung Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973, 1999. 70
- [114] Yuan-Cheng Lee, Jiancong Chen, Ching Wei Tseng, and Shang-Hong Lai. Accurate and robust face recognition from rgb-d images with a deep learning approach. In *British Machine Vision Conference (BMVC)*, 2016. 19
- [115] Grégoire Lefebvre, Samuel Berlemont, Franck Mamalet, and Christophe Garcia. Blstm-rnn based 3d gesture classification. In *International conference on artificial neural networks*, pages 381–388. Springer, 2013. 70
- [116] Jiahui Lei, Srinath Sridhar, Paul Guerrero, Minhyuk Sung, Niloy Mitra, and Leonidas J Guibas. Pix2surf: Learning parametric 3d surface models of objects from images. In *European Conference on Computer Vision*, 2020. 30
- [117] MJPM Lemmens. A survey on stereo matching techniques. *International Archives of Photogrammetry and Remote Sensing*, 27(B8): 11–23, 1988. 8
- [118] Billy YL Li, Ajmal S Mian, Wanquan Liu, and Aneesh Krishna. Using kinect for face recognition under varying poses, expressions, illumination and disguise. In *2013 IEEE workshop on applications of computer vision (WACV)*, pages 186–192. IEEE, 2013. 19, 43, 44, 58, 59, 61, 63, 64
- [119] Hai Li, Weicai Ye, Guofeng Zhang, Sanyuan Zhang, and Hujun Bao. Saliency guided subdivision for single-view mesh reconstruction. In *2020 International Conference on 3D Vision (3DV)*, pages 1098–1107. IEEE, 2020. 29, 118
- [120] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. In *Advances in Neural Information Processing Systems*, 2020. 30
- [121] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d

- reconstruction via semantic consistency. In *European Conference on Computer Vision*, 2020. 29, 115, 121
- [122] Chen-Hsuan Lin, Oliver Wang, Bryan C Russell, Eli Shechtman, Vladimir G Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 969–978, 2019. 29, 113
- [123] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 42, 85, 92, 94
- [124] Han Liu, Feixiang He, Qijun Zhao, and Xiangdong Fei. Matching depth to rgb for boosting face verification. In *Chinese Conference on Biometric Recognition*. Springer, 2017. 19
- [125] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *IEEE International Conference on Computer Vision*, pages 7708–7717, 2019. 29, 116
- [126] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. SpheroFace: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 18, 53
- [127] Weiyuan Liu. Natural user interface-next mainstream product user interface. In *2010 IEEE 11th International Conference on Computer-Aided Industrial Design & Conceptual Design 1*, volume 1, pages 203–205. IEEE, 2010. 70
- [128] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*, pages 963–973, 2019. 21
- [129] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 18, 53

- [130] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014. 29
- [131] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in Neural Information Processing Systems*, 2018. 123
- [132] Priyanka Mandikal, Navaneet KL, and R Venkatesh Babu. 3d-psrnet: Part segmented 3d point cloud reconstruction from a single image. In *European Conference on Computer Vision Workshops*, 2018. 29
- [133] Fabio Manganaro, Stefano Pini, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Hand gestures for the human-car interaction: the briareo dataset. In *International Conference on Image Analysis and Processing*, pages 560–571. Springer, 2019. 70, 83, 84, 92
- [134] Tomas Mantecon, Carlos R del Bianco, Fernando Jaureguizar, and Narciso García. Depth-based face recognition using local quantized patterns adapted for range data. In *IEEE International Conference on Image Processing*, pages 293–297. IEEE, 2014. 19, 57
- [135] Tomás Mantecón, Carlos R del Blanco, Fernando Jaureguizar, and Narciso García. Visual face recognition using bag of dense derivative depth patterns. *IEEE Signal Processing Letters*, 2016. 19
- [136] Giulio Marin, Fabio Dominio, and Pietro Zanuttigh. Hand gesture recognition with leap motion and kinect devices. In *IEEE International Conference on Image Processing (ICIP)*, pages 1565–1569. IEEE, 2014. 23, 24
- [137] Manuel Martin, Alina Roitberg, Monica Haurilet, Matthias Horne, Simon Reiß, Michael Voit, and Rainer Stiefelwagen. Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles. In *IEEE International Conference on Computer Vision*, pages 2801–2810, 2019. 53
- [138] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *IEEE International Conference on Computer Vision*, 2017. 97

- [139] Iacopo Masi, Yue Wu, Tal Hassner, and Prem Natarajan. Deep face recognition: A survey. In *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pages 471–478. IEEE, 2018. 56
- [140] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The jester dataset: A large-scale video dataset of human gestures. In *IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 22, 80
- [141] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015. 21, 57, 62
- [142] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *2018 International Conference on 3D Vision (3DV)*, 2018. 92
- [143] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz. Hand gesture recognition with 3d convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7, 2015. 24
- [144] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4207–4215, 2016. 22, 23, 24, 45, 71, 73, 75, 77, 78, 79, 80, 84
- [145] Mahdi Momeni-k, Sotirios Ch Diamantas, Fabio Ruggiero, and Bruno Siciliano. Height estimation from a single camera view. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 358–364, 2012. 28
- [146] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5088, 2018. 21

- [147] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Posefix: Model-agnostic general human pose refinement network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 27
- [148] MSCOCO. COCO - Keypoint Evaluation. <http://cocodataset.org/#keypoints-eval>, 2016. 88
- [149] Guodong Mu, Di Huang, Guosheng Hu, Jia Sun, and Yunhong Wang. Led3d: A lightweight and efficient deep approach to recognizing low-quality 3d faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5773–5782, 2019. 11, 19, 20
- [150] Y. Nakagawa, H. Uchiyama, H. Nagahara, and R. Taniguchi. Estimating surface normals with depth image gradients for fast and accurate registration. In *2015 International Conference on 3D Vision*, pages 640–647, Oct 2015. 12
- [151] Yosuke Nakagawa, Hideaki Uchiyama, Hajime Nagahara, and Rin-ichiro Taniguchi. Estimating surface normals with depth image gradients for fast and accurate registration. In *2015 International Conference on 3D Vision (3DV)*, pages 640–647. IEEE, 2015. 74
- [152] Pradyumna Narayana, Ross Beveridge, and Bruce A Draper. Gesture recognition: Focus on the hands. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5235–5244. IEEE, 2018. 22
- [153] João Baptista Cardia Neto and Aparecido Nilceu Marana. Utilizing deep learning and 3dlbp for 3d face recognition. In *Iberoamerican Congress on Pattern Recognition*, pages 135–142. Springer, 2017. 19
- [154] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*. Springer, 2016. 25, 26
- [155] Xuan Son Nguyen, Thanh Phuong Nguyen, and François Charpillet. Effective surface normals based action recognition in depth images. In *International Conference on Pattern Recognition (ICPR)*, pages 817–822. IEEE, 2016. 74
- [156] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based

- approach and evaluations. *IEEE transactions on intelligent transportation systems*, 15(6):2368–2377, 2014. 23, 24, 53
- [157] Andrea Palazzi, Luca Bergamini, Simone Calderara, and Rita Cucchiara. End-to-end 6-dof object pose estimation through differentiable rasterization. In *European Conference on Computer Vision Workshops*, 2018. 29
- [158] Andrea Palazzi, Luca Bergamini, Simone Calderara, and Rita Cucchiara. Warp and learn: Novel views generation for vehicles and other objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 47
- [159] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 117
- [160] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 120
- [161] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, 6:3, 2017. 67
- [162] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *Neural Information Processing Systems Workshops*, 2017. 85, 121
- [163] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. In *British Machine Vision Conference (BMVC)*, 2018. 119
- [164] Angkoon Phinyomark, Franck Quaine, and Yann Laurillau. The relationship between anthropometric variables and features of electromyography signal for human–computer interface. In *Applications*,

- Challenges, and Advancements in Electromyography Signal Processing*, pages 321–353. IGI Global, 2014. 106
- [165] Stefano Pini, Andrea D’Eusanio, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Baracca: a multimodal dataset for anthropometric measurements in automotive. In *International Joint Conference on Biometrics (IJCB)*, 2020. 70
- [166] Stefano Pini, Guido Borghi, Roberto Vezzani, Davide Maltoni, and Rita Cucchiara. A systematic comparison of depth map representations for face recognition. *Sensors*, 21(3):944, 2021. 58, 59
- [167] Leonid Pishchulin, Stefanie Wuhler, Thomas Helten, Christian Theobalt, and Bernt Schiele. Building statistical shape spaces for 3d human modeling. *Pattern Recognition*, 67:276–286, 2017. 28
- [168] Stefanie TL Pöhlmann, Elaine F Harkness, Christopher J Taylor, and Susan M Astley. Evaluation of kinect 3d sensor for healthcare imaging. *Journal of medical and biological engineering*, 36(6):857–870, 2016. 53
- [169] Liliana Lo Presti and Marco La Cascia. 3d skeleton-based human action classification: A survey. *Pattern Recognition*, 2016. 92
- [170] Thomas Probst, Andrea Fossati, Mathieu Salzmann, and Luc Van Gool. Efficient model-free anthropometry from depth data. In *2017 International Conference on 3D Vision (3DV)*, pages 486–495. IEEE, 2017. 28
- [171] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 12, 20, 57, 62, 98
- [172] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 20, 57, 62
- [173] Arnaud Ramey, Víctor González-Pacheco, and Miguel A Salichs. Integration of a low-cost rgb-d sensor in a social robot for gesture

- recognition. In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 229–230. IEEE, 2011. 53
- [174] Fabio Remondino and David Stoppa. *TOF range-imaging cameras*, volume 68121. Springer, 2013. 8
- [175] Yu Ren and Chengcheng Gu. Hand gesture recognition based on hog characters and svm. *Bulletin of Science and Technology*, 2:011, 2011. 70
- [176] Danilo Jimenez Rezende, SM Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *Advances in Neural Information Processing Systems*, pages 5003–5011, 2016. 29
- [177] Stephan R Richter and Stefan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1936–1944, 2018. 29
- [178] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3577–3586, 2017. 21
- [179] Maik Riestock, Karl Fessel, Thomas Depner, and Hagen Borstell. Survey of depth cameras for process-integrated state detection in logistics. In *Smart SysTech 2019; European Conference on Smart Objects, Systems and Technologies*, pages 1–6. VDE, 2019. 8
- [180] Kathleen M Robinette, Hans Daanen, and Eric Paquet. The caesar project: a 3-d surface anthropometry survey. In *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No. PR00062)*, pages 380–386. IEEE, 1999. 27
- [181] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net: Localization-classification-regression for human pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 25
- [182] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 27

- [183] Elliot N Saba, Eric C Larson, and Shwetak N Patel. Dante vision: In-air and touch gesture sensing for natural surface interaction with combined depth and thermal cameras. In *2012 IEEE International Conference on Emerging Signal Processing Applications*, pages 167–170. IEEE, 2012. 70
- [184] Gaoli Sang, Jing Li, and Qijun Zhao. Pose-invariant face recognition via rgb-d images. *Computational Intelligence and Neuroscience*, 2016. 19
- [185] Hamed Sarbolandi, Damien Lefloch, and Andreas Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. *Computer Vision and Image Understanding*, 139:1–20, 2015. 8, 54, 59
- [186] Arman Savran, BüLent Sankur, and M Taha Bilge. Comparative evaluation of 3d vs. 2d modality for automatic detection of facial action units. *Pattern recognition*, 45(2):767–782, 2012. 58
- [187] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 18
- [188] Zhiyuan Shi and Tae-Kyun Kim. Learning and refining of privileged information-based rnns for action recognition from depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3461–3470, 2017. 11, 19
- [189] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304. IEEE, 2011. 11, 19, 26, 37, 46, 53, 85, 87, 88, 89, 90, 91, 92
- [190] Alessandro Simoni, Luca Bergamini, Andrea Palazzi, Simone Calderara, and Rita Cucchiara. Future urban scenes generation through vehicles synthesis. In *International Conference on Pattern Recognition (ICPR)*. IEEE, 2021. 47
- [191] Alessandro Simoni, Andrea D’Eusanio, Stefano Pini, Guido Borghi, and Roberto Vezzani. Improving car model classification through

- vehicle keypoint localization. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2021. 47
- [192] Alessandro Simoni, Stefano Pini, Roberto Vezzani, and Rita Cucchiara. Multi-category mesh reconstruction from image collections. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021. 119, 120, 124
- [193] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014. 78, 80
- [194] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 18, 53, 57, 61, 86
- [195] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402, 2005. 75
- [196] Olga Sorkine. Differential representations for mesh processing. In *Computer Graphics Forum*, 2006. 118
- [197] Jong-Chyi Su, Matheus Gadelha, Rui Wang, and Subhransu Maji. A deeper look at 3d shape classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 55
- [198] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019. 25, 42, 85, 92, 93, 99, 102, 104, 107, 112
- [199] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 47

- [200] Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015. 18
- [201] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 18, 53, 57, 61
- [202] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 18
- [203] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337, 2016. 29, 113
- [204] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004. 59
- [205] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 6411–6420, 2019. 20
- [206] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision*, pages 4489–4497, 2015. 22, 78
- [207] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 57, 62
- [208] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable

- ray consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2634, 2017. 29, 113, 121
- [209] Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2897–2905, 2018. 29
- [210] Shubham Tulsiani, Nilesh Kulkarni, and Abhinav Gupta. Implicit mesh reconstruction from unannotated image collections. *preprint arXiv:2007.08504*, 2020. 29, 47, 113, 121
- [211] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 1991. 18
- [212] Nevin Utqualp and Ilker Ercan. Anthropometric measurements usage in medical sciences. *BioMed research international*, 2015, 2015. 106
- [213] Julien Valentin, Adarsh Kowdle, Jonathan T Barron, Neal Wadhwa, Max Dzitsiuk, Michael Schoenberg, Vivek Verma, Ambrus Csaszar, Eric Turner, Ivan Dryanovski, et al. Depth from motion for smartphone ar. *ACM Transactions on Graphics*, 37(6):1–19, 2018. 53
- [214] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 85
- [215] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 22, 70, 73
- [216] Marco Venturelli, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Deep head pose estimation from depth data for in-car automotive applications. In *International Workshop on Understanding Human Activities through 3D Sensors*, pages 74–85, 2016. 85
- [217] Norman Villaroman, Dale Rowe, and Bret Swan. Teaching natural user interaction using openni and the microsoft kinect sensor. In *Proceedings of the 2011 conference on Information technology education*, pages 227–232, 2011. 53

- [218] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 114, 120
- [219] Chengde Wan, Angela Yao, and Luc Van Gool. Hand pose estimation from local surface normals. In *European Conference on Computer Vision*, pages 554–569. Springer, 2016. 74
- [220] Qingfu Wan, Weichao Qiu, and Alan L Yuille. Patch-based 3d human pose refinement. *arXiv preprint arXiv:1905.08231*, 2019. 27
- [221] Heng Wang, Dan Oneata, Jakob Verbeek, and Cordelia Schmid. A robust and efficient video representation for action recognition. *International Journal of Computer Vision*, 119(3):219–238, 2016. 78, 80
- [222] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *European Conference on Computer Vision*, pages 52–67, 2018. 29, 113, 118
- [223] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics*, 36(4):1–11, 2017. 21
- [224] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, 2019. 20
- [225] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 123
- [226] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 25, 85
- [227] Alexander Weiss, David Hirshberg, and Michael J Black. Home 3d body scans from noisy image and range data. In *IEEE International Conference on Computer Vision*, pages 1951–1958. IEEE, 2011. 27

- [228] Olivia Wiles and Andrew Zisserman. Silnet : Single- and multi-view reconstruction by learning from silhouettes. In *British Machine Vision Conference (BMVC)*, 2017. 29
- [229] Fernando A Wilson and Jim P Stimpson. Trends in fatalities from distracted driving in the united states, 1999 to 2008. *American journal of public health*, 100(11):2213–2219, 2010. 70
- [230] Chenxia Wu, Jiemi Zhang, Silvio Savarese, and Ashutosh Saxena. Watch-n-Patch: Unsupervised understanding of actions and relations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 26, 37, 46
- [231] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 121
- [232] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9621–9630, 2019. 20
- [233] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82, 2014. 47, 114, 120
- [234] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision*, 2018. 25, 85
- [235] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation. In *Conference on robot learning*, pages 1369–1378. PMLR, 2020. 11, 19
- [236] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *IEEE International Conference on Computer Vision*, 2019. 29, 113

- [237] Xinchun Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*, 2016. 29
- [238] Xiaodong Yang and YingLi Tian. Super normal vector for activity recognition using depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 804–811, 2014. 78
- [239] Xudong Yang, Di Huang, Yunhong Wang, and Liming Chen. Automatic 3d facial expression recognition using geometric scattering representation. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–6. IEEE, 2015. 20
- [240] Yi Yang, Jingkuan Song, Zi Huang, Zhigang Ma, Nicu Sebe, and Alexander G Hauptmann. Multi-feature fusion via hierarchical regression for multimedia analysis. *IEEE Transactions on Multimedia*, 15(3):572–581, 2012. 75
- [241] Ho Yub Jung, Soochahn Lee, Yong Seok Heo, and Il Dong Yun. Random tree walk toward instantaneous 3d human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 104
- [242] Pietro Zanuttigh and Ludovico Minto. Deep learning for 3d shape classification from multiple depth maps. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3615–3619, 2017. 59
- [243] Chenyang Zhang, Xiaodong Yang, and YingLi Tian. Histogram of 3d facets: A characteristic descriptor for hand gesture recognition. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, pages 1–8. IEEE, 2013. 74
- [244] Jing Zhang, Wanqing Li, Philip O Ogunbona, Pichao Wang, and Chang Tang. Rgb-d-based action recognition datasets: A survey. *Pattern Recognition*, 60:86–105, 2016. 11, 19
- [245] Jinjin Zhang, Di Huang, Yunhong Wang, and Jia Sun. Lock3dface: A large-scale database of low-cost kinect 3d faces. In *2016 International*

- Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2016. 44, 45, 55, 58, 59, 60, 61, 63, 64
- [246] Liang Zhang, Guangming Zhu, Lin Mei, Peiyi Shen, Syed Afaq Ali Shah, and Mohammed Bennamoun. Attention in convolutional lstm for gesture recognition. In *Advances in Neural Information Processing Systems*, pages 1953–1962, 2018. 70
- [247] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 119
- [248] Wenchao Zhang, Shiguang Shan, Wen Gao, Xilin Chen, and Hongming Zhang. Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition. In *IEEE International Conference on Computer Vision*. IEEE, 2005. 18
- [249] Xin Zhang and Ruiyuan Wu. Fast depth image denoising and enhancement using a deep convolutional network. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2499–2503. IEEE, 2016. 59
- [250] Zihao Zhang, Lei Hu, Xiaoming Deng, and Shihong Xia. Weakly supervised adversarial learning for 3d human pose estimation from point clouds. *IEEE transactions on visualization and computer graphics*, 2020. 27, 103, 104
- [251] Kai Zhao, Jingyi Xu, and Ming-Ming Cheng. Regularface: Deep face recognition via exclusive regularization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1136–1144, 2019. 18
- [252] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 92, 93
- [253] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 21

BIBLIOGRAPHY

- [254] Guangming Zhu, Liang Zhang, Peiyi Shen, and Juan Song. Multimodal gesture recognition using 3-d convolution and convolutional lstm. *IEEE Access*, 5:4517–4524, 2017. 70
- [255] Rui Zhu, Hamed Kiani Galoogahi, Chaoyang Wang, and Simon Lucey. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *IEEE International Conference on Computer Vision*, 2017. 29
- [256] Karel Zuiderveld. Contrast limited adaptive histogram equalization. *Graphics gems*, pages 474–485, 1994. 59