UNIVERSITÀ
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

DESIGN OF VIEWPOINT-EQUIVARIANT NETWORKS TO IMPROVE HUMAN POSE ESTIMATION

Nicola Garau

Advisor

Prof. Nicola Conci

Università degli Studi di Trento

April 2022

Acknowledgements

I would like to thank my advisor Nicola for his precious advice during my PhD program. He taught me many things that I never knew even existed and guided me well in my decisions. I hope to continue collaborating with him in the future! I thank the reviewers of my thesis for their considerations and all the improvement suggestions.

Cheers to all the past and present mmlabbers, from the ancient ones (Niccolò, Andrea and Michele) to the babies (Zeno, Giulia, Lorenzo, Ardan, Antonio...and of course Giuseppe the insider) and all the others in the middle of it. Thank you for all the good times in 228 and in 170!

Going back in time, cheers to all the Stiamo Malon people! Thank you for all the good years together and for all the champagnones. Thank you for being my big family in Trento. Going even further back in time, cheers to the Chwazi people! Thank you for being so nerd together and all the study sessions at Palla's house. Thank you for being my Tilliders and my big family in Cagliari. Going to pre-history, thank you Giulia for being my oldest and best friend. Thank you for always being there no matter what.

Of course, many thanks go to my real family in Cagliari. Thank you mom and dad, thank you Marti for being always so supportive and caring. I always think of you even if we are far away from each other. Many thanks to all of my grandparents for being my life models and inspiring me to do everything I like to do with passion. I'll always treasure your words of guidance.

Cheers to Frattaglia for always being a superstar and for all the licks and bites! (not so much for all the fur). Thanks to the Zanon family for being so caring!

And last but not least, thank you Annarita for being my favourite rubber ducky. Thank you for all the love and support during all these times, I could not ask for anything better. I cherish all the good moments together and hope to make them $e^{moments}$*!*

Abstract

Human pose estimation (HPE) is an ever-growing research field, with an increasing number of publications in the computer vision and deep learning fields and it covers a multitude of practical scenarios, from sports to entertainment and from surveillance to medical applications. Despite the impressive results that can be obtained with HPE, there are still many problems that need to be tackled when dealing with real-world applications. Most of the issues are linked to a poor or completely wrong detection of the pose that emerges from the inability of the network to model the viewpoint. This thesis shows how designing viewpoint-equivariant neural networks can lead to substantial improvements in the field of human pose estimation, both in terms of state-of-the-art results and better real-world applications. By jointly learning how to build hierarchical human body poses together with the observer viewpoint, a network can learn to generalise its predictions when dealing with previously unseen viewpoints. As a result, the amount of training data needed can be drastically reduced, simultaneously leading to faster and more efficient training and more robust and interpretable real-world applications.

Keywords

Human Pose Estimation, Viewpoint Equivariance, Computer Vision, Deep Learning, Capsule Networks

Contents

List of Tables

List of Figures

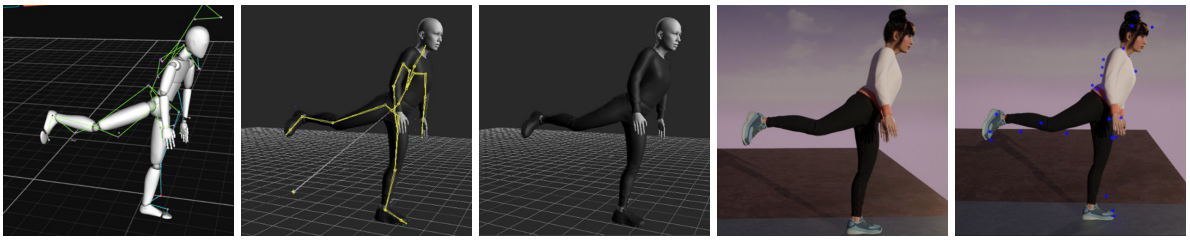
Chapter 1

Introduction

Human pose estimation (HPE) is the study of systems and methods for inferring and recovering the pose of articulated human bodies in space. There is strong evidence in literature that humans can parse visual information of articulated human bodies in specific category-selective regions in the brain [10]. Understanding how other people are posed in space allows us to infer more complex information about our surroundings. We are able to estimate actions and emotions, as well as effortlessly move in crowded spaces or predict other people motion (e.g. while playing group sports, in which anticipating the opponents movements is fundamental). More in general, every time we interact with other human beings we rely on our category-selective region of the brain to provide some high level information of how other people around us are moving.

Accordingly, endowing machines with similar human parsing capabilities has been an important focus in the latest research efforts, mainly in the computer vision and deep learning fields. Human pose estimation is traditionally carried out in different ways, as seen in Fig. 1.1.

As for the practical applications of HPE, they are numerous and span multiple areas of interests. Some of them are related to (i) robotics, to make embodied robots understand how to interact with people, (ii) assisted living and medical applications, to aid rehabilitation by correlating the human pose with articu-



(a) Motion capture (b) Skeleton pose estimation (c) 3D mesh retrieval (d) Virtual avatar posing (e) Markers estimation

Figure 1.1: Human pose estimation can be tackled in many different ways, depending on the task and on the available input data.

lation pain and muscular fatigue, (iii) sports applications, to monitor athletes' performances and fine-grained movements, (iv) pedestrian simulation and autonomous driving, to detect people with the focus of predicting their short and long time behaviour, (v) character animation and video games, to provide less intrusive alternatives to current state-of-the art motion capture systems. We will provide additional examples of these applications throughout the next chapters.

However, despite providing in theory a good solution for many applications, in practice human pose estimation still presents a loss of issues when applied to real world scenarios.

This thesis focuses on understanding why HPE does not usually meet the expectations of practical applications and how to improve it on different levels to make it more robust and well suited for the real world. More in detail, we focus on the aspect of viewpoint generalization. We show how neural networks are not able to intrinsically represent viewpoints if not explicitly given as a task. The missing viewpoint information often leads to poor generalization to unseen camera views and lack of robustness.

We first show how to embed a full perspective camera model in current human pose estimation pipelines, that can be used for calibrating camera networks with 3D human poses. Then we show how to implicitly represent the viewpoint information so that the network can learn it directly from data and not as an

additional task. We also show how to design biologically plausible networks inspired by the brain that can help solve many of the aforementioned issues. Finally, we provide some examples of how to deploy these enhanced human pose estimation networks in real-world ambient-assisted living applications.

More in detail, the main structure of the thesis can be subdivided in three macro-blocks.

In the first part (**Chapter2**), we explore how to design and improve HPE neural networks to solve real world problems, related to automatic camera calibration of dynamic networks of IP cameras (**Chapter 2**).

In the second part (**Chapter 3**) we go deeper into how neural networks for HPE work. We expose their biggest flaws and suggest new and improved network designs that allow to reach greater accuracy as well as far better generalisation to real-world scenarios and practical applications. We show how capsule networks and transformer-like architectures that take into account the viewpoint can infer part-whole hierarchies directly from data. This allows drastically improve both generalisation to unseen viewpoints and interpretability of the model predictions.

In the third part (**Chapter 4**) we show how all the improvements to both automatic camera calibration and HPE network architecture may be applied to highly dynamic environments such as medical applications and Ambient-Assisted Living (AAL). We present a preliminary implementation of both the technologies in a real AAL environment.

In **Chapter 5** we draw some conclusions on the thesis and discuss about some emerging topics that could help improve both the deep learning and the human pose estimation fields.

Chapter 2

Improving camera pose estimation through human pose estimation

Marker-less human pose estimation has seen a lot of improvements in recent literature. Despite marker-based solutions usually perform better with respect to joint-level accuracy, they present a few disadvantages when compared to markerless solutions. In the first place, they require specialized and expensive hardware and are subject to errors in non-controlled environments with reflective surfaces. Secondly, they must be precisely calibrated whenever the cameras configuration is changed. While calibrating a motion capture system is a well known and robust procedure, calibrating a network of cameras can be often troublesome. Camera calibration is a necessary preliminary step in computer vision for the estimation of the position of objects in the 3D world. Despite the intrinsic camera parameters can be easily computed offline, extrinsic parameters need to be computed each time a camera changes its position, thus not allowing for fast and dynamic network re-configuration.

In this chapter we present two frameworks for dynamically calibrating IP camera systems with an arbitrary numbers of cameras:

1. **[Section 2.2]** A fully markerless, unsupervised, and automatic tool for the estimation of the extrinsic parameters of a camera network, based on 3D human mesh recovery from RGB videos. We show how it is possible to

retrieve, from monocular images and with just a weak prior knowledge of the intrinsic parameters, the real-world position of the cameras in the network, together with the floor plane. Our solution also works with a single RGB camera and allows the user to dynamically add, re-position, or remove cameras from the network.

2. **[Section 2.3]** An extension to real-time applications which leverages optimised 3D human mesh recovery from a single image and a 3-stage parallel pipeline that enables fast inference and online dynamic re-calibration when needed.

2.1 HPE role in camera calibration

Before the advent of deep learning, classical HPE approaches were based on the so-called pictorial structures framework [7]. Later on, this kind of hand-crafted features, as well as customised hardware solutions (e.g., RGBD-based sensors) became less popular, making room for HPE algorithms based on deep learning paradigms.
Many human pose estimation techniques [21, 194] are based on bottom-up 2D skeleton estimation to guarantee good performances. Recent contributions [182] explore two-stage approaches, in which the 2D pose is first estimated and then used as a baseline to infer the corresponding 3D pose.

2.1.1 Bottom-up approaches

Estimating the human pose in a bottom-up fashion means first estimating all the joints in a frame and then linking them together in a meaningful, structured hierarchy. Cao et al.'s Realtime multi-person 2D pose estimation using part affinity fields [21] is one of the most popular multi-person real-time 2D pose estimation works in literature. It combines the architecture of a CNN-based variation of Pose Machines, called Convolutional Pose Machines [194], leveraging on part

affinity fields. Part affinity fields can be defined as a group of oriented vectors linking different joints. In other words, part affinity fields can be seen as confidence maps identifying bones, while joint confidence maps identify joints and articulations.
The solution, presented in [21], is very robust to large scale occlusions and self occlusions. Its dual-branch architecture for CNN-based joint parts and pairs estimation is optimised to run in real-time on consumer hardware, making it suitable for many research applications, and known as *OpenPose* [20]. However, it is still not faster than many top-down approaches when dealing with low density scenarios. Recently, it has been extended with a single track architecture [71], rendering it much faster than before, also embedding the hand and face joint information.

2.1.2 End-to-end solutions for 3D human pose estimation

End-to-end recovery of human shape and pose[87] is one of the most popular works in joint 3D human shape and pose. From a single RGB image of a person, the human pose θ and body shape β are regressed, together with camera scale s, rotation R and translation T.
An issue with this approach is that it is not suitable for run-time application and it is highly affected by viewpoint changes and flickering between frames, due to the lack of temporal coherence. Other works (see [99][89][143]) inspired by [87] try to solve the flickering issue by using temporal cues or predicting future poses.

2.1.3 Automatic calibration

Most of the automatic extrinsic calibration works in literature leverage on the so called Manhattan World Assumption [34], which assumes that the geometry typical of urban areas makes it easier to discover vanishing points from

images taken in those kind of environments. As an example, Zhang et al. in [208] propose a solution that exploits the geometry of solar panels to estimate orthogonal vanishing points. While such assumption is valid and works well in city scenarios, it does not generalize sufficiently, especially when indoor scenes are taken into consideration. Many methods in literature deal with the problem of finding the parameters of a single camera which is being plugged in to an existing and already calibrated camera network. Vasconcelos et al. in [189] exploit sets of pairwise correspondences among images in order to estimate the pose of the new camera. Despite the high deployability, this method only works when the extrinsic parameters of the other cameras are known. In literature, methods for self-calibration of pan-tilt [94] and tilt-zoom [155] cameras can also be found. Traditionally, such kind of self calibration problems are handled using geometrical constraints; however, with the growing popularity of deep learning, some approaches tried to solve the problem, as in Hold-Geoffroy et al.'s work [77], which aims at estimating pitch, roll and focal length of a single camera employing convolutional neural networks. Of particular interest from this viewpoint, some very recent works focus on embedding CNN capabilities directly into camera sensors. One of the first works to achieve such results is Bose et al.'s *A Camera That CNNs: Towards Embedded Neural Networks on Pixel Processor Arrays*, an interesting proposal that could open new possibilities for in-camera self-calibration. With the growing popularity of omnidirectional cameras, Miyata et al. in [131] show how to exploit their large field of view to anchor non-overlapping views. However, such solution are not employable in some scenarios, such as AAL, since occlusions may play an important role for the failure of the keypoints detectors. Augmented reality (AR) also played an important role for refreshing the field of camera calibration. Zhao et al. in [210] employ augmented reality markers placed directly on top of cameras in order to perform camera calibration. However, the method requires the usage of an additional dedicated camera just for the recognition of the AR markers.

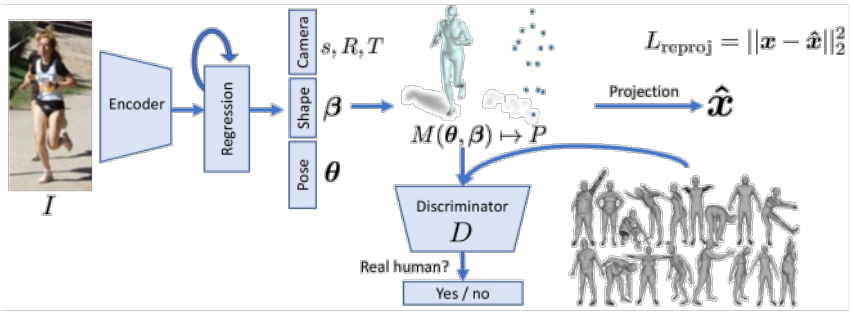


Figure 2.1: Joint pose, shape and camera estimation of *End-to-end recovery of human shape and pose* [87] pipeline.

Perhaps the most popular approaches for automatic camera calibration are the ones employing vanishing points estimation. Tang et al. in ESTHER [176] propose a complete pedestrian trajectory-based solution for joint intrinsic and extrinsic parameters estimation, especially focusing on intrinsic calibration for distortion correction. However, their method requires pedestrian to walk in a standard upright position, as well as a prior knowledge of the cameras vertical height.
To our best knowledge, few other works exploit human pose cues for camera calibration, and most of them exploit 3D sensors data, such as depth maps or cameras disparity information.

Desai et al. in [40] propose a skeleton-based method for semi-automatic continuous calibration of Kinect V2 sensors. In their work, they also explore some issues related to working with depth sensors, such as low range of vision, skeleton flipping and high computational costs. Among the many recent 3D human pose estimation works, some also try to jointly retrieve human pose and weak camera parameters. Kanazawa et al.'s approach [87] provides an estimation of the subject in terms of mesh, shape and pose representation, as well as some shallow cues of the camera pose.

2.2 Automatic camera pose estimation via HPE

A lot of effort has been put over the years into automating the process of camera resectioning. The available literature, though, still lacks of a fully unsupervised and markerless approach, which could significantly simplify the deployment of a camera network. Such deficiency in the state of the art might be perhaps due to the manifoldness of sensors and lenses on the market, which frustrates any generalization attempt. Another aspect that increases the difficulty of automatic calibration is the dynamic nature of the environments where the camera network is going to be installed. For example, in many scenarios, including video surveillance, Ambient Assisted Living (AAL), environmental monitoring, the reconfiguration of the camera network is generally common, often due to the re-positioning of pieces of furniture, or more in general obstacles that can partially or fully limit the visibility on the scene. In addition, especially when dealing with outdoor scenarios, cameras with PTZ (pan-tilt-zoom) capabilities are often adopted, as they are capable of changing their internal configuration, making it necessary to re-calibrate one or more sensors. Wind or other weather conditions may also be critical, introducing a lot of noise, and making it difficult to accomplish most of the classic vision tasks, such as keypoint extraction, motion detection, and tracking.

Many good solutions to estimate the intrinsic camera parameters to correct lens distortion have been provided in the past, the internal configuration of a camera remains usually fixed, except when zooming or refocusing. On the other hand, extrinsic parameters model the relation between the camera coordinates and the real-world coordinates. For this reason, even a slight movement of the camera can cause a loss in calibration precision. This is problematic because the standard calibration procedures, though not complex, are rather time consuming requiring every time manual intervention.

Many approaches try to simplify the calibration process by reducing the number

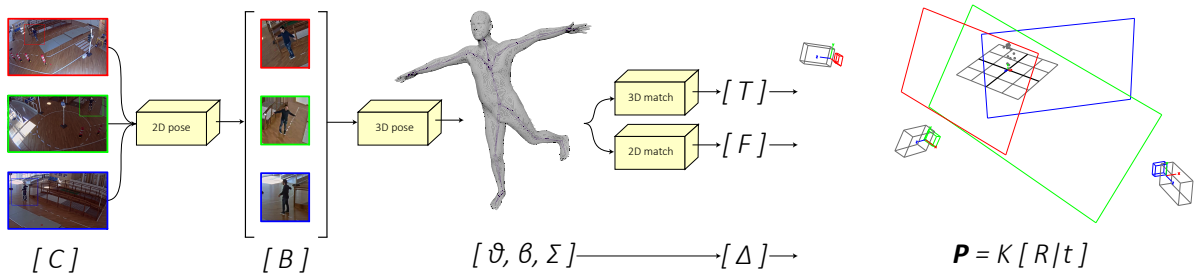


Figure 2.2: Illustration of the proposed pipeline: from RGB video streams to the estimation of the extrinsic parameters of the camera network.

of required checkerboard images. In autonomous driving, good markerless solutions can be found by using visual odometry [138], SLAM [46], or optical flow [119] for feature tracking; however, they are often not suitable for surveillance cameras, due to the completely different application scenario. Other methods use SIFT/SURF features and feature matching between camera views to estimate the camera parameters [81], but they require additional data from other sensors, such as active range sensors. Recent methods focus on calibrating cameras by using the cues provided by walking humans, both for intrinsic and extrinsic parameters estimation. In particular, these approaches are usually based on:

- Manhattan World Assumption
- Planar trajectories
- Skeleton data from 3D sensors

The ones based on the Manhattan World Assumption [34] usually work well in city environments due to their geometric uniformity, but may fail in all the other scenarios. Human detection and tracking has been used in the past to find vanishing points and estimate the ground plane from multiple camera views [176]. However, they require a prior knowledge of the people or camera height

and are not robust to occlusions, noise, or non-standard standing poses. Recently, RGBD sensors such as the Microsoft Kinect V2 [1] allowed to obtain a better understanding of the scene through depth and 3D skeleton pose estimation. However, the main issues in calibrating a camera network from the skeleton information via Kinect are the small range (¡3 m), low precision due to occlusions, and the high infrastructural and processing cost, as multiple computers are required to handle a network of RGBD sensors.
During the last years, researchers in computer vision have spent a lot of efforts in the area of human pose estimation from monocular images. There have been many successful examples, such as [148], [194] and [21]. Many of them have been made possible thanks to the availability of very large datasets, in particular from CMU's Panoptic Studio [84], leading to popular open source frameworks, such as OpenPose [20]. Among the different kinds of techniques used in the past to extract 3D human pose from monocular videos, we can distinguish between:

- two-stages approaches
- direct approaches

Two-stages approaches, such as [182], first estimate 2D joints and then recover the depth component. On the other hand, direct approaches try to recover the 3D skeleton or mesh in one shot. Among them, Kanazawa et al.'s *End-to-end recovery of human shape and pose* [87] is one of the most known.
Starting from Kanazawa's work, we extend it to work with multiple views and with a more realistic camera model, that allows for extrinsic parameters estimation. Our results show that, over a small number of frames, the human skeleton alone can provide a sufficient number of keypoints for fully unsupervised camera calibration, to retrieve the real-world 3D position of cameras in a camera

[1] https://developer.microsoft.com/en-us/windows/kinect

network. We show some results with different configurations and discuss on how our method can be further extended for better accuracy.

2.2.1 The proposed model

A simplified overview of our calibration architecture can be seen in figure 2.3. In short, the proposed method takes as input sequences of RGB frames from an arbitrary number of camera streams $C_{0,\dots,n}$, then applies 2D pose estimation to detect matching skeletons and the corresponding bounding boxes $B_{0,\dots,n}$. We then infer the 3D position of skeletal joints and their real-world scale through human mesh recovery. Finally, we align the skeletons centroids and use a least squares approach to find a set of rigid transformations $T_{\{i\to 0|i=1,\dots,n\}}$ from each skeleton to another in 3D world space. By minimizing the displacement error between skeletons in 3D space, we can exploit the epipolar geometry as well as the world-space and image-space position of joints to retrieve both the extrinsic parameters for rotation and translation $R \mid \mathbf{t}$ and the fundamental matrices $F_{\{i\to 0|i=1,\dots,n\}}$.
A big advantage of the model we propose is the possibility to re-calibrate the network when the distances between the detected 3D skeletons diverges. Another feature is the capability of continuously improving calibration over time, for example in case new cameras are added, thanks due to the proposed 3D matching approach.

In the following paragraphs we denote the camera matrix P, while the intrinsic and extrinsic parameters are modelled by K and $[R \mid \mathbf{t}]$ respectively.

$$P = \overbrace{\begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}}^{\text{K}} \times \overbrace{\left(\begin{array}{c|c} R & \mathbf{t} \\ \hline 0 & 1 \end{array}\right)}^{[\text{R} \mid \mathbf{t}]} \tag{2.1}$$

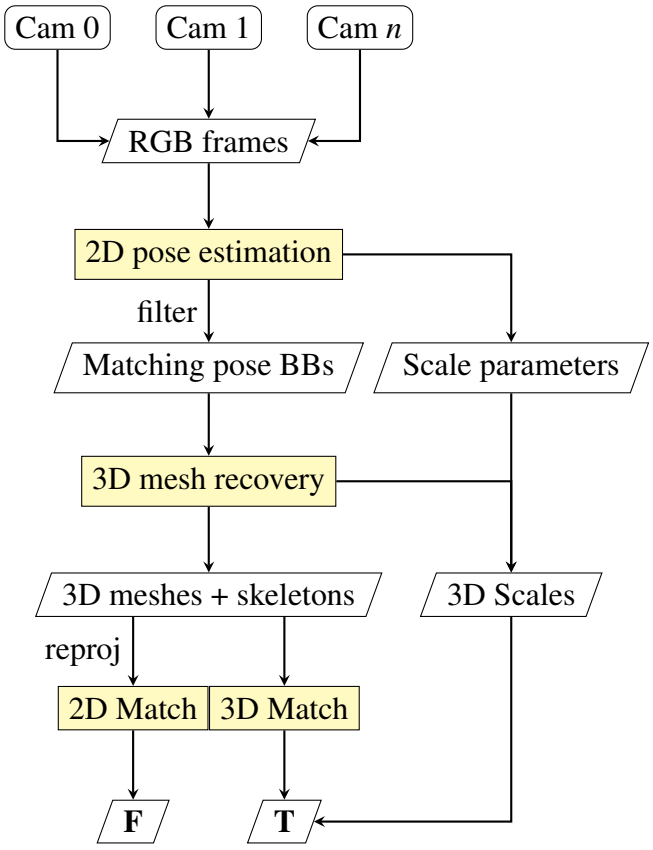


Figure 2.3: An overview of our camera network calibration pipeline

Figure 2.4: 2D pose estimation for bounding box extraction

2D pose matching

The first module of our architecture takes as input n sequences of RGB frames and produces as output n bounding boxes of the pedestrian with the overall highest detection confidence score among all the views and with matching skeletal configuration. To ensure a robust detection, we employ the method described in [21] for joint parts and pairs detection. In a single pedestrian scenario and by relaxing the occlusions or noise constraints, even the classic vision background subtraction methods or some refined versions such as [175] can be used.

Although the skeletal information can be already exploited at this point to estimate the fundamental matrices, our proposal computes the matrices after re-projecting the 3D-world skeleton onto the image planes for more accurate results. To do so, we keep a reference to the displacement of the central point $[D_x^{pix}, D_y^{pix}]$ and the pixel-size of each bounding box, as well as an unscale factor.

Note that in order to obtain the most accurate results possible, our framework accepts as input the prior knowledge of the people heights. However, approximate results can be obtained by providing an estimation of the known average height.

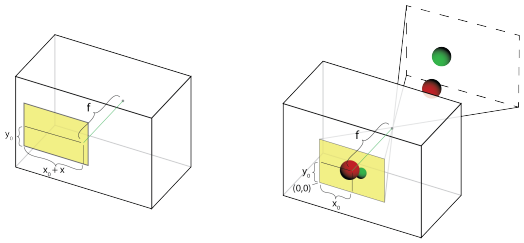


Figure 2.5: Principal point offset for mesh positioning in the weak perspective camera model does not correspond to a real-world mesh translation [162].

Mesh recovery

Each scaled bounding box $B_{0,\dots,n}$ contains now a crop of the original frames, containing the same pose of the subject, as seen from different viewpoints. Our goal is to jointly retrieve the 3D skeleton position from each viewpoint.
To achieve this, we employ a modified version of the method described in [87]. By feeding each bounding box B_i to the network, we obtain the vector Θ, corresponding to the SMPL [116] human body model parameters and configured as follows:

$$\Theta = [\overbrace{s, t_x, t_y}^{\text{camera}}, \overbrace{\theta}^{\text{pose}}, \overbrace{\beta}^{\text{shape}}] \tag{2.2}$$

From each human mesh $M(\theta,\beta)_i$ it is possible to obtain a set of $J = 19$ world-scale 3D joints ξ_i. The 10 body shape parameters β are used to refine the weak 2D pose matching described in 2.2.1 and remove remaining outliers. We discard the camera parameters s, t_x, t_y since they model a weak perspective pinhole camera model with its principal point shifted by $[t_x, t_y]$ (Fig. 2.5). In this model, the world translation of the mesh is computed as $z = F/s$, where s is a scaling factor.

The weak perspective model is not accurate enough for retrieving the real-world mesh transformation because it does not take into account perspective transformations. For this reason, we recover the real-world mesh displacement

Δ^{mm} in millimeters, based on the perspective camera model:

$$\Delta^{mm} = \left[\overbrace{\frac{\Delta_z^{mm}\Delta_x^{pix}}{f_x^{pix}}}^{\Delta_x^{mm}} \quad , \quad \overbrace{\frac{\Delta_z^{mm}\Delta_y^{pix}}{f_y^{pix}}}^{\Delta_y^{mm}} \quad , \quad \overbrace{\frac{f^{mm}wB^{mm}}{WB^{pix}}}^{\Delta_z^{mm}} \right] \tag{2.3}$$

where $f^{pix} = [f_x^{pix}, f_y^{pix}]$ corresponds to the focal length values in pixels, w is the image width in pixels and W is the sensor width in millimeters. B^{pix} and B^{mm} are the image-coordinates and world-coordinates sizes of the bounding boxes retrieved from 2.2.1. Now Δ_i^{mm} contains the real-world relative translation from C_i to ξ_i.

Skeleton matching

As of now, we built n camera-centric systems with n different 3D skeletons. We need to find the rotation matrices that map each skeleton $\xi_{1,\dots,n}$ to ξ_0. We achieve this by considering a skeleton-centric system, in which each skeleton centroid c is placed in the world center. In this space, we can find the relative skeleton-to-skeleton transformations in terms of rotation and translation by a single value decomposition (SVD) of H, as explained in equations 2.4 and 2.5.

$$H = \xi_0 \cdot \xi_i \quad , \quad U, S, V^\intercal = SVD(H) \tag{2.4}$$

$$R = V \cdot U^\intercal \quad , \quad \mathbf{t} = c_i - (R \cdot c_0^\intercal) \tag{2.5}$$

By starting from the skeleton-centric space, we apply the transformations of eq. 2.5 to ξ_i. Then, we move back to the camera-centric space and find the transformation that maps ξ_0 to Δ_0^{mm}. We finally find the inverse transformations Δ_i^{mm}, starting from eq. 2.3.

By applying this procedure, we obtain a 3D space in which C_0 is placed in

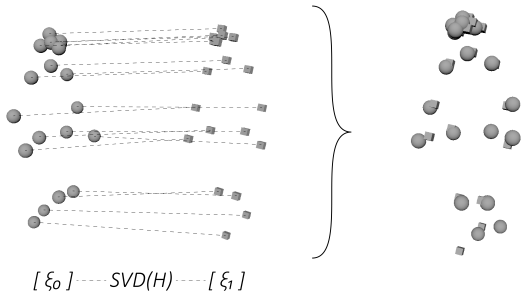


Figure 2.6: 3D skeleton matching for global transformation

the center of the coordinate system, the n skeletons are in Δ_0^{mm} and the relative position of all the other virtual cameras is known. A simple representation of the 3D skeleton match can be seen in Figure 2.6.

Fundamental matrix

With the skeletons $\xi_{0,\dots,i}$ correctly positioned in the 3D world, we calculate Σ as the 3D skeleton containing the mean values of all the joints in world-space coordinates. Since we also know the position of each camera in the 3D world, we can project Σ to each image plane of C_i, obtaining σ_i. We then build a vector σ_i containing 2D skeleton joints values for a batch of frames coming from C_i and use it as ordered keypoints to find the fundamental matrices $F_{i\to 0}$ that match camera C_i with camera C_0.
This allows us to find the epipolar lines and corresponding matching points between two camera views. Moreover, since the extrinsic matrices have been previously retrieved, it is possible to describe how points in world coordinates map to each camera coordinate system, and viceversa.

2.2.2 Results

We conducted three real-world experiments to test our framework in three different scenarios with different configurations, as listed in Table 2.1. The first

experiment is similar to the setup provided in [40], even if it has been carried out at longer distances and with regular IP cameras instead of RGBD sensors. The second experiment considers the possibility of using videos from two arbitrary smartphones, to show how our method can also simultaneously work with two totally different sensors. The third scenario has been carried out inside a small indoor space, with two IP cameras facing one each other and where the views suffer partial occlusions due to the environment configuration. It is worth noting that our method also works with a single camera, since the depth is being estimated directly from the 3D skeleton extracted from each individual frame.

Quantitative results

The main results of our experiments are listed in Table 2.1 and Table 2.2. As can be seen, they are comparable with the results provided by [40], particularly taking into consideration that we only employ monocular cameras and no additional depth sensors. The metrics MinSDE, ASDE and MaxSDE, describe the minimum, average, and maximum displacement of skeletal joints, respectively, after the matching in 3D space, in meters, calculated by the Euclidean distance:

$$SDE = \sqrt{\sum_{i=1}^{n} \left(\xi_0 - \xi_i \right)} \tag{2.6}$$

RPD, VPD (real and virtual plane displacements) are the measures of the displacement from the origin along the real world plane and the virtual world plane respectively. The RPD has been calculated starting from ground truth annotations, while the VPD can be calculated once again with an Euclidean distance from the origin, discarding the z component. The plane displacement error (PDE) is computed as $\mid RPD - VPD \mid$, once again in meters. The MRE is the mean reprojection error calculated after applying the fundamental matrix F to the set of points σ_i.

Our results for scenario 1 and 2 are also better than the checkerboard results obtained by [40] using the method described in [209].

	Experiments							
	Gym			Laboratory		Apartment		
Configuration								
Cameras	3			2		2		
Frames	250			250		875		
Sensor size	1/3” 9.1 mm			1/3.2” 5.7 mm	1/2.8” 6.3 mm	1/2.8” 7 mm		
Focal length	3.0	3.0	5.5	4.0	4.07	4.5	6	
3D Matching								
MinSDE	0.04			**0.01**		0.10		
ASDE	0.08			**0.06**		0.13		
MaxSDE	0.14			**0.08**		0.16		
Real-world displacement								
RPD	7.73			5.66		3.26		
VPD	7.92			5.79		3.35		
PDE	0.21			0.13		**0.09**		

Table 2.1: Experimental results. Columns: the three different test scenarios. Rows: number of cameras, number of frames, 3D skeleton displacement error (min, average, max), displacement along the plane (real plane, virtual plane, plane displacement error)

Reprojection error

After finding the fundamental matrices F for each scene and the corresponding epipolar lines, we assess the precision of our method by calculating the reprojection error in term of point-line-distance, as follows:

$$\frac{\mid ax_0 + bx_0 + c \mid}{\sqrt{a^2 + b^2}} \tag{2.7}$$

where a, b and c are the epipolar lines coefficients and $[x_0, y_0]$ are the coordinates of the projected points. In table 2.2 the reprojection errors in pixels for each FullHD scene are listed, showing that the proposed method is robust in all

the three test scenarios considered.

	Experiments		
	Gym	Laboratory	Apartment
MRE (pix)	**6.15**	12.03	8.38

Table 2.2: Reprojection errors for the three test environments

Qualitative results

In Figures 2.7, 2.8, 2.9 we provide some qualitative results through the Autodesk Maya® 3D animation, modeling, simulation, and rendering software. In each image a reconstruction of the 3D scene is shown, including the 3D skeleton used for the matching, the virtual plane and every virtual camera with correct roll, pitch, yaw, translation, focal length and frustum size. We decided to discard the approximate differentiable render OpenDR [117] used by [116] and [87] in favour of Maya because the latter lets us configure in fine details many camera parameters such as the focal length, the film gate and frustum size in millimeters. Moreover, our entire code can directly run into the Maya environment, allowing us to easily extend the scope of our work to weak monocular 3D human motion capture from video footage, also from a single camera. Our 3D reconstruction module in Maya is standalone and can receive camera and skeleton data from an external machine via a command port socket.

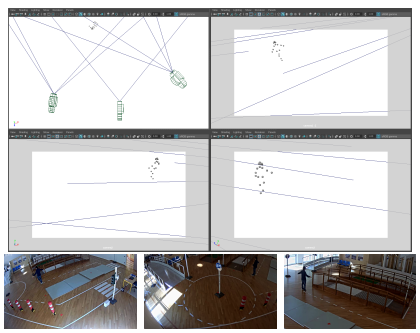


Figure 2.7:
Gym scenario

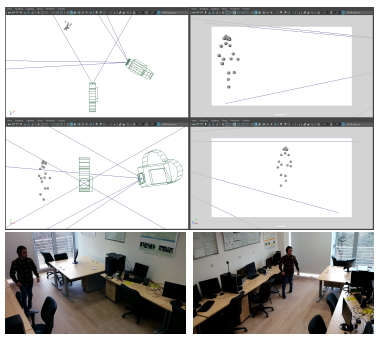


Figure 2.8:
Laboratory scenario

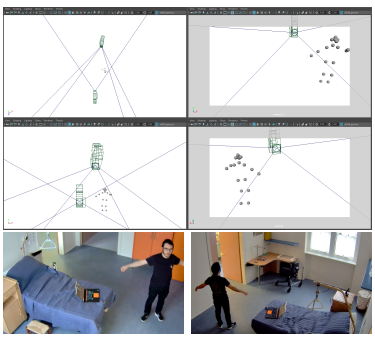


Figure 2.9:
Apartment scenario

2.2.3 Discussion

We presented a framework for a fully unsupervised markerless multi-camera calibration through monocular 3D human pose estimation. It consists of a 3-stage approach which employs (i) 2D human detection and matching between views, (ii) monocular 3D pose estimation through mesh recovery and (iii) joint 2D and 3D skeleton matching in camera-centric and skeleton-centric spaces. The overall output of our work consists of the extrinsic parameters that link cameras coordinates with world coordinates, as well as all the fundamental matrices to link the camera views. Compared to the other related works in literature, our approach works with at least one monocular RGB sensor and it is hardware-independent. It is robust to occlusions and noise in the scene thanks to the skeleton matching approach, and it is able to re-calibrate the camera network in real-time. Additionally, its accuracy can be refined over time in a continuous fashion.

2.3 Extension to real-time applications

In computer vision and 3D reconstruction, many works over the years have tried to automate the process of camera resectioning and calibration. Having the possibility to minimise the manual intervention within the calibration pipeline could simplify its deployment in many contexts and in a significant way. However, there is still a lack for fully unsupervised and markerless approaches for camera calibration in literature. The manifoldness of camera sensors and lenses present in the market hinders any generalization attempt. Another aspect that plays an important role in increasing the difficulty of automatic calibration is the dynamic nature of the environments, in which camera networks are generally being installed. For example, in many scenarios, including video surveillance, Ambient Assisted Living (AAL) and environmental monitoring, the reconfiguration and consequent re-calibration of the camera network is a common pro-

cess, often due to the re-positioning or addition of pieces of furniture, or, more in general, the presence of obstacles that can partially or fully limit the visibility of the observed environment. In addition, cameras with pan-tilt-zoom (PTZ) capabilities are often used. A big issue linked to the usage of PTZ cameras is that they are capable of changing their internal configuration, making it necessary to re-calibrate the whole network whenever these changes occur. In addition, wind or other weather conditions may also further complicate the scenario, introducing noise, and making it difficult to accomplish even the simplest vision tasks, such as keypoints extraction, motion detection and tracking.

Generally speaking, the internal configuration of a camera usually remains fixed, unless when zooming, refocusing or changing the lens parameters. Many good solutions to estimate the intrinsic parameters of a camera have been provided in the literature, and they usually require the usage of a checkerboard or other calibration tools.

On the other hand, extrinsic parameters model the relation between the camera coordinates and the real-world coordinates. Ideally, extrinsic parameters remain unaltered if both the camera and world long-term steadiness can be guaranteed. For this reason, even a slight movement of the camera can cause a loss in calibration precision, leading to the need of re-calibrating the whole system. This is problematic because the standard calibration procedures, though not complex, are rather time consuming and require the usage of third party calibration instruments by an expert technician who needs to be on the spot to perform the task. Another issue is that whenever a calibration is in progress, the camera network remains busy and inoperable.

A few approaches in literature [57] try to simplify the calibration process by increasing the accuracy of the calibration pattern detectors, thus reducing the number of required checkerboard images. However, despite being fast, they still require manual intervention. For example, although in a different application scenarios, the adoption of markerless solutions has been explored in the

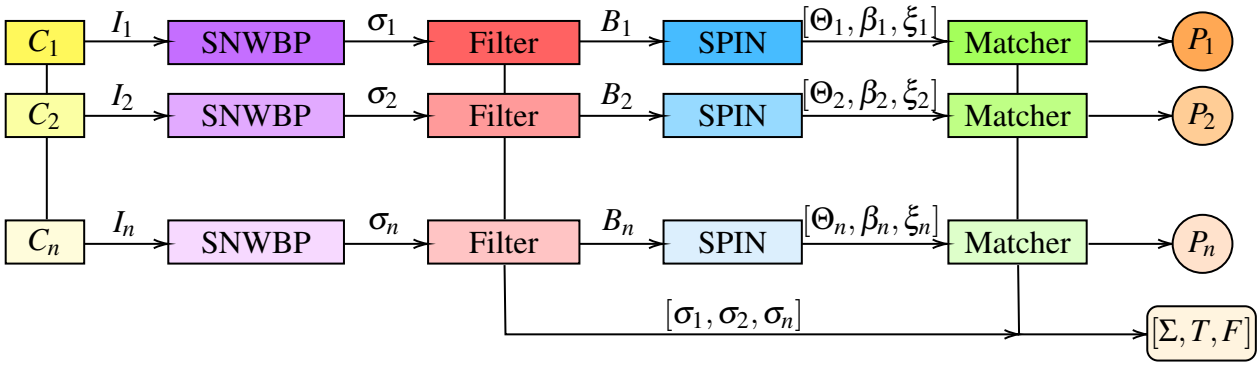


Figure 2.10: An overview of our camera network calibration pipeline. In the figure, SNWBP refers to a Single-Network Whole-Body Pose Estimation, the filters prepare the inputs for the SPIN human mesh recovery module. The matcher simply matches the obtained 3D skeletons in order to compute the output poses.

autonomous driving context, exploiting visual odometry [138], SLAM [46], and optical flow [119] for feature tracking; however, they are often not suitable for surveillance scenes, since such methods require a fixed camera configuration with respect to the vehicle, in order to exploit the car movement information for calibration. Other methods use SIFT/SURF feature matching between camera views to estimate the extrinsic camera parameters [81], but they usually require additional data from other sensors, such as active range sensors.

A recent trend in computer vision is pedestrian-based camera calibration, which focuses on finding how to estimate both intrinsic and extrinsic camera parameters by exploiting the cues provided by walking humans. In particular, these approaches are usually based on:

- Manhattan World Assumption
- Planar trajectories
- Skeleton data from 3D sensors

The approaches based on the Manhattan World Assumption [34] are usually adopted in city-like environments due to their geometric homogeneity, but may

fail in other scenarios, when no such geometric cues are being found. Human detection and tracking have been explored in literature as a support for vanishing point estimation and to estimate the ground plane from multiple camera views [176]. However, these methods often require a prior knowledge of the cameras' vertical position or of the people height, they are not robust to occlusions, noise, and can be fooled by unconventional human poses. Recently, RGBD sensors such as the Microsoft Kinect V2[2] and the Intel RealSense[3] allowed obtaining a better understanding of the scene through depth and 3D skeleton pose estimation [159][161][147]. However, there are many issues linked to the usage of RGBD sensors such as Kinect and RealSense to calibrate a camera network from the skeleton information. Among them, the most relevant ones are:

- Small range (usually $4m$) of the depth sensor; this constraint is not suitable for large environments.

- Low precision; occlusions, ambiguities and reflections in the scene are an important factor for the skeleton extraction precision.

- High infrastructural and processing cost; multiple computers and GPUs are usually required to process data coming from a network of RGBD sensors in real-time.

A recent trend in computer vision concerns the area of human pose estimation from monocular images. There have been many successful examples, such as [148], [194] and [21]. Many of the good results have been made possible thanks to the availability of very large datasets, in particular CMU's Panoptic Studio [84], which contributed to speed up the development of many popular and open source frameworks, such as OpenPose [20].

[2]https://developer.microsoft.com/en-us/windows/kinect

[3]https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html

Amongst the different kinds of 3D monocular human pose estimation techniques used in literature, we can distinguish between:

- two-stage approaches
- end-to-end approaches

Two-stage approaches, such as [182], first estimate 2D joints and then recover the depth component. On the other hand, end-to-end approaches try to recover the 3D skeleton or mesh in one shot. Kanazawa et al.'s work [87] is one of the most recent ones, which takes as input an image, encoding it into body pose, shape and weak camera pose via a CNN encoder; then, a discriminator is used as supervisor to encourage a better loss, by comparing the produced 3D model with a pool of real scanned 3D human poses (Fig. 2.1). Despite it being a very good approach to estimate the 3D mesh of a person, it may still fail, especially when dealing with unusual viewpoints and in time-constrained scenarios. Kolotouros et al. in SPIN (SMPL oPtimization IN the loop) [99] provide a fix to these issues by employing an hybrid top-down and bottom-up approach that aims at optimising the human mesh recovery (HMR) phase. Their method is based on the iterative application of optimisation and regression-based approaches (such as [87]) to further improve human mesh recovery, by mixing the advantages of both the approaches, in particular the accuracy of the first one with the speed of the second one.

Starting from Kanazawa's work, we extend it in a similar fashion as the one described in [99], and re-purpose in order to work with multiple views and with a more realistic camera model, that allows us to better estimate the extrinsic parameters for each camera. Our results show that, starting from a single frame, the retrieved human skeleton alone can provide a sufficient number of keypoints to estimate the real-world 3D position of cameras in a network, thus achieving fully unsupervised camera calibration. We show results in different scenarios and with different cameras configurations and discuss on how our method can

be further extended for better accuracy. This work is an extension of our previous work [54]. The main contribution, compared to the work in [54] consists of the capability of the system to obtain real-time camera network calibration, at comparable accuracy. This is achieved thanks to the adoption of a faster SNWBP network [71] and a more precise human mesh recovery pipeline [99]. These improvements allow for an even easier deployment in real-world scenarios, and are particularly helpful when dealing with large camera networks and real-time constraints.

2.3.1 Method overview

In this section, we refer to Fig. 2.11 to provide a red thread to explain our method's pipeline. During phase A, for each camera C_i in the network we acquire a single frame in a synchronous fashion. Then, each frame is forwarded to a Single-Network Whole-Body Pose Estimation (SNWBP) [71] network (phase B), which is a very fast convolutional network that is able to infer the 2D skeleton (σ) of multiple people inside the image in real-time. In this phase we also use the 2D skeleton information to compute a 2D bounding box B_i for each detected person in each frame. Then, during phase C, we use our joint human mesh recovery and camera pose estimation network, which is based on [99]. Starting from the monocular human mesh recovery network described in [99], we extend it by modifying the underlying camera model, providing a full perspective camera model. This addition, makes it possible, during phase D, to exploit the information acquired in the previous steps, such as the bounding boxes, 2D and 3D skeletons, body shape and pose parameters, to retrieve a good estimation of the camera pose for each camera in the network. All the four phases can run in parallel for each camera in the system, and in a continuous loop, in order to maximise both performance and precision by refining the calibration results over time.

2.3.2 The proposed model

In this section, we propose our one-shot method for fully automatic and unsupervised camera network calibration that leverages on monocular 3D human pose estimation from single images. In Figs. 2.11 and 2.10 we describe the pipeline and the different steps of our architecture. Looking at the bigger picture, our framework receives as input a single RGB frame $I_{0,\dots,n}$ from $n >= 1$ camera video streams $C_{0,\dots,n}$. We then apply fast, single network 2D pose estimation [71] for each frame, in order to filter matching subjects across frames and obtain the corresponding bounding boxes $B_{0,\dots,n}$ (Fig. 2.12). We then apply our custom optimised human mesh recovery method based on [87][99] to infer the 3D position of skeletal joints together with their real-world scale. Finally, when dealing with $n > 1$ cameras, we align the skeletons centroids and use a least squares approach to find a set of rigid transformations $T_{\{i\to 0|i=1,\dots,n\}}$ from each skeleton to another one in 3D world space. After minimizing the displacement error between skeletons in 3D space, we can exploit the epipolar geometry as well as the world-space and image-space position of joints to retrieve both the extrinsic parameters for rotation and translation $R \mid \mathbf{t}$ and the fundamental matrices $F_{\{i\to 0|i=1,\dots,n\}}$. In case of a single camera, the matching step and the fundamental matrix calculations are being ignored and we simply retrieve the camera matrix as well as the 3D human pose and shape.
Whenever the framework detects a difference in the detected 3D-space joints, which is bigger than a threshold, it triggers a new re-calibration cycle, in order to keep the network calibrated over time, progressively refining its accuracy. Another big advantage of our method is its flexibility. In fact, it can work even with a single camera and it allows for new cameras to be plugged into the system in a dynamic fashion.

In the next sections we refer to the camera matrix as P. The intrinsic matrix is defined by K, where $f_x = fm_x$ and $f_y = fm_y$ represent the focal length values

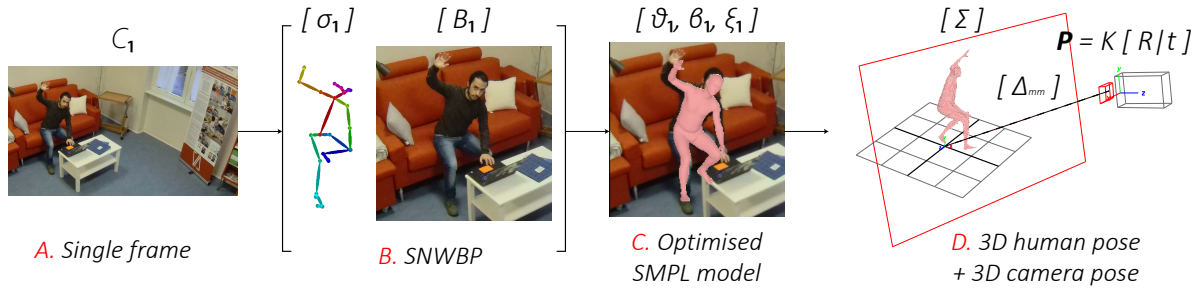


Figure 2.11: Illustration of the proposed pipeline: from a single RGB image to the estimation of the extrinsic parameters of the camera. In case of a camera network, the same pipeline is applied for each camera, before a 3D matching phase, as described in Fig. 2.10

in pixels, scaled along x and y by a scaling value m. The principal point of the camera is represented by (x_0, y_0). Extrinsic parameters are modelled by $[R \mid \mathbf{t}]$, where R is the rotation matrix and t identifies the translation vector.

$$P = \overbrace{\begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}}^{\mathrm{K}} \times \overbrace{\left(\begin{array}{c|c} R & \mathbf{t} \\ \hline 0 & 1 \end{array}\right)}^{[\mathrm{R} \,|\, \mathbf{t}]} \tag{2.8}$$

2D pose matching

As a first step, we provide a module that handles fast multi-person 2D pose estimation and filters detected skeletons to ensure good pose-based subject matches across the views. This first part of the architecture takes as input n RGB frames and outputs a bounding box for the target person in each image in terms of 2D pose, together with the overall highest detection confidence score among all the views. To ensure real-time performances, we employ an improved version of the method described by Cao in [21] for joint parts and pairs detection, namely Single-Network Whole-Body Pose Estimation (SNWBP) [71]. Alternatively, under particular conditions such as fixed, single-person, noise-free and occlusions-free scenarios, it is possible to employ classic background subtrac-

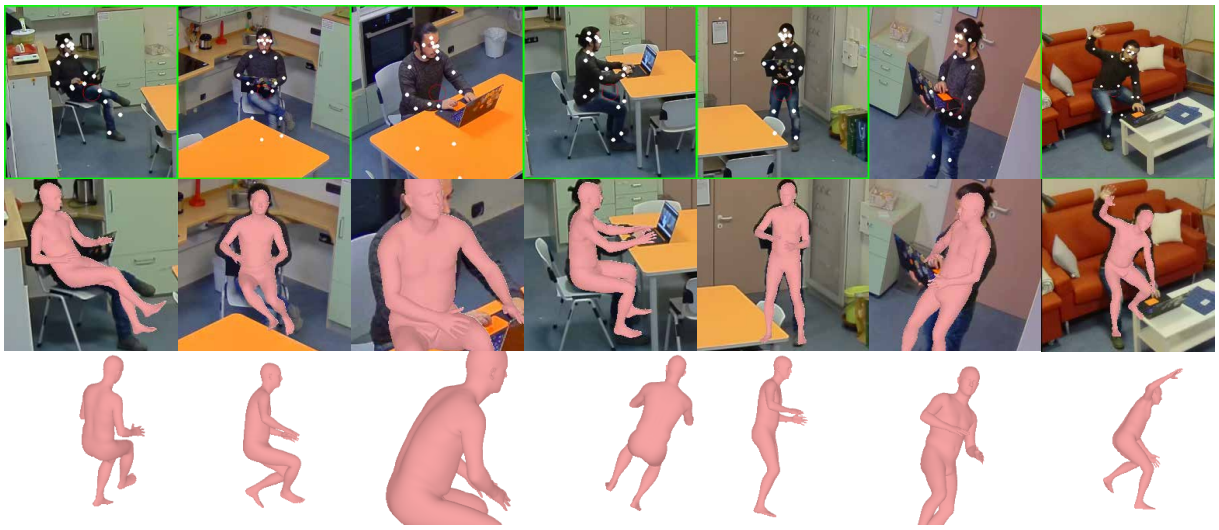


Figure 2.12: 2D pose estimation for bounding box extraction

tion methods or more advanced versions such as [175] to extract the bounding boxes.

At this point the skeleton joints information is already sufficient to calculate the fundamental matrices that link the views. However, we decided to further reinforce this estimation by providing additional points obtained from the re-projection of the 3D skeleton onto the image planes, as we will explain later on. By doing so, we observe an increment in the accuracy of the final fundamental matrices. Therefore, at this phase we only keep a reference to the displacement of the central point $[D_x^{pix}, D_y^{pix}]$ and the pixel-size of each bounding box, as well as an *unscale* factor, which serves as a parameter that can be used to reverse the scaling of the bounding boxes.
The 3D mesh recovery module based on [87] is able to retrieve an estimation of the person height, which can be used as a substitute to the real height. However, we provide as an option the possibility to give as additional input the real height of the considered subject in order to maximise the accuracy of the calibration.

Mesh recovery

Once we recovered the matching bounding boxes across all the different views together with the optional joint information, we need to recover 3D joints information that we will use to calculate the extrinsic parameters. At this point, each scaled bounding box $B_{0,\dots,n}$ is configured as a crop of the frames containing the subject chosen by 2D pose-similarity, as seen from different viewpoints. We now need to retrieve the 3D skeleton joints from each viewpoint, in a monocular fashion without relying on information from the other views.
To achieve this, we employ our modified version of the method described in [87] and [99] (SPIN). By feeding each bounding box B_i to the network, we obtain the vector Θ, corresponding to the SMPL (Skinned Multi-Person Linear Model) [116] human body model parameters, which is configured as follows:

$$\Theta = [\overbrace{s, t_x, t_y}^{\text{camera}}, \overbrace{\theta}^{\text{pose}}, \overbrace{\beta}^{\text{shape}}] \tag{2.9}$$

From each human mesh $M(\theta,\beta)_i$ it is possible to obtain a set of $J = 19$ world-scale 3D joints ξ_i (in meter coordinates). The 10 body shape parameters β encode different deformations of the mesh shape, and are used to refine the weak 2D pose matching described in 2.3.2 as well as removing remaining outliers. We discard the original camera parameters s, t_x, t_y since they model a weak perspective pinhole camera model with its principal point shifted by $[t_x, t_y]$ (Fig. 2.13). In the original model described in [87], the world translation of the mesh is computed as $z = F/s$, where s is a scaling factor.

The weak perspective model is not accurate enough for retrieving real-world mesh displacements because it does not take into account perspective transformations. In fact, in weak perspective geometry, perspective transformations are modeled via a simple scaling in the subject size, proportionally to its distance from the camera. In practice, if we take into account the manifold of commercially available sensors and lenses, employing a weak camera model is a strong

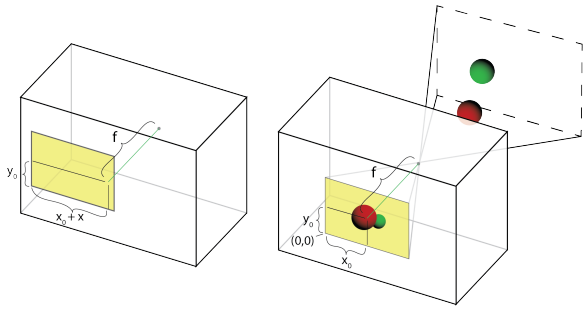


Figure 2.13: Principal point offset for mesh positioning in the weak perspective camera model does not correspond to a real-world mesh translation [162].

generalisation, which can lead to substantial errors. For this reason, we substitute the original weak camera model with a fully-fledged perspective one, to recover the real-world mesh displacement Δ^{mm} in millimeters, as shown in Eq. 2.10:

$$\Delta^{mm} = \left[\overbrace{\frac{\Delta_z^{mm}\Delta_x^{pix}}{f_x^{pix}}}^{\Delta_x^{mm}} , \overbrace{\frac{\Delta_z^{mm}\Delta_y^{pix}}{f_y^{pix}}}^{\Delta_y^{mm}} , \overbrace{\frac{f^{mm}wB^{mm}}{WB^{pix}}}^{\Delta_z^{mm}} \right] \tag{2.10}$$

where $f^{pix} = [f_x^{pix}, f_y^{pix}]$ corresponds to the focal length values in pixels, w is the image width in pixels and W is the sensor width in millimeters. B^{pix} and B^{mm} are the image-coordinates and world-coordinates sizes of the bounding boxes retrieved from 2.3.2. At this point, Δ_i^{mm} contains the real-world relative translation going from the camera C_i to the 3D skeleton ξ_i.

Skeleton matching

At this stage, in presence on an arbitrary number $n > 1$ cameras, we have obtained n camera-centric systems each one referring to a 3D skeleton. The next step is setting each skeleton's centroid c_i as the pivot point for each corresponding camera C_i. Thus, we need to find the rotation matrices R_i that map each skeleton $\xi_{1,\dots,n}$ to ξ_0. We achieve this by moving towards a skeleton-centric sys-

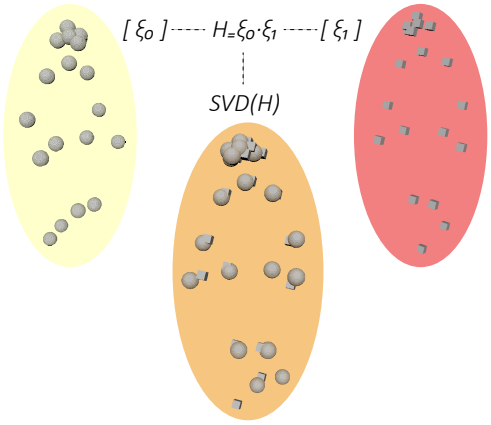


Figure 2.14: Our SVD approach for 3D skeleton matching

tem, in which each skeleton centroid c is positioned in the center of coordinates $(0,0,0)$. In this space, we can find the relative skeleton-to-skeleton transformations in terms of rotations and translations using a single value decomposition (SVD) approach, as explained in equations 2.11 and 2.12.

$$H = \xi_0 \cdot \xi_i \quad , \quad U,S,V^\top = SVD(H) \tag{2.11}$$

$$R = V \cdot U^\top \quad , \quad \mathbf{t} = c_i - (R \cdot c_0^\top) \tag{2.12}$$

More in detail, we calculate H as the dot product of a pair of 3D point sets of joints ξ_0 and ξ_i. We then apply an SVD to H to find the matrices U,S,V, as explained in Eq. 2.11. Finally, we find the rotation matrix R and the translation vector t as detailed in Eq. 2.12. A simple representation of the 3D skeleton match can be seen in Fig. 2.14.

Then, we move back to the camera-centric space and find the transformation that maps ξ_0 to Δ_0^{mm}. We finally find the inverse transformations Δ_i^{mm}, starting from Eq. 2.10.

By applying this procedure, we obtain a 3D space, in which the first camera C_0 is positioned at the center of the coordinate system, the n skeletons are in Δ_0^{mm} and the relative position of all the other virtual cameras is known. An example of the final output of the whole pipeline can be seen in Fig. 2.15.

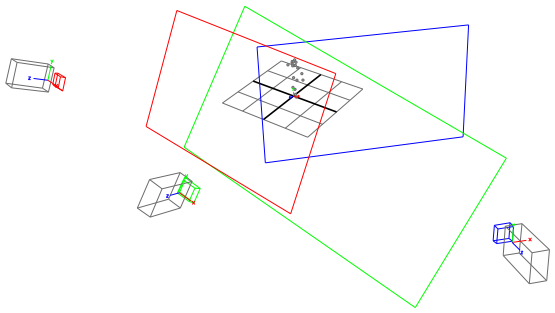


Figure 2.15: Visualisation of the final result of our automatic calibration pipeline.

Fundamental matrix

With the skeletons $\xi_{0,\dots,i}$ correctly positioned in the 3D world, we calculate Σ as the merged 3D skeleton containing the mean values of all the joints coming from $\xi_{0,\dots,i}$ in world-space coordinates. Since we also know the position of each camera in the 3D world, we can project Σ to each image plane of C_i, obtaining σ_i. We then build a vector σ_i containing 2D skeleton joints values for a batch of frames coming from C_i and use it as ordered keypoints to find the fundamental matrices $F_{i\to 0}$ that match camera C_i with camera C_0.
This allows us to find the epipolar lines and corresponding matching points between pairs of camera views. Moreover, since the extrinsic matrices have been previously retrieved, it is possible to describe, how points in world coordinates map to each camera coordinate system, and viceversa.

2.3.3 Results

To test our framework, we conducted seven real-world experiments in different scenarios, as listed in Table 2.4. The first four experiments were carried out in a real living lab consisting of three rooms, which is equipped with a network of identically-configured HD and FullHD cameras monitoring all the rooms. The last three experiments serve as a comparison of the proposed pipeline with our previous method, which employed video sequences instead of single frames, as well as slower and less precise human pose estimators. Our results are com-

parable with the ones provided by [40], both in terms of spatial configuration and precision, even if we rely on just monocular information from simple RGB cameras and not on depth or triangulation. Experiments 2,3,7 show how our method is robust against important occlusions in the scene. In experiment 5 we demonstrate how our method can also work with very distant and little overlapping cameras. In experiment 6 we employ two handheld smartphones (not stabilized) and successfully retrieve a good estimation of their pose in the 3D world.

Quantitative results

The main results of our experiments are listed in Table 2.3 and Table 2.4. As can be seen, they are comparable with the results provided by [40], particularly taking into consideration that we only employ monocular cameras and no additional depth sensors. The metrics MinSDE, ASDE and MaxSDE, describe the minimum, average, and maximum displacement of skeletal joints, respectively, after the matching in 3D space, in meters, calculated by the Euclidean distance:

$$SDE = \sqrt{\sum_{i=1}^{n} \left(\xi_0 - \xi_i \right)} \tag{2.13}$$

RPD, VPD (real and virtual plane displacements) are the measures of the displacement from the origin along the real world plane and the virtual world plane respectively. The RPD has been calculated starting from ground truth annotations, while the VPD can be calculated once again with an Euclidean distance from the origin, discarding the z component. The plane displacement error (PDE) is computed as $\mid RPD - VPD \mid$, once again in meters. The MRE is the mean reprojection error calculated after applying the fundamental matrix F to the set of points σ_i.
Our results for most of the scenarios are also better than the checkerboard results obtained by [40] using the method described in [209].

Reprojection error

After finding the fundamental matrices F for each scene and the corresponding epipolar lines, we assess the precision of our method by calculating the reprojection error in term of point-line-distance, as follows:

$$\frac{\mid ax_0 + bx_0 + c \mid}{\sqrt{a^2 + b^2}} \tag{2.14}$$

where a, b and c are the epipolar lines coefficients and $[x_0, y_0]$ are the coordinates of the projected points. In Table 2.3 the reprojection errors in pixels for each scenario are listed, showing that the proposed method is robust in all the four test environments considered.

	Scenarios			
	Kitchen	Gym	Laboratory	Apartment
MRE	12.77	**6.15**	12.03	8.38

Table 2.3: Reprojection errors in pixels for the four test scenarios

Qualitative results

In Figs. 2.16 and 2.17 we provide some qualitative results through the Autodesk Maya® 3D animation, modeling, simulation, and rendering software. In each image a reconstruction of the 3D scene is shown, including the 3D skeleton used for the matching, the virtual plane and every virtual camera with correct roll, pitch, yaw, translation, focal length and frustum size. We decided to discard the approximate differentiable render OpenDR [117] used by [116] and [87] in favour of Maya because the latter lets us configure in fine details many camera parameters including the focal length, the film gate and frustum size in millimeters. Moreover, our entire code can directly run into the Maya environment, allowing us to easily extend the scope of our work to weak monocular 3D human motion capture from video footage, also from a single camera. Our

3D reconstruction module in Maya is standalone and can receive camera and skeleton data from an external machine via a command port socket in real-time. As an alternative, we provide bindings for Open3D.

Concluding, despite the lack of proper datasets to benchmark these kind of applications, we also provide, in addition to the original experiments, some good qualitative results from the Panoptic Dataset [84] and from fully simulated scenarios. In Fig. 2.18 we show an example of camera pose estimation from 8 different views caught from 8 virtual cameras inside the Unity 3D environment. Similar results can be obtained for all the 480 VGA cameras in the Panoptic Dataset.

2.3.4 Discussion

We presented an extension of the automatic calibration framework described in 2.2, capable of calibrating a single camera or a camera network only from monocular human pose estimation cues in real-time, also enabling the possibility for continuous calibration. We employ a 3-stage approach which comprises (i) fast, single network whole body pose estimation and matching among camera views, (ii) perspective corrected, optimised monocular human mesh recovery from a single frame and (iii) joint 2D and 3D skeleton matching in camera-centric and skeleton-centric coordinates. Compared to the other related works in literature and with our previous approach, the presented framework enables the possibility for real-time, one-shot network calibration, which is camera-independent and which requires only one frame as input. It is robust to occlusions and noise in the scene thanks to the 3D skeleton matching approach, and it is able to perform real-time re-calibration thanks to its streamlined parallel architecture.

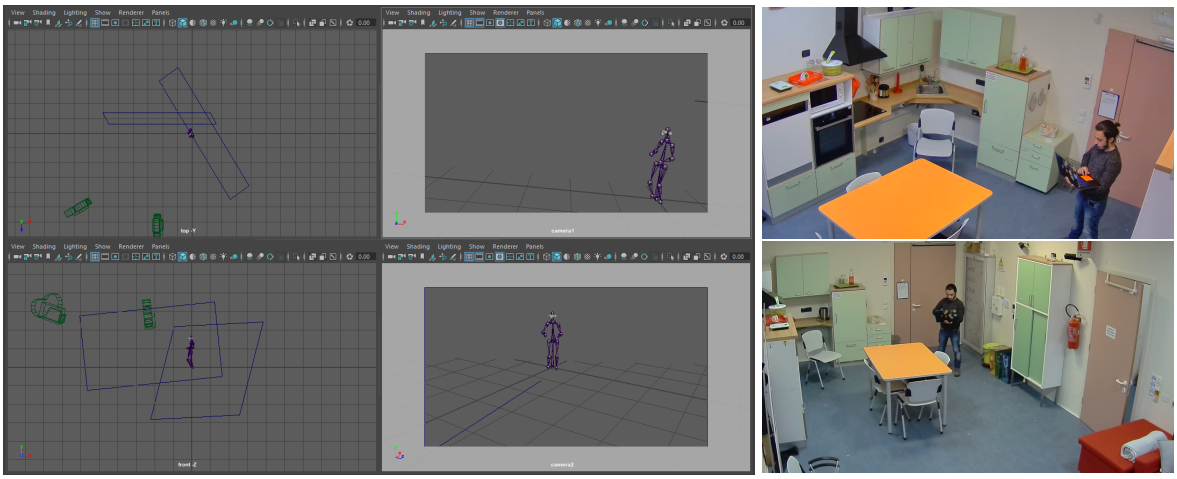


Figure 2.16: Kitchen scenario

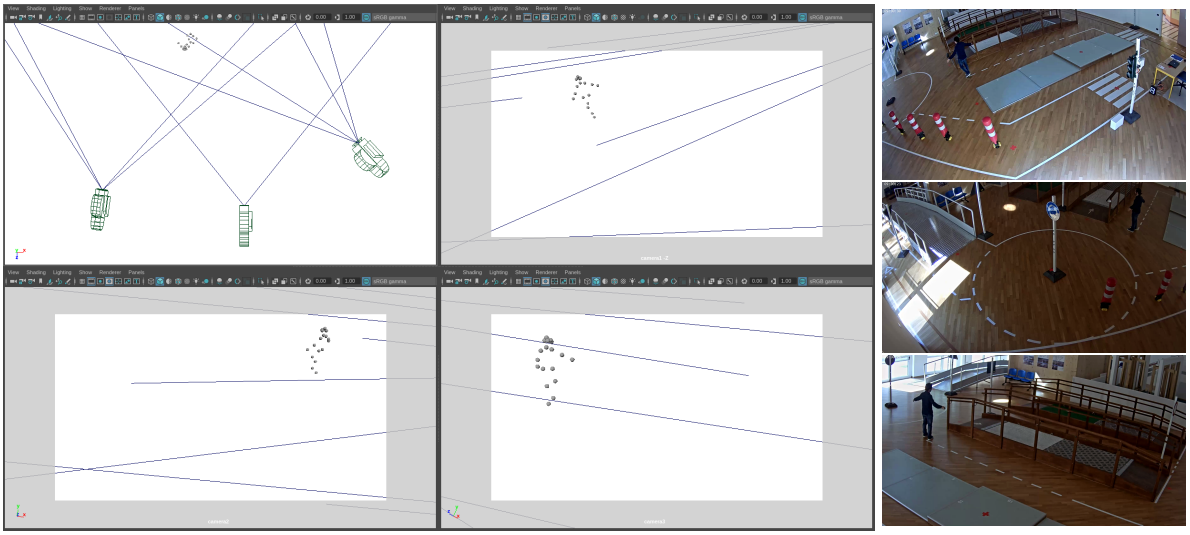


Figure 2.17: Gym scenario

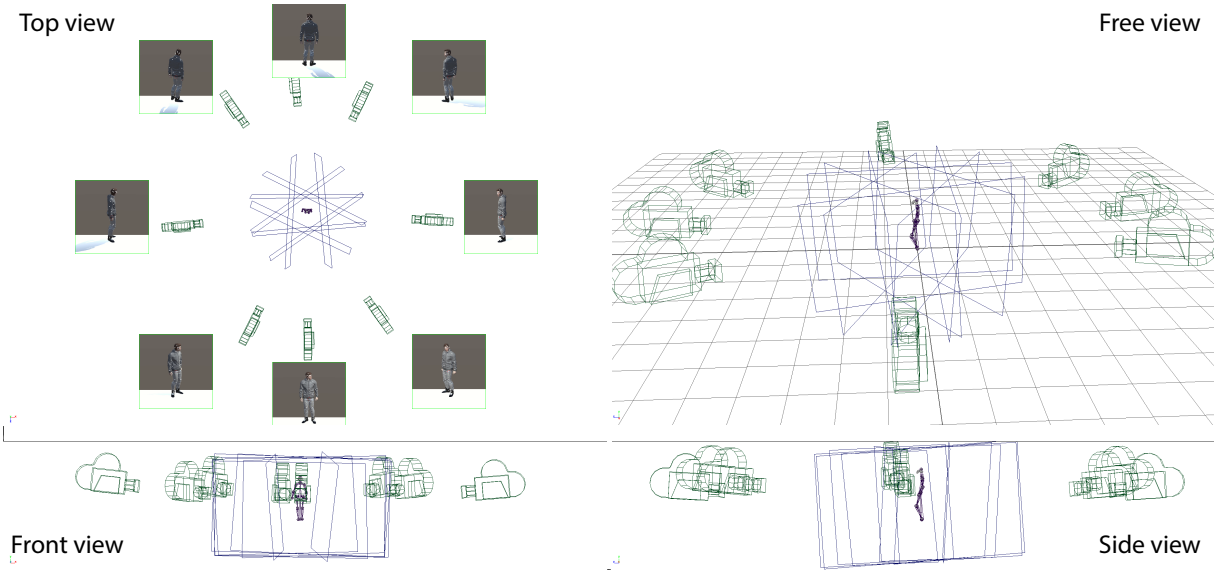


Figure 2.18: Simulated scenario: estimating the pose of 8 virtual cameras inside Unity

	Experiments													
	1 K.A		2 K.B		3 K.C		4 Liv.	5 Gym			6 Lab		7 Apart.	
Configuration														
Cams	2		2		2		1	3			2		2	
Frames	1		1		1		1	250			250		875	
Sensor	1/3” 6.28mm							1/3” 9.1mm			1/3.2” 5.7mm	1/2.8” 6.3mm	1/2.8” 7mm	
fl	4.0	5.0	4.0	5.0	4.0	5.0	5.0	3.0	3.0	5.5	4.0	4.07	4.5	6
3D matching														
SDE_{min}	3.34e-08		3.28e-08		**2.24e-08**		n.d.	0.04			0.01		0.10	
SDE_{avg}	**0.04**		0.07		0.04		n.d.	0.08			0.06		0.13	
SDE_{max}	**0.06**		0.10		0.08		n.d.	0.14			0.08		0.16	
Real-world displacement														
RPD	6.09		6.97		5.86		5.45	7.73			5.66		3.26	
VPD	6.46		7.04		6.24		5.50	7.92			5.79		3.35	
PDE	0.37		0.07		0.38		**0.05**	0.21			0.13		0.09	

Table 2.4: Experimental results. Columns: the seven different test scenarios (the last three are results from our previous method [54]). Rows: number of cameras, number of frames, sensor sizes, focal length, 3D skeleton displacement error (min, average, max), displacement along the plane (real plane, virtual plane, plane displacement error)

Chapter 3

Designing better human pose estimation networks

Deep neural networks achieve outstanding results in a large variety of tasks, often outperforming human experts. However, a known limitation of current neural architectures is the poor accessibility to understand and interpret the network response to a given input. This is directly related to the huge number of variables and the associated non-linearities of neural models, which are often used as black boxes. When it comes to critical applications as autonomous driving, security and safety, medicine and health, the lack of interpretability of the network behavior tends to induce skepticism and limited trustworthiness, despite the accurate performance of such systems in the given task. Furthermore, a single metric, such as the classification accuracy, provides a non-exhaustive evaluation of most real-world scenarios.

When dealing with Human Pose Estimation, this poor interpretability of the network responses is particularly relevant, often times leading to misdetection of joints or even entire poses. This happens mainly because current 3D HPE methods suffer a lack of viewpoint equivariance, namely they tend to fail or perform poorly when dealing with viewpoints unseen at training time. Deep learning methods often rely on either scale-invariant, translation-invariant, or rotation-invariant operations, such as max-pooling. However, the adoption of

such procedures does not necessarily improve viewpoint generalization, rather leading to more data-dependent methods. Moreover, existing datasets do not provide multiple-viewpoint observations and mostly focus on frontal views. Recently, capsule networks (CapsNets) have been introduced as a viable alternative to CNNs, ensuring viewpoint-equivariance and drastically reducing both the dataset size and the network complexity, while retaining high output accuracy. Introducing viewpoint-equivariance in HPE networks could be key to better interpretation of the network activations, leading to the emergence of spatially posed part-whole hierarchies directly from data.

In this chapter, we present four developments that tackle the problem of viewpoint-equivariance and interpretability in neural networks, in the following order:

1. **[Section 3.2]** We propose CapsulePose, a real-time end-to-end human pose estimation (HPE) network which employs the state-of-the-art matrix capsules [76] and a fast variational Bayesian capsule routing, without relying on pre-training, complex data augmentation or multiple datasets. We achieve comparable results to the HPE state-of-the-art, and the lowest error among methods using CapsNets, while at the same time achieving other desirable properties, namely greater generalization capabilities, stronger viewpoint equivariance and highly decreased data dependency, allowing for our network to be trained with only a fraction of the available datasets and without any data augmentation.

2. **[Section 3.3]** We introduce PanopTOP, a fully automatic framework for the generation of semi-synthetic RGB and depth samples with 2D and 3D ground truth of pedestrian poses from multiple arbitrary viewpoints. Starting from the Panoptic Dataset [85], we use the PanopTOP framework to generate the PanopTOP31K dataset, consisting of 31K images from 23 different subjects recorded from diverse and challenging viewpoints, also

including the top-view. Finally, we provide baseline results and cross-validation tests for our dataset, demonstrating how it is possible to generalize from the semi-synthetic to the real-world domain. The dataset and the code are publicly available.

3. **[Section 3.4]** We propose a novel capsule autoencoder network with fast Variational Bayes capsule routing, named DECA. By modeling each joint as a capsule entity, combined with the routing algorithm, our approach can preserve the joints' hierarchical and geometrical structure in the feature space, independently from the viewpoint. By achieving viewpoint equivariance, we drastically reduce the network data dependency at training time, resulting in an improved ability to generalize for unseen viewpoints. In the experimental validation, we outperform other methods on depth images from both seen and unseen viewpoints, both top-view, and front-view. In the RGB domain, the same network gives state-of-the-art results on the challenging viewpoint transfer task, also establishing a new framework for top-view HPE.

4. **[Section 3.5]** We introduce a possible step forward towards interpretability in neural networks, providing new tools to interpret their behavior. Starting from visual cues, humans can identify part-whole relationships between elements in a scene. If deep learning networks can internally identify such relationships, their representation is not interpretable by humans. Moreover, people are naturally able to analyze objects, delivering aggregate information such as their definition and organization in conceptual-semantic categories. Common deep learning networks can not perform such categorization, resulting in samples with close conceptual-semantic and lexical relations being far away from each other in the feature space. We present Agglomerator, a framework capable of providing a representation of part-whole hierarchies from visual cues and organizing the input dis-

tribution matching the conceptual-semantic hierarchical structure between classes. We evaluate our method on common datasets, such as Small-NORB, MNIST, FashionMNIST, CIFAR-10, and CIFAR-100, providing a more interpretable model than other state-of-the-art approaches.

We strongly believe that interpretability in neural networks is especially relevant now, because deep networks are already very capable at solving multiple tasks, but often lack robustness in real-world scenarios because of their black-box nature. Whenever a network fails in its task, it is important to understand why it does so, to unlock the possibility for better network designs and safer real-world applications.

3.1 The role of the viewpoint and capsules

In this section we will explore the existing literature for both capsule networks and 3D human pose estimation, finally exploring the evolution of neural architecture, from convolutional neural networks to capsules and transformer-like architectures.

3.1.1 Capsule networks

Capsule networks have been proposed in literature with the purpose of modeling a system capable of learning a part-whole relationships between so-called *entities* across different viewpoints, similarly to how our visual cortex system operates, according to the recognition-by-components theory [13]. This problem is also known as the viewpoint invariance problem, namely, how the network activations change with the change of the viewpoint, usually after a transformation (translation, scaling, rotation, shearing). CNNs' scalar activations are not suited to efficiently manage these kind of viewpoint transformations, thus needing to often rely to max-pooling and aggressive data augmentation. However, by do-

ing so, CNNs achieve viewpoint-invariance, meaning that a slight modification of the input image would lead to the same activation value. However, a more desirable property would be to capture and retain the transformation applied to the input image, in order for the network activations to be aware of the different transformations applied to the input. Being able to model network activations that change in a structured way according to the input viewpoint transformations is also called viewpoint-equivariance (Eq. 3.2), which is what CapsNets propose to achieve. In viewpoint-invariance (Eq. 3.1), a viewpoint transformation T does not change the outcome of the network activations. On the other hand, in viewpoint-equivariance (Eq. 3.2), the network activations change according to the applied viewpoint transformation T. Unlike traditional CNNs, which usually retain viewpoint-invariance (Eq. 3.1), capsule networks can explicitly model and jointly preserve a viewpoint transformation T through the network activations, drastically reducing the number of trainable parameters, depending on the application.

$$f(Tx) = f(x) \tag{3.1}$$

$$f(Tx) = Tf(x) \tag{3.2}$$

This is achieved by introducing the concept of *capsules*: groups of neurons, which explicitly encode the intrinsic viewpoint-invariant relationship that exists between different parts of the same object. As of today, three official capsule networks iterations have been presented [76, 102, 152]. The first one, by Sabour et al., introduces for the first time a routing algorithm for vector capsules (Fig. 3.1), called *routing-by-agreement* as a better max-pooling substitute, achieving very promising results on simple datasets such as MNIST, CIFAR10 and smallNORB.

The second official iteration of capsule networks [76], by Hinton et al., further improves accuracy, reducing the number of test errors on the smallNORB dataset by 45%. This is achieved through a more complex capsule structure

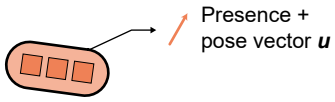


Figure 3.1: First iteration of a capsule's structure (vector capsules), suitable for dynamic-like routing, as described in [152]. Classic CNNs scalar-output feature detectors are replaced by vector-output capsules. Each capsule contains a single vector which describes both the pose and the presence of an entity.

(Fig. 3.2) and an Expectation-Maximization routing (*EM-routing*) for capsules. Unfortunately, the EM-routing and the 4×4 pose matrix embedded in the capsule contribute to increasing the training time, when compared to both CNNs and [152].

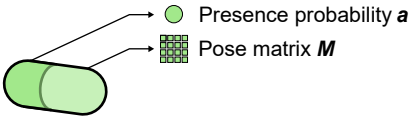


Figure 3.2: Second iteration of a capsule's structure (matrix capsules), suitable for expectation-maximization-like routing, as described in [76]. It contains both a scalar value and a 4X4 matrix, respectively describing the presence probability and a more robust 3D pose compared to the first capsule iteration.

The third official capsule network [102] by Kosiorek et al., constitutes a big leap forward, introducing for the first time an unsupervised capsule-based autoencoder. Ribeiro et al. in [150] build up on the EM-routing version of capsule by proposing for the first time a VB capsule routing for fitting a mixture of transforming Gaussians. They present state-of-the-art results on smallNORB by using $\sim 50\%$ less capsules, thus unlocking additional possibilities both in terms of performance gain and network complexity reduction. Recent works in literature explore the possibility of further reducing capsule network complexity through quaternions [140], while increasing performance as well. However, all the mentioned works only consider small datasets for benchmark.

3.1.2 Human pose estimation

Estimating the human pose from images can be seen as an ill-posed regression problem. Once model-based, learning-based human pose estimation (HPE) methods have recently gained a lot of interest in the research community, particularly real-time 2D HPE approaches [21], and only recently 3D HPE and human mesh recovery (HMR) approaches.
3D HPE usually relies on additional cues, such as 2D predictions [177, 182, 192], multiple images [214], pre-trained autoencoders [90] and pose dictionaries [153]. Other recent works aim at end-to-end, learning-based 3D HPE [114, 151, 179].
Bogo et al. in [16] considered the possibility of improving the human pose modeling via skinned multi-person linear models (SMPL), and in literature end-to-end 3D HMR approaches can be found [87, 99]. To our knowledge, only a recent work from Ramírez [149] tackles the problem of using capsule networks to solve the ill-posed 3D HPE problem in an end-to-end fashion. They propose a Bayesian formulation of the original version of the capsule network (with dynamic routing, as described in [152]) and benchmark it for the ill-posed problem of 3D HPE from single images, obtaining very promising results and showing how CapsNets can be used even with complex, big datasets.

The most common form of HPE consists of solving the task of estimating 2D joints and their connection, starting from RGB images and videos. However, HPE can be carried out even in different domains, such as depth images [47, 123, 203], LiDAR [52] or even radio signals [213]. Additionally, human pose estimation has recently shifted towards 3D estimation, by lifting 2D human poses to 3D [30, 157, 211], as well as with end-to-end approaches [171, 180]. Among the most recent developments of HPE, we can find human mesh recovery, which deals with the problem of retrieving the human pose from images or videos in terms of a fully rigged 3D mesh [96, 100]. Granularity also plays

an important role in HPE, with an increasing number of methods extending the pose to hand [80, 200], feet [202] or even face pose [83]. In recent years, research on human pose estimation has been focusing largely on single views, using either RGB [21, 68] or depth images [63, 133], as shown in Table 3.3.

3.1.3 Viewpoint-invariant HPE from RGB images.

In literature we can find two classes of approaches that extract human pose from RGB images. The bottom-up methods [19, 28, 136] detect firstly the human parts and then locate them in each object, and top-down methods [184, 195, 215] locate the key points in the human body and then compose the single parts into a person. Tekin *et al.* in [178] recover the 3D pose of people from consecutive frames of a video. They use at the same time appearance and motion information and regress directly from short sequences of frames to 3D poses in the central one. However, this method is limited to image sequences. Most recent methods exploit 2D pose estimation using CNNs [19, 137, 173, 195, 198]. The well-known work done by Cao *et al.*[19] detects the 2D pose of multiple people in an image. This approach associates body parts with individuals through the use of a parametric representation called *Part Affinity Fields*. In the context of multi-person pose estimation, one of the most recent works is the one proposed by Duan et al. [45] that implements a solution named location-sensitive network (LSNet) that unifies three recognition tasks like object detection, instance segmentation, and human pose estimation. The authors also present a novel loss function called *cross-IOU loss* that calculates the cross-IOU of each anchor-landmark to approximate the global IOU between prediction and ground-truth.

3D HPE usually leverages on additional cues, such as 2D predictions [177, 182, 192], multiple images [214], pre-trained models [90] and pose dictionaries [153]. Other recent works aim at end-to-end, learning-based 3D HPE [114, 151, 179]. In the RGB domain, common HPE datasets such as Human3.6M [82], provide images from multiple views, like front-view or side-view, while

the top-view component is generally missing. It is then evident that the lack of suitable multi-view (top-view in particular) data implies that state-of-the-art methods [21, 97, 99, 182] necessarily perform poorly when presented with an unseen viewpoint at test time, as shown in Fig. 3.20(a).

3.1.4 Viewpoint-invariant HPE from depth images.

Viewpoint-invariant HPE methods have been focusing exclusively on depth images [63, 133, 200] from top-view and side-view, and only a limited number of works address this problem. This might be due to the lack of datasets containing real or synthetic labeled depth data. Additionally, the majority of the depth-based datasets are small; this does not match the requirements of deep learning, which requires large amounts of data for proper training. In addition, they do not provide an accurate ground truth most of the time, rather automatic annotations, *i.e.* the position of the body joints is predicted using pose detectors such as [160]. The work by Shotton et al. [160] has been decisive in the human pose estimation from depth maps field, especially for its application in many successful commercial scenarios, such as the *Microsoft Kinect* and its SDKs. The authors propose a method for human pose estimation based on a *Random Forest* trained on a synthetic dataset (not publicly available), by classifying each pixel into body parts. The 3D position of joints is predicted from the labeled depth map with a local mode-finding approach based on Mean Shift. In very specific scenarios, such as strict front viewpoints, these methods obtain reasonable accuracy results and real-time performance, given the context of the application (gaming, interactive applications). In [70], Hernandez-Vela *et al.* propose an object segmentation framework using depth maps combining the use of Random Forest and Graph-cuts theory for the segmentation of human limbs in-depth maps. Firstly, Random Forest assigns a set of labeled probability for each depth sample belonging to a set of possible object labels. Then, with the use of Graph-cuts, the precedent procedure is optimized both locally, spa-

tially, and temporally. Ye *et al.* in [204] extract a point cloud from a depth map, and after the point cloud has been cleaned, transformed in frame coordinates, the body pose is predicted.

Viewpoint invariant HPE methods have been developed using depth images [63, 133, 200] from top-view and side-view, using datasets like the K2HPD Body Pose Dataset [193] and the ITOP dataset [63]. To take advantage of the 3D information encoded in 2D depth images, one recent research trend is to resort to 3D deep learning. The paid efforts can be generally categorized into 3D CNN-based and point-set-based families. To enhance the 3D proprieties of depth data and compute more significant features, current methods rely on 3D CNNs [63, 133] or 2D CNNs with dense features [200].

3D CNN-based methods [63, 133] perform a voxelization operation on pixels to transform them into 3D objects. To process the 3D data, each network performs costly 3D convolutions on the input data. These operations are responsible for the high computational burden and the difficulty to properly tune a high number of parameters in 3D CNNs. In the domain of 2D CNNs, Xiong et al. [200] capture the 3D structure by computing dense features in an ensemble way, thus avoiding computationally intensive CNN layers, but they still rely on a backbone pre-trained network to extract 2D features. Still, the above-mentioned approaches usually achieve weak viewpoint-invariance but fail to model viewpoint-equivariance. Moreover, we argue that the 3D geometry of the data should be interpreted by the network without relying on the voxelization embedding, or a 2D pre-trained feature extraction network.

3.1.5 Capsule networks for HPE.

Capsule networks have shown the ability to model the geometric nature of training data thanks to the network structure and features [76, 102, 152]. Sabour et al. introduce a routing algorithm for vector capsules, called *routing-by-agreement* as a better max-pooling substitute. Hinton et al. [76] further improve

accuracy through a more complex matrix capsule structure and an Expectation-Maximization routing (*EM-routing*) for capsules. Unfortunately, the EM-routing and the 4×4 pose matrix embedded in the capsule contribute to increasing the training time, when compared to both CNNs and vector CapsNets. Kosiorek et al. [102] introduce for the first time an unsupervised capsule-based autoencoder. Ribeiro et al. in [150] build upon the EM-routing version of capsule by proposing for the first time a Variational Bayes capsule routing (VB routing) fitting a mixture of transforming Gaussians. They present state-of-the-art results using $\sim 50\%$ fewer capsules, achieving both performance gain and network complexity reduction. However, all the mentioned works only consider small datasets, such as MNIST, smallNORB, and CIFAR-10 for benchmarking. In the RGB domain, Ramírez [149] tackles the problem of RGB HPE using dynamic vector capsule networks [152] to solve the 3D HPE problem in an end-to-end fashion. However, their work only exploits lateral viewpoints from the Human3.6M dataset and only considering RGB data.

3.1.6 The evolution of neural architectures

Convolutional Neural Networks (CNNs) [67, 163] have risen to a prominent role in computer vision when they started to outperform the existing literature in the image classification task of the ImageNet challenge [104]. The convolution operator can effectively describe spatially-correlated data resulting in a feature map, while the pooling operation down-samples the obtained feature map by summarizing the presence of certain features in patches of the image. The pooling operation in CNNs has been the subject of criticism since it does not preserve the information related to the part-whole relationship [165] between features belonging to the same object [152].

Transformers [44, 93, 115] have proven able to outperform CNNs, thanks to their ability to encode powerful features using self-attention and patch-based analysis of images. Multi-headed transformers [41] require the query, key, and

value weights to be trained differently for each head, which is a costly operation. The main advantage compared to CNNs is the ability of the multiple heads to combine information from different locations in the image with fewer losses than the pooling operation [108]. However, when compared with CNNs, Transformer-like models usually require intensive pre-training on large datasets, to achieve state-of-the-art performances.

Multi Layer Perceptrons (MLPs) [110, 181] are characterised by fully connected layers, in which each node is connected to every other possible node of the next layer. Even though they are easier to train and have simpler architecture compared to CNNs, the fully connected layers may cause the network to grow too fast in size and number of parameters, not allowing powerful scalability. MLPs have recently experienced a resurgence, thanks to patch-based approaches [110, 181], that allowed reaching state-of-the-art performances. They can also be seen as 1x1 convolutions [72, 110, 181], which do not require the pooling operation.

Capsules networks [76, 102, 124, 134, 150, 152] try to mimic the way the human brain creates a parse tree of parts and wholes by dynamically allocating groups of neurons (capsules) that can model objects at different levels of the part-whole hierarchy. The routing algorithm determines which capsules are activated to describe an object in the image, with lower-level capsules describing the parts (e.g. eyes and limbs), and higher-level capsules describing wholes (e.g. mammals and fish). While effectively routing information from different locations in the image, activated capsules cannot describe every single possible object in the image, thus limiting their effectiveness on more complex datasets (e.g. ImageNet, CIFAR-100), while achieving state-of-the-art results on simpler ones (e.g. MNIST).

There has been a recent push toward the so-called biologically inspired Artificial Intelligence (AI) [65, 78], which tries to build deep learning networks able to mimic the structure and functions of the human brain. In [65], the au-

thors propose a column-like structure, similar to hyper-columns typical of the human neocortex. In [186], the authors build upon cortical columns implemented as separate neural networks called Cortical Column Networks (CCN). Their framework aims at representing part-whole relationships in scenes to learn object-centric representations for classification.

The author in [72] proposes a conceptual framework, called GLOM, based on inter-connected columns, each of which is connected to a patch of the image and is composed of auto-encoders stacked in levels. Weights sharing among MLP-based [110] auto-encoders allows for an easily trainable architecture with fewer weights, while knowledge distillation [73] allows for a reduction of the training parameters. The patch-based approach combined with the spatial distribution of columns allows for a sort of positional encoding and viewpoint estimation similarly to what is used in *neural fields* [127, 165]. At training time, the author recommends that GLOM should be trained using a contrastive loss function [27]. This procedure, combined with a Transformer-like self-attention [190] mechanism on each layer of the columns, aims at reaching a consensus between columns. Routing the information with layer-based attention and stacked autoencoders would theoretically allow GLOM to learn a different level of abstraction of the input at a different location and level in the columns, creating a part-whole structure with a richer representation if compared to capsule networks [152].

3.2 CapsulePose

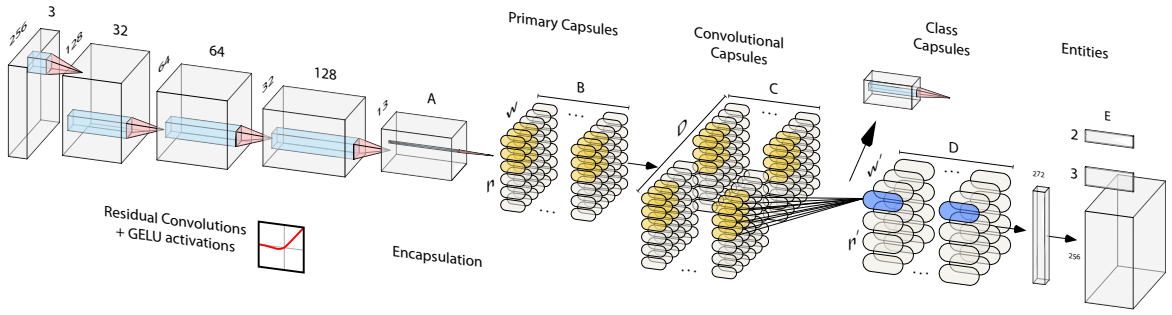


Figure 3.3: CapsulePose's network architecture: Residual convolutions, matrix capsules, multi-task decoding.

The literature on human pose estimation (HPE) has received an increasing number of contributions with applications to many different domains, including motion capture and tracking, augmented reality, robotics, sports analysis, activity recognition and ambient-assisted living, to name a few. Some crucial aspects that concern real-world applications are reliability, generalization, and real-time compliance of the employed architectures. However, it is known that, in order to maximize performance in many deep learning applications, including HPE, a huge amount of labeled data is required. Moreover, annotating 3D human pose data is not feasible without specific expensive equipment, such as 3D motion capture systems; since generalization greatly depends on the quantity and quality of data, HPE networks have grown in complexity, in proportion to the growth of the datasets. To further push the accuracy of the estimation, many methods take into consideration the usage of multiple datasets, which often rely on different joints labeling, non-standard data augmentation, biometric models and the usage of 2D ground truth during training or even testing. However, in real world applications, 2D ground truth data is not always available, and estimating 2D data to use it as a pseudo-ground truth is time consuming and leads to error propagation. For this reason, end-to-end methods are usually preferable and

more reliable than two-stage methods relying on 2D estimations. Nonetheless, still today, end-to-end 3D human pose estimation architectures are very complex and hard to train, especially when using a single dataset, and they usually rely on additional data and non-standard data augmentation in order to reach state-of-the art performance. Additionally, there are other factors to be considered when dealing with real-world applications, such as, for example, crowding and unconventional cameras positions or occlusions. These factors make generalization on unseen data harder, thus requiring more complex networks and increasing the risk of overfitting.

A novel learning system has been presented by Hinton et al. [76]. Capsule networks (CapsNets) address some of the issues of traditional convolutional neural networks (CNNs), such as poor information routing through *max pooling* and the limits of *scalar activation values* for generalization, by replacing them with *routing-by-agreement algorithms* and *capsule activation values*, respectively. CapNets have already shown excellent results on simple datasets, such as MNIST, smallNORB and CIFAR-10, proving to have superior generalization capabilities across unseen viewpoints and better interpretability, thanks to the embedded inverse graphics capabilities, while at the same time requiring significantly fewer parameters compared to traditional CNNs. All these qualities elicited a growing interest in the research community. However, despite their major advantages, CapsNets are not a popular choice when dealing with high resolution datasets, because of the longer training times of the capsule routing algorithms. The contribution of this paper is manifold: we propose a simple yet effective baseline network for end-to-end and real-time 3D human pose estimation. To our knowledge, it is the first architecture to employ state-of-the art matrix capsules with 4×4 3D pose matrices. We achieve state-of-the art performance on the Human3.6M dataset without employing additional data or non-standard data augmentation. We employ a Variational Bayes (VB) capsule routing paired with an optimized and modular codebase, to minimize both

training and testing times. We consider two of the most important novelties of this work to be the embedding of viewpoint-equivariance as frames of reference for greater generalization capabilities and the unprecedented usage of matrix capsules and VB routing for HPE. During testing, our network is almost twice as fast as other capsule-based architectures[1].

The paper is organized as follows: in Section 3.1 we propose an overview of the state-of-the-art for both capsule networks and HPE; in Section 3.2.1 we dissect our proposed architecture down to its core, commenting on the design choices and their reasons; finally, in Section 3.2.2, we discuss our experiments, showing quantitative and qualitative results, commenting on the improved generalization capabilities of the proposed architecture.

3.2.1 Proposed architecture

We present a novel capsule network architecture with Variational Bayes routing [150] for real-time end-to-end 3D human pose estimation from a single image. We aim at keeping a simple network structure, which can be summed up in three main blocks:

1. **Encoding**: residual convolutional encoder with GELU activations [69].
2. **Encapsulation**: matrix capsules (primary, convolutional, class capsules) with VB routing [150].
3. **Decoding**: custom DDGELU decoding with transposed convolutions and multi-task self-balanced loss.

For training and evaluation, we rely on the Human3.6M dataset [82], showing that capsule networks should be able to correctly learn 3D pose representations without the need of multiple datasets. As for the optimizer, we conducted

[1]The code, dataset and pre-trained models will be made available for fair comparison and replicability upon acceptance

some tests with both the Adam [95] and the improved AdamW [118] optimizer with decoupled weight decay regularization. We observed that the latter provides faster overall loss convergence and comparable performance, when training with a learning rate of $1e{-}10$, a weight decay value of $1e{-}2$ and a batch size of 32.
An overview of the proposed architecture can be seen in Fig. 3.3 and a more detailed layer-wise pipeline is shown in Algorithm 1.

Initialization

We start by normal-initializing all the loss register buffers s_{3D}, s_{2D}, s_H, s_W to 1 and every convolutional, capsule and dense weight w_c with values sampled from $\mathcal{U}(-\alpha, \alpha)$, where

$$\alpha = gain \times \sqrt{\frac{6}{n_i + n_{i+1}}} \tag{3.3}$$

according to the Xavier initialization [58]. This kind of initialization was demonstrated to achieve quicker convergence and higher accuracy on CIFAR10 [58].
We define the capsule pose matrix size $P = 4$ and assign capsule parameters $[A, B, C, D, E, F]$ the values $[64, 8, 16, 16, 17, 13]$. The number of input channels for the primary capsules is defined by A, while B, C, D are the numbers of output channels for primary, convolutional (first and second) capsules respectively; E defines the number of classes, which in our case is the number of joints. Finally, F defines the size of the feature space during encapsulation.

Encoding

For the first block, we accept as input $BS \times 3 \times 256 \times 256$ (BS=batch size) previously-cropped RGB images and sequentially apply 4 convolutional steps to obtain a $BS \times A \times F \times F$ feature space, which will be eventually converted into matrix capsules. Each convolutional layer is composed of residual convo-

Algorithm 1: CapsulePose network overview: from RGB images to 2D, 3D human poses and $E = 17$ joints heatmaps

CapsulePose (x)

inputs : A batch $x = x_0 \dots x_{BS}$, $\#BS$ = batch size of bounding-box RGB images containing a person

outputs: $\hat{y}_{3D} \in R^{BS \times E \times 3}$,
$\hat{y}_{2D} \in R^{BS \times E \times 2}$,
$\hat{y}_H \in R^{BS \times E \times 256 \times 256}$

$s_{3D}, s_{2D}, s_H, s_W \leftarrow 1$;

$w_c \leftarrow xavier_{uniform}() \quad \forall c \in ConvLayers$;

foreach $i \in ConvLayers$ **do**

  $x \leftarrow Conv2d_i(x) + Residual_i(x)$;

  $x \leftarrow \text{GELU}(x)$;

  $x \leftarrow InstanceNorm2d_i(x)$;

  $x \leftarrow Dropout_{0.3}(x)$;

$a, x \leftarrow PrimaryCapsules2d(x)$;

foreach $j \in ConvCapsuleLayers$ **do**

  $a, x \leftarrow ConvCapsules2d_j(a, x)$;

  $a, x \leftarrow VBRouting2d_j(a, x)$;

$a, x, \hat{y}_W \leftarrow ClassCapsules(a, x)$;

$a, x \leftarrow ClassRouting(a, x)$;

$x \leftarrow Entities(x)$;

$\hat{y}_{3D} \leftarrow tanh(\text{DDGELU}_{0.3}(\text{DDGELU}_{0.3}(x)))$;

$\hat{y}_{2D} \leftarrow sigm(\text{DDGELU}_{0.3}(\text{DDGELU}_{0.3}(x)))$;

$\hat{y}_H \leftarrow ReLU6(\text{DDGELU}_{0.3}(\text{DDGELU}_{0.3}(x)))$;

$\hat{y}_H \leftarrow ConvTranspose2d(reshape(\hat{y}_H))$;

return $[\hat{y}_{3D}, \hat{y}_{2D}, \hat{y}_H, \hat{y}_W]$;

lutions, to prevent vanishing gradients, as detailed in [67]. As for the activation function, we employ the Gaussian Error Linear Unit (GELU, Eq. 3.4) [69].

$$\begin{aligned} \text{GELU}(x) &= xP(X \leq x) \\ &= x\phi(x) \\ &\approx 0.5x(1+\tanh\sqrt{2/\pi}(x+0.044715x^3)) \end{aligned} \tag{3.4}$$

Figure 3.4: Gaussian Error Linear Unit (GELU) [69] activation function formula.

GELU activation functions have been used in many recent Transformer networks and have been proven to perform better in many learning tasks, including computer vision.

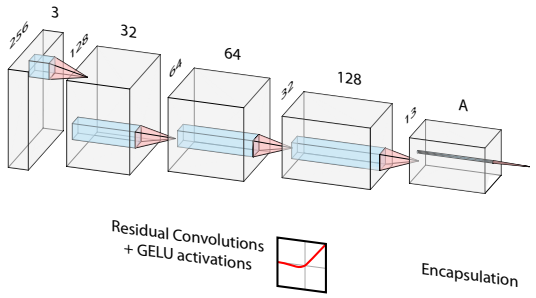


Figure 3.5: CNN encoder with residual convolutions and GELU [69] activations.

As for normalization, we adopted a combination of instance normalization followed by a dropout layer with rate 0.3. We chose instance normalization over batch normalization because the latter combined with dropout may lead to anomalous behaviour during training. On the other hand, dropout can serve as additional normalization, as well as a means to capture the model uncertainty, as detailed in [53]. Each convolutional layer has a kernel of size 9×9 and stride 2, except the last one which has stride 3 and additional padding.

Encapsulation

The $BS \times A \times F \times F$ output coming from the convolutional layers is now 'encapsulated' into B primary capsules, resulting in a shape of $BS \times B \times P \times P \times F \times F$. Each primary capsule is a matrix capsule (Fig. 3.2), and thus is composed of an activation value a_i and a pose matrix M_i. To our knowledge, this is the first work in literature that uses matrix capsules [76] to tackle the human pose estimation problem. A brief overview of the capsule layers is shown in Fig. 3.6.

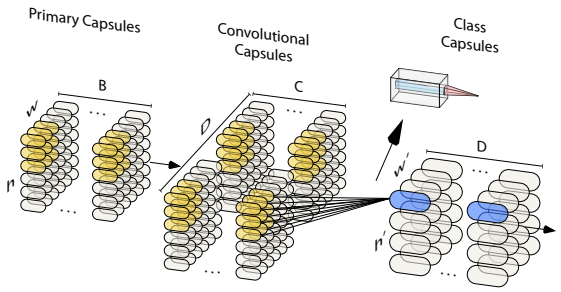


Figure 3.6: Capsules layers: primary, convolutional and class capsules.

Primary capsules are followed by 3 convolutional capsules (*ConvCaps*) layers, of which the last one is a class capsules (*ClassCaps*) layer. The first ConvCaps layer outputs capsules of shape $BS \times C \times P \times P \times 6 \times 6$, while the second has output $BS \times D \times P \times P \times 4 \times 4$. They both employ a 3×3 convolutional kernel, with strides 2 and 1 respectively. Finally, the ClassCaps layer gives as output E capsules of $BS \times E \times P \times P$ by sharing weights matrices W across spatial dimensions and using a kernel of size 1 with stride 1. For both convolutional and class capsules voting we use the VB matrix capsule routing procedure detailed in [150]. Additionally, during capsule convolutions we learn an inverse graphics matrix $\hat{y}_W$, similarly to what Ramìrez et al. introduce in chapter 3 of [149], but working with matrix capsules and the VB routing. Given each lower level capsule i and the corresponding higher level capsule j, we define M_i as the lower level pose matrix and $W_{ij} \in R^{4\times4}$ a trainable viewpoint-invariant

transformation matrix such that:

$$V_{j|i} = M_i W_{ij} \tag{3.5}$$

Figure 3.7: Capsule voting through trainable transformation matrices.

where $V_{j|i}$ is the vote coming from lower capsules i for higher capsules j, as we show in Eq. 3.5.
The output of the ClassCaps layer will be flattened into a $BS \times 272$ latent space vector representation, which contains compressed 3D, 2D and joint heatmaps data for the entire batch.

Decoding

The final part of the proposed architecture deals with decoding the $BS \times 272$ latent space coming from capsule layers, to obtain 2D, 3D and heatmaps joints representations. During joints reconstruction we employ three separate dense sub-networks, without sharing any layer between the three. This is crucial because the 2D information is not used to 'lift' the 3D joints, and viceversa. For every dense sub-network, we introduce a DDGELU (Dense Dropout GELU), which we define as a sequential combination of a dense linear layer, followed by a GELU activation and finally dropout, as shown in Eq. 3.6.

$$\text{DDGELU}_{0.3}(x) = Dropout_{0.3}(\text{GELU}(Linear(x))) \tag{3.6}$$

Figure 3.8: Dropout Dense Gaussian Error Linear Unit (DDGELU).

During training we noticed both an increment of convergence speed and a decrease of the train-test loss gap, which should lead to better generalization of the model. As shown in Algorithm 1, we iterate 2 DDGELU layers for each

desired output, and choose three different activation functions, depending on the output:

- *sigm* activation for the 2D joints
- *tanh* activation for the 3D joints, which extended range between -1 and 1 better suits the considered 3D domain
- *ReLU6* activation for the joint heatmaps, which provides a good convergence during training for the reconstruction task

Finally, we reshape the heatmaps vector into $BS \times E \times 64 \times 64$ feature maps, which are given as input to the final transposed convolution layer; this will produce full-size $BS \times E \times 256 \times 256$ feature maps. During training, transposed convolutions have shown better results and faster convergence when compared to simple bilinear or bicubic interpolation. A summary of the three final outputs is shown in Fig. 3.9.

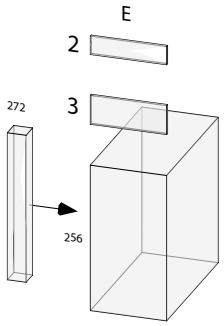


Figure 3.9: Decoding phase: from entities vector to 2D, 3D joints and 256×256 feature maps.

Loss Multi-tasks network training has shown multiple advantages in learning efficiency and prediction accuracy over the years. For this reason, we encourage the network to jointly learn multiple tasks (3D, 2D, and heatmaps) by employing a self-balancing loss that takes into account the contribution of each task (Eq. 3.7). To clarify, we stress the fact that during joints reconstruction we employ

three separate reconstruction sub-networks, so that the network is forced to learn a multi-task enabled latent space.

$$\begin{aligned}\mathcal{L}(x) &= \sum_{\tau \in \mathcal{T}} \left(s_\tau + e^{-s_\tau}\mathcal{L}_\tau\right) \\ &= \left(s_{3D} + e^{-s_{3D}}\mathcal{L}_{3D}\right) + \left(s_{2D} + e^{-s_{2D}}\mathcal{L}_{2D}\right) \\ &+ \left(s_H + e^{-s_H}\mathcal{L}_H\right) + \left(s_W + e^{-s_W}\mathcal{L}_W\right) \\ \mathcal{T} &= \{3\mathcal{D}, 2\mathcal{D}, \mathcal{R}, \mathcal{W}\}\end{aligned} \tag{3.7}$$

Figure 3.10: Self-balanced multi-task loss for the proposed architecture (3D joints, 2D joints, joint heatmaps, variational capsule routing regularization).

The proposed loss L is able to self balance through trainable register buffers s_{3D}, s_{2D}, s_H, s_W which dynamically change value to weight the contributions for each task, at the same time mitigating overfitting for single contributions. To demonstrate the positive effects of multi-task learning in the considered training scenario, we experimented with different sub-networks combinations for the decoder, as well as with only one sub-network. The overall convergence at training time was slower and worse in every test involving less than 3 tasks, while the best results were obtained when enforcing at least 3 tasks (2D joints estimation, 3D joints estimation, joint heatmaps reconstruction). The main explanation is that by forcing the network to perform multiple tasks, it is encouraged to organize its latent space in a more efficient way, leading to a more coherent and less cluttered features representation. We stress the fact that every sub-network does not share any parameter with each other, to further promote this effect. Multi-task learning also enforces a better representation of data points in the latent space (Fig. 3.11). Same class joints tend to cluster together better when enforcing the multi-task constraints, while at the same time reducing the number of outliers.

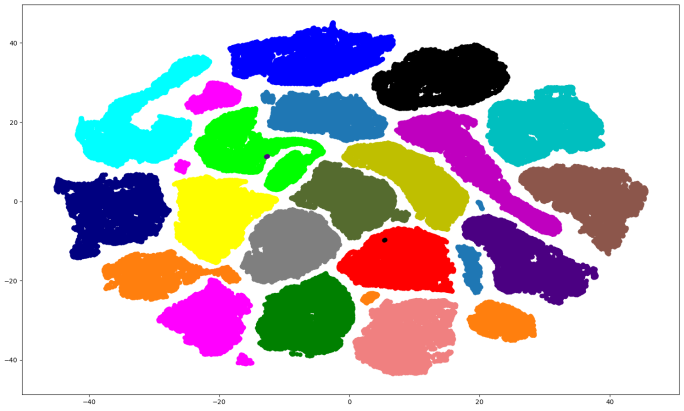


Figure 3.11: Organization of the latent space after t-SNE: the color of each sample point corresponds to a joint class.

3.2.2 Results

For training and evaluation we follow the default protocol #1 from Human3.6M [82], by reserving subjects 9 and 11 for evaluation, while only training on data from subjects $1, 5, 6, 7, 8$. Our architecture is fully end-to-end, requiring as input just one image and no additional information such as 2D joints ground truth, multiple sequential frames, or non-standard data-augmentation. Compared to the majority of methods present in literature, we don't rely on additional datasets for training, at the same time showing high generalization capabilities even after training on a subset of the possibly available data. The metrics we use for comparison are the Mean Per Joint Position Error (MPJPE) in millimiters for each of the 15 activities in the Human3.6M dataset and the average by activity MPJPE, for each camera in the dataset. As for the implementation, the network we present is written using Pytorch Lightning, focusing on high modularity, allowing for real-time joint 3D and 2D predictions, achieving over 229 FPS ($0.00436 s/frame$) on an Nvidia GeForce 1080Ti (desktop) and over 52 FPS ($0.01913 s/frame$) on and Nvidia GeForce 1050 Mobile (laptop), almost twice as fast as what is reported in [149]. All the results were

conducted on the same exact hardware and in the same conditions. In Fig. 3.12 we show the activity-wise and mean frames per second of our architecture compared to the other capsule-based networks [149] on a high-end, desktop-grade GPU. In this scenario, our architecture allows for a 2.33× speed-up. Even in more resource-constrained scenarios (laptop-grade GPU, Fig. 3.13) we manage to gain an additional $\sim 15FPS$ on average. According to our experiments, the biggest improvements in terms of speed mostly come down to a combination of simplified network structure, the usage of the improved capsule paradigm and faster routing. In the following sections we show some quantitative and qualitative results as well, both from the Human3.6 dataset and in-the-wild.

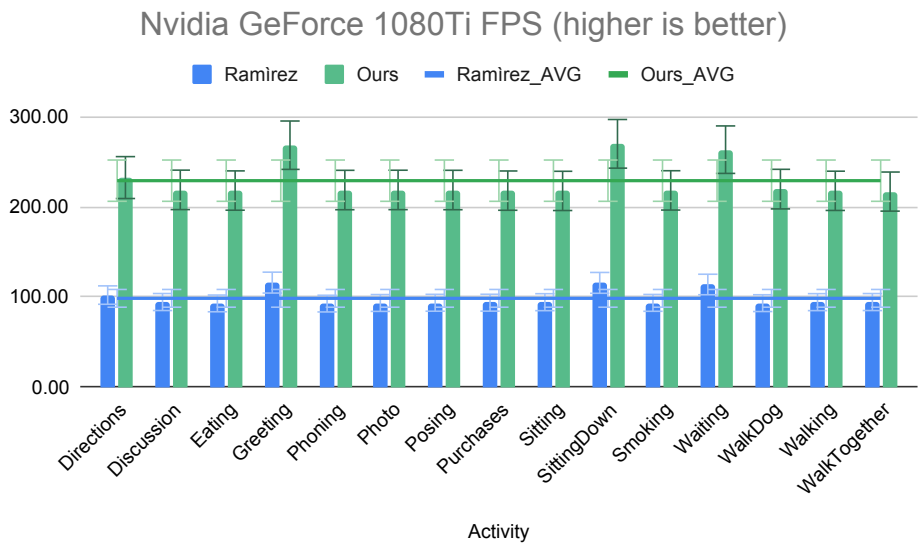


Figure 3.12: Activity-wise and average inference speed comparison on the same hardware (Nvidia GeForce 1080Ti).

Quantitative results

In Table 3.1 we show our results compared to the state-of-the-art methods, both the ones using Procrustes transformation (right) and the ones reporting results without Procrustes (left). We achieve the lowest average MPJPE on both the

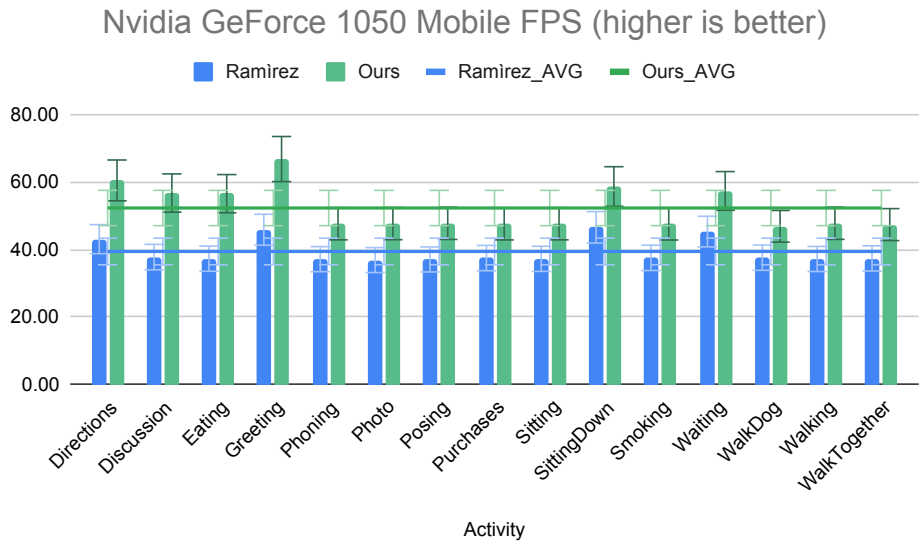


Figure 3.13: Activity-wise and average inference speed comparison on the same hardware (Nvidia GeForce 1050 Mobile).

categories and on most of the activities, without using additional information or non-standard data augmentation. Works using additional data, such as 2D-to-3D lifting, ground truth 2D joints, multiple datasets or temporal information are marked in Table 3.1 with a * symbol. We achieve similar or better results even with those methods, without relying on additional information, dataset or data augmentation, as shown in Table 3.2. Even considering other similar works that employ additional information, we obtain the lowest average MPJPE scores (yellow row). Compared to the only other work in literature using CapsNet [149], our model achieves better MPJPE in almost every activity.

For the sake of completeness, we selected the top recent works in literature (2019-2020) with the lowest average MPJPE on the Human3.6M dataset, working on monocular data (Table 3.2). However, as Table 3.2 shows, most of the works are aided by 2D ground truth information, meaning that they cannot be properly considered end-to-end. Additionally, many of them even exploit temporal frame sequences to refine joint predictions, thus non working with sin-

Activity	No Procrustes								Procrustes			
	Zhou * [214]	Tekin * [177]	**Tome, I** * [182]	Ramìrez, I [149]	Tome, II * [182]	Ramìrez, II [149]	Ramìrez, III [149]	Ours, I	Sanzari * [153]	Bogo * [16]	Ramìrez, IV [149]	Ours, II
Directions	87.36	85.03	68.55	79.42	*64.98*	73.15	73.33	**70.16**	*48.82*	62	57.55	**55.02**
Discussion	109.31	108.79	78.27	83.73	*73.47*	84.95	83.45	**76.67**	*56.31*	60.2	61.32	**58.06**
Eating	87.05	84.38	77.22	84.01	*76.82*	85.87	85.33	**78.41**	95.98	67.8	66.48	**60.91**
Greeting	103.16	98.94	89.05	83.15	86.43	80.12	79.08	**76.87**	84.78	76.5	64.49	**61.69**
Phoning	116.18	119.39	91.63	**86.42**	*86.28*	91.44	89.99	87.99	96.47	92.1	68	**66.49**
Photo	143.32	95.65	110.05	112.38	110.67	**109.42**	109.95	109.49	105.58	*77*	83.16	**80.02**
Posing	106.88	98.49	74.92	81.34	*68.93*	76.40	76.08	**72.23**	66.3	73	56.05	**54.94**
Purchases	99.78	93.77	83.71	77.65	74.79	76.72	73.61	**73.12**	107.41	75.3	54.85	**52.89**
Sitting	124.52	*73.76*	115.94	105.10	110.19	105.54	**104.12**	108.84	116.89	100.3	**77.65**	80.11
SittingDown	199.23	170.40	185.72	135.55	173.91	**130.15**	136.27	149.53	129.63	137.3	**97.32**	99.84
Smoking	107.42	85.08	88.25	88.25	*84.95*	88.07	87.59	**87.29**	97.84	83.4	**67.31**	67.86
Waiting	118.09	116.91	88.73	79.24	85.78	80.25	79.19	**75.14**	65.94	77.3	59.63	**57.71**
WalkDog	114.23	113.72	92.37	87.45	*86.26*	88.75	**87.13**	87.70	130.46	79.7	**64.76**	65.28
Walking	79.39	62.08	76.48	67.56	71.36	**66.10**	66.31	**65.38**	92.58	86.8	**49.96**	51.19
WalkTogether	97.70	94.83	77.95	80.45	*73.14*	76.84	76.88	**75.76**	102.21	81.7	**60.47**	61.04
Avg, by activity	112.91	100.08	93.26	88.78	88.53	87.58	87.22	**86.17**	93.15	82.03	65.93	**64.98**
Std, Dev,	27.78	24.21	27.63	16.28	26.21	**15.86**	17.15	20.97	23.97	17.9	**11.74**	12.55

Table 3.1: Activity-wise MPJPE scores for comparable works (with and without Procrustes transformation), including the top-3 in CVPR'17 Human 3.6 challenge and the top-3 IJCVm Jan'18. Columns marked with * make use of additional information or datasets, among the ones depicted in Table 3.2. Results in **bold** show the best MPJPE score among methods not relying on multiple datasets or additional information at training time. *Underlined* results show the best MPJPE score among all the methods, including the ones employing additional training time information.

gle images. Others use additional datasets, hand-crafted data augmentation of biometric models during training. We stress the fact that a big advantage of employing capsule networks is the increased generalization capabilities, which highly reduce the need for additional training data, and at the same time boosting network efficiency. Nonetheless, even considering the more recent results that use additional information or datasets, our results remain comparable.

Qualitative results

In Figs. 3.14a, 3.14b we show some qualitative results for the *Walking* and *Sitting Down* activities from test examples of the Human3.6 dataset. Starting from the upper left: input RGB image, predicted 3D pose, ground truth 3D pose,

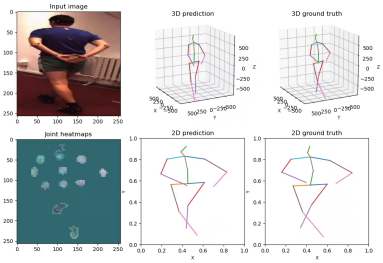
	Year	L.	T.	M.D.	D.A.
Cheng [29]	2020		X	X	X
Pham [144]	2019	X	X	X	
Zhao [212]	2019	X		X	
Chen [26]	2020		X		X
Lin [112]	2019	X	X		
Sharma [158]	2019	X		X	X
Tripathi [185]	2020	X	X		X
Wandt [191]	2019	X			X
Arnab [8]	2019	X	X	X	
Mehta [125]	2019	X	X	X	
Ours	2020				

Table 3.2: Comparison of the most relevant competing methods from 2019-2020 (top Average MPJPE on Human3.6M). **L.**: using 2D joints ground truth and/or lifting from 2D joints, **T.**: using temporal information, **M.D.**: using multiple training datasets, **D.A.**: using non-standard data augmentation techniques or biometric models. In the table we did not include works with lower Average MPJPE than ours.

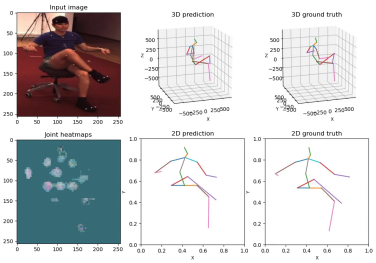
combination of the 17 "attention" heatmaps, predicted 2D pose and ground truth 2D pose. In Fig. 3.14c we show some in-the-wild results (no ground truth is present in this case).

Generalization capabilities and the effects of data augmentation

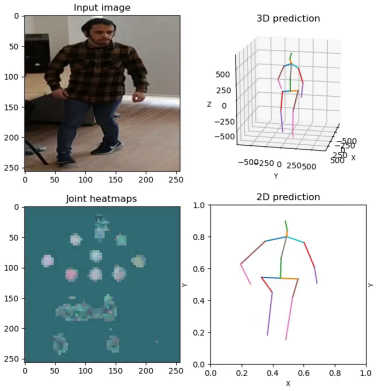
One of the main issues that we address in this paper is the promotion of generalization and viewpoint-equivariance capabilities in deep networks. A huge drawbacks of using deep networks is their intrinsic data-dependency, meaning that the task of learning a dataset almost always leads to some degree of overfitting of the network to the underlying data. For this reason, even the best performing HPE methods in Table 3.2, completely fail when dealing with previously unseen novel viewpoints. As we show in Fig. 3.15 our network is capable to cope with novel and extreme viewpoints, such as the top-view viewpoint, thus proving the advanced generalization capabilities of our network. Even when training on data coming from a single viewpoint (e.g. front-view) and testing with data from a completely different viewpoint (e.g. top-view, Fig. 3.15), our network shows very good generalization capabilities, which are achieved



(a) Results from 'Walking' activity.



(b) Results from 'Sitting Down' activity.



(c) In-the-wild results.

Figure 3.14: Qualitative results on the Human3.6M dataset (a, b) and in-the-wild (c)

.

through the hierarchical representation of joints as capsule entities in the latent space. Moreover, in our tests we experienced little to no benefit of using classic data augmentation during training, effectively showing how implicitly learning frame of references and viewpoints has a broader impact on vanilla networks during training time, allowing to greatly reduce the training dataset size, simultaneously simplifying the network complexity and boosting its generalization capabilities.

3.2.3 Discussion

We presented CapsulePose, the first human pose estimation architecture based on matrix capsules [76]. The method is real-time and operates in an end-to-end fashion with single images. The simple, modular architecture, paired with a fast Variational Bayes routing and modern frameworks, contributes to achieve very

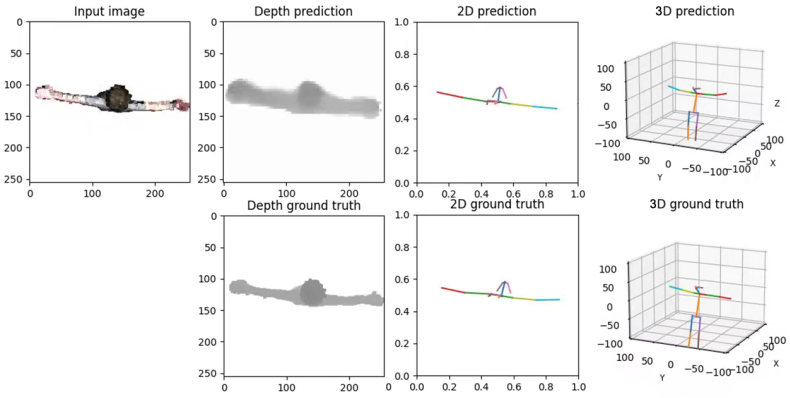
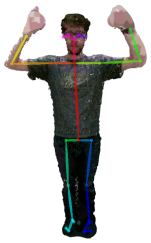


Figure 3.15: CapsulePose working on rasterized top-view images of the Panoptic Dataset [84] point clouds. Assigned tasks: depth estimation, 2D and 3D top-view pose estimation.

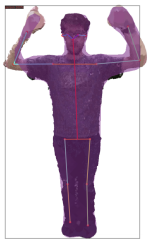
fast performance, running almost twice as fast as other CapsNet-based methods, while at the same time performing better on Human3.6M benchmarks. We show how the presented architecture is competitive with respect to state-of-the-art networks, even by not relying on any additional information or data augmentation at training time, making it a simple yet effective baseline network.

3.3 PanopTOP

In the field of human pose estimation (HPE), depth and RGB sensors are commonly employed in a wide range of applications, from robotics to immersive entertainment and from surveillance to smart spaces. [21, 63, 68]. Such a diverse application range requires cameras to capture humans from a wide variety of different angles. Thus, HPE frameworks should be able to retrieve the body pose from multiple different viewpoints. Currently, existing human pose estimation methods [21, 68, 88, 97, 101, 183] achieve good performances from many different camera viewpoints, but the most challenging ones. As shown in Fig. 3.16, good performances are achieved when retrieving the human pose from front-view images, and poor results can be obtained when dealing, for

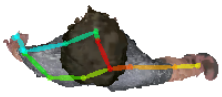


(a) OpenPose front



(b) MaskRCNN front

(c) HMR front



(d) OpenPose top

(e) MaskRCNN top

(f) HMR top

Figure 3.16: OpenPose, MaskRCNN and Human Mesh Recovery baselines (front, top views). All the methods perform very well on front and side views (Fig. 3.16a, 3.16b, 3.16c). However, when dealing with top-view images, current methods fail to correctly retrieve the human pose (Fig. 3.16d, 3.16f) or to even recognise the object as a human body (Fig. 3.16e).

Dataset	RGB	Depth	Top-view	Multi-View	2D Pose GT	3D Pose GT	Camera parameters
PanopTOP31K	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**
ITOP	N	**Y**	**Y**	**Y**	N	**Y**	**Y**
EVAL	N	**Y**	N	N	N	**Y**	N
TVPR	**Y**	**Y**	**Y**	N	N	N	N
TVPR 2	**Y**	**Y**	**Y**	N	N	N	N
K2HPD	N	**Y**	N	N	N	**Y**	N
UBC3V	N	**Y**	N	**Y**	N	**Y**	**Y**
Human3.6M	**Y**	N	N	**Y**	N	**Y**	**Y**

Table 3.3: Reference datasets for multi-view and viewpoint-invariant networks training. Only few datasets propose true top-view ground truth data, and most of them mainly focus on depth images, discarding the RGB component.

example, with the top-view. In this paper, we introduce a complete pipeline, called PanopTOP, which is based on computer graphics and that allows generating new RGB, depth, and pose samples from arbitrary viewpoints from the raw point cloud data. Our work aims at solving the following issues:

- ground truth alignment: we provide pixel-perfect aligned RGB and depth images regardless of the viewpoints, as well as the 2D and 3D ground truth pose;
- we encourage the usage of true multi-view cameras, allowing to obtain ground truth data from virtually every viewpoint and specifically the top-view one;
- our method employs a full pinhole camera model, allowing us to customize every aspect of the camera parameters, including intrinsic and extrinsic parameters. The remaining data (RGB, depth images, and reprojected 2D joints ground truth) is automatically changed according to the changes in the camera parameters.

To prove the effectiveness of our PanopTOP framework, we introduce PanopTOP31K, a training dataset specifically built for viewpoint invariant human pose estimation from depth and RGB images, consisting of 31K images of 23 different subjects recorded from diverse and challenging viewpoints. By using our PanopTOP method, it is possible to configure virtual cameras while fixing the existing 3D ground truth. To the best of our knowledge, PanopTOP31K is the first dataset that provides both top-view RGB images, as well as the corresponding 3D and 2D pose ground truth. Annotated poses in top-view RGB datasets are not available because of the difficulty of annotation, mostly due to occlusions. In Table 3.3 we show how the PanopTOP31K dataset provides the most complete and diverse set of poses and ground truths when compared with similar datasets. We argue that the complete set of multi-view RGB and depth

images along with 2D and 3D ground truth provide HPE researchers with the necessary data for the development of viewpoint-invariant frameworks.

We create the PanopTOP31K dataset starting from the six-degrees-of-freedom (6DoF) videos of the Panoptic dataset [85]. The dataset provides the pose ground truth for each video frame. Since 6DoF videos provide a 3D model of the scene, it is possible to generate a virtually infinite number of new 2D semi-synthetic RGB and depth images, in a bullet time fashion and from multiple viewpoints, simultaneously. In this way, we can create realistic videos on the fly, captured from different angles. To prove the suitability of the dataset for further developments, we show how baseline algorithms [133] trained on our PanopTOP31K can generalize on real data leading to improved results on multiple datasets.

Our contributions can be summarised through the following key steps:

i. We propose a method to generate new RGB and depth datasets with a virtually infinite number of semi-synthetic viewpoints, called PanopTOP.

ii. We propose a multi-view dataset, called PanopTOP31K, which consists of 31 thousand poses of 23 different subjects, rendered from the front, side, and top viewpoints in both RGB and depth domains.

iii. We provide baseline results for the novel PanopTOP31K on RGB images.

iv. We show the improvement of performances on multiple views given by our dataset cross-validating on different scenarios and viewpoints.

Datasets

In the context of RGB images, there is a lack of datasets providing multiple viewpoints, and in particular the top-view viewpoint. For example, common large HPE datasets in literature such as Human3.6M [82] and the Panoptic

Dataset [85] provide RGB images from multiple views, still lacking the top-view component. Other datasets, such as K2HPD Body Pose Dataset [193] and ITOP [63] only provide top-view and side-view depth images, lacking the matching RGB ground truth data. TVPR and TVPR2 datasets [111, 142] also provide a top-view point of the scene, but they do not provide information about 3D joints, making the dataset not suitable to solve HPE-related tasks. The lack of multi-view datasets in the RGB domain leads to popular out-of-the-box networks, such as OpenPose [21] and MaskRCNN, [68] experiencing a big decrease in performance whenever the viewpoint is changed. A possible solution to the aforementioned annotation problem is to rely on fully synthetic datasets to generate data from custom viewpoints. However, many works [48, 188] show that the gap between visual realism in computer-generated and real-world images contributes to an even bigger gap in the models' prediction performance. Relying on photo-realistic rendering helps to mitigate this issue. However, many other aspects contribute to the overall perceived realism, especially when dealing with humans and human poses. Our perception of reality is influenced by unrealistic or inconsistent body proportions through time, fake lighting, physics, and slightly off skeletal movements. The same effect is shown when training HPE models on synthetic data and testing on real-world images, or vice versa. Other works [197] employ synthetic data to augment the already available datasets, but the same issues apply, even if to a lesser extent. In this work we adopt a hybrid solution, relying on real-world 3D human scans to generate new semi-synthetic data. We thus preserve the photo-realism of the rendered scene while maintaining all the advantages typical of fully synthetic approaches.

3.3.1 The PanopTOP framework

In this section we present PanopTOP, a fully automated pipeline for creating multi-view HPE datasets, starting from real-world 3D data and ground truth

joints. Next, we employ the PanopTOP framework to create the PanopTOP31K dataset. The main advantages of our method are: (i) it automatically provides RGB and depth images, along with the 2D and 3D ground truth, requiring the user's input only when positioning the virtual cameras; (ii) it is highly customizable, meaning that it allows to precisely tune each camera, including their intrinsic and extrinsic parameters, as well as the desired RGB and depth output quality; (iii) it outputs data suitable for multiple tasks, such as 2D/3D human pose estimation, detection, segmentation, view synthesis and others in both RGB and depth domain.

The proposed framework generates RGB, depth, 2D and 3D joints ground truth data, starting from raw point clouds. We used the Panoptic dataset [85], although similar datasets can also be used. The complete pipeline is shown in Fig. 3.17 and includes the following steps:

1. point cloud retrieval and coordinate system setup;
2. skeleton-based point cloud filtering;
3. mesh reconstruction;
4. virtual cameras positioning;
5. hidden points removal;
6. rendering.

Each processing step is further detailed in the next subsections. The proposed method allows for a manifold of different configuration parameters, such as the number of virtual cameras to be used for the image generation and their global position, the density of point clouds and meshes, and the resolution of the output images. Furthermore, it also takes care of all the steps required to generate the dataset, from fetching the raw point clouds to saving the desired dataset to memory. Moreover, since our architecture leverages high-speed GPU

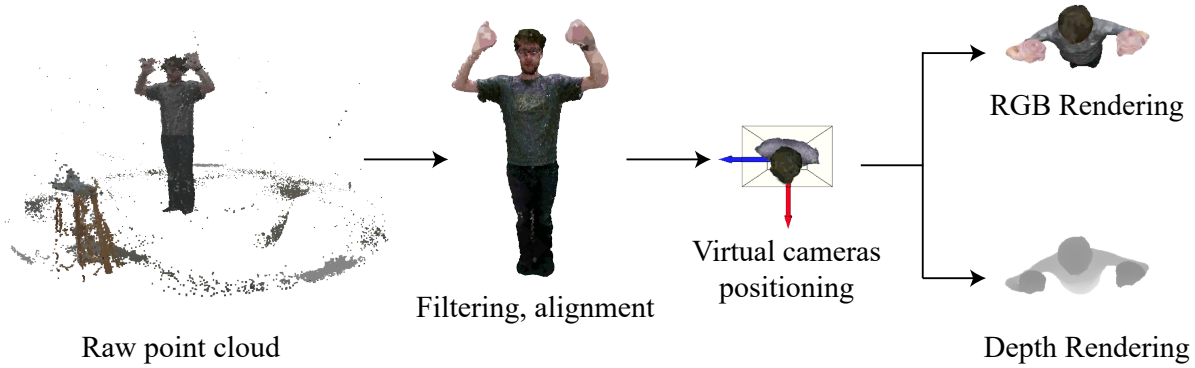


Figure 3.17: PanopTOP rendering process: the raw point cloud is filtered and aligned with the chosen coordinate system. Virtual cameras are positioned into the scene for simultaneous RGB + depth rendering.

operations, it could be used to automatically generate new batches on-the-fly, and directly use them as input to train a neural network, without saving them to memory in advance.

Point cloud retrieval and coordinate system setup

Starting from the Panoptic dataset [85], each point cloud is retrieved and then transformed to a center of origin, scale, and coordinate system of choice. For consistency, we adopt the same coordinate system and scale used in the ITOP dataset [63] by default. The center of origin is selected based on the ground truth pose data associated with the point cloud. At this stage, no additional filtering is employed, the output is a raw point cloud with many outliers and aligned to a chosen coordinate system.

Skeleton-based point cloud filtering

The ground truth 3D pose data is used to compute an axis-aligned 3D bounding box containing the subject. All the points that do not belong to the subject (Fig. 3.17) are outside of the bounding box, and are thus removed. An additional skeleton-based point cloud filtering based on the L2 distance d between the

closest joint j in the 3D skeleton and each 3D point p_i is then applied:

$$d = \sqrt{\sum_{i=0}^{P} (p_i - j)^2} \tag{3.8}$$

Subsequently, statistical and sphere radius outliers are removed, only keeping the points that belong to the subject, thus greatly reducing noise.

Mesh reconstruction

Here we describe how 3D meshes can be reconstructed from the filtered point clouds if needed. They can be useful as ground truth for a more efficient 3D model analysis since the mesh provides more structured information and a smoother texture. The point cloud's vertex normals are estimated by looking for adjacent points and using covariance analysis to calculate their principal axis. Then the normalized point cloud is converted into a 3D mesh via Screened Poisson Surface Reconstruction [92]. Finally, surface subdivision and mesh smoothing are employed. These steps allow to obtain a smoother surface and thus a better rendering of both the RGB and depth outputs.

Virtual cameras positioning

A configuration file is designed to create virtual cameras with user-defined intrinsic and extrinsic parameters, for later rendering of both RGB images and depth maps. Users can also manually adjust the camera position in an interactive visualization window containing a preview of the rendering.

Hidden points removal

Optional hidden points removal is performed via Direct Visibility of Point Sets [91]. Since the input point cloud may be a combination of multiple viewpoints clouds, as in the Panoptic dataset, it may be necessary to remove occluded points

to replicate the standard format of the majority of the other datasets. By default, we keep this option enabled to promote consistency with the ITOP dataset [63].

Rendering

RGB and depth images are finally rendered for each camera and for each point cloud sample in the dataset. The 2D and 3D ground truth of the scene is automatically generated starting from the 3D pose matrices and the extrinsic and intrinsic camera matrices. By default, we also convert the 19-joints skeleton to the 15-joints model to be consistent with the ITOP dataset [63].

3.3.2 The PanopTOP31K dataset

Our method shown in Sec. 3.3.1 allows for the generation of an arbitrary number of viewpoints for each point cloud. Starting from the Panoptic dataset, we apply our pipeline to generate a new dataset, called PanopTOP31K. The dataset contains approximately $30K$ RGB images, $30K$ depth maps, $10K$ filtered point clouds, and $10K$ 3D meshes from 23 different subjects recorded from the front, side, and top view ($\sim 10K$ RGB images for each viewpoint). Each RGB image and depth image have size 256×256 with depth 3. The provided pose ground truth employs the 15 joints skeleton model as in [63].

3.3.3 Experiments

In this section, we show the results obtained by some popular out-of-the-box human pose estimation networks, both for the front, side, and top view on our PanopTOP31K dataset on RGB images. Then, we validate our semi-synthetic dataset showing how it achieves good results when used for data augmentation and domain adaptation on depth images.

	Experiment	Head	Neck	Shoulders	Elbows	Hands	Torso	Hips	Knees	Feet
(a)	**[I],[I],[I]**	99.50	99.60	99.05	**97.90**	**90.80**	***100.00***	98.55	95.20	87.15
(b)	**[I],[I],[P]**	96.60	97.90	93.80	76.10	63.60	97.80	89.90	84.60	46.50
(c)	**[I],[I+P],[P]**	97.20	98.10	95.45	77.15	59.10	98.00	90.25	70.20	35.80
(d)	**[I+P],[I+P],[P]**	**98.50**	**99.70**	**99.70**	***98.20***	***90.90***	**99.70**	**99.40**	95.80	***95.55***
(e)	**[P],[P],[P]**	**98.50**	**99.70**	**99.70**	97.80	90.85	99.60	99.35	***96.30***	95.45
(f)	**[P],[P],[I]**	99.50	99.50	98.10	93.90	61.45	99.30	94.85	75.45	26.80
(g)	**[P],[I+P],[I]**	99.60	99.80	97.95	94.00	66.60	99.50	94.45	83.55	59.20
(h)	**[I+P],[I+P],[I]**	***100.00***	***100.00***	***100.00***	97.80	90.35	***100.00***	***99.55***	***96.30***	**89.35**

Table 3.4: Percentages of correctly detected joints for the ITOP and PanopTOP31K datasets in our 8 conducted experiments. Each experiment is identified by a letter (**a-h**) and a data split **[train],[validation],[test]** (**P** = PanopTOP31K, **I** = ITOP). Each value represents the percentage of joints with L2 distance smaller than a threshold $T = 0.2m$ from the ground truth. The top scores for each joint regarding tests on the ITOP dataset are highlighted in **blue**, while the PanopTOP31K ones are highlighted in **green**. The top overall scores for each joint are in ***italic***.

Baselines on RGB images

OpenPose [19, 21], MaskRCNN [68] and HMR [88, 100] are three popular methods for HPE in the RGB domain. We take the off-the-shelf pre-trained networks of all the baseline algorithms for testing on our new dataset, PanopTOP31K. As shown in Figs. 3.16a, 3.16b, 3.16c, we obtain good results on the side views. However, when dealing with top view images, all the methods fail in detecting the pose from more than 90% of the dataset images. For example, MaskRCNN misclassifies as 'cake' a subject as seen from the top view (Fig. 3.16e), while OpenPose fails to produce a coherent skeletal structure (Fig. 3.16d), despite being trained on the Panoptic dataset. A similar issue is encountered in HMR, which fails to correctly fit a mesh to the top-view image (Fig. 3.16f). This incorrect behavior explains that most human pose estimation networks are not trained to handle extreme viewpoints and thus they do not achieve viewpoint-invariance. Our method allows creating multi-view datasets for human pose estimation, that could be used to develop viewpoint invariant HPE

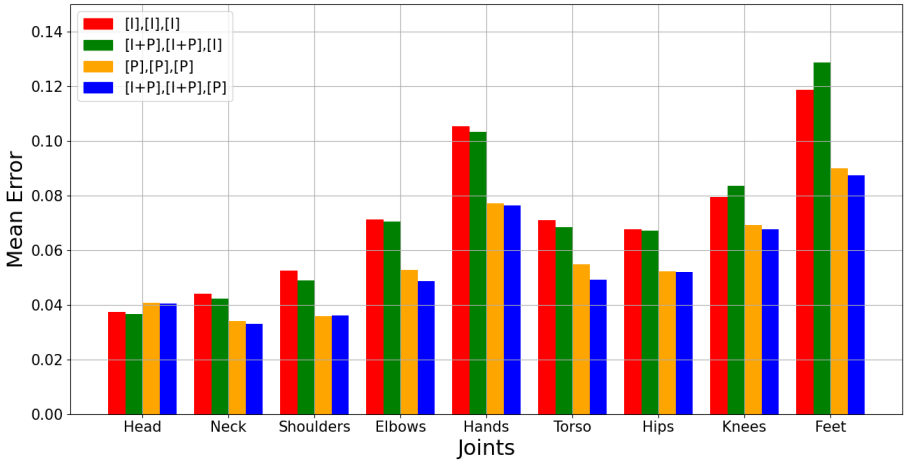


Figure 3.18: Mean per-joints errors in meters for ITOP and PanopTOP31K datasets, respectively, with (green, blue) and without (red, orange) training-wise augmentation. Red, green, yellow and blue bars correspond to experiments (a), (h), (e) and (d) respectively.

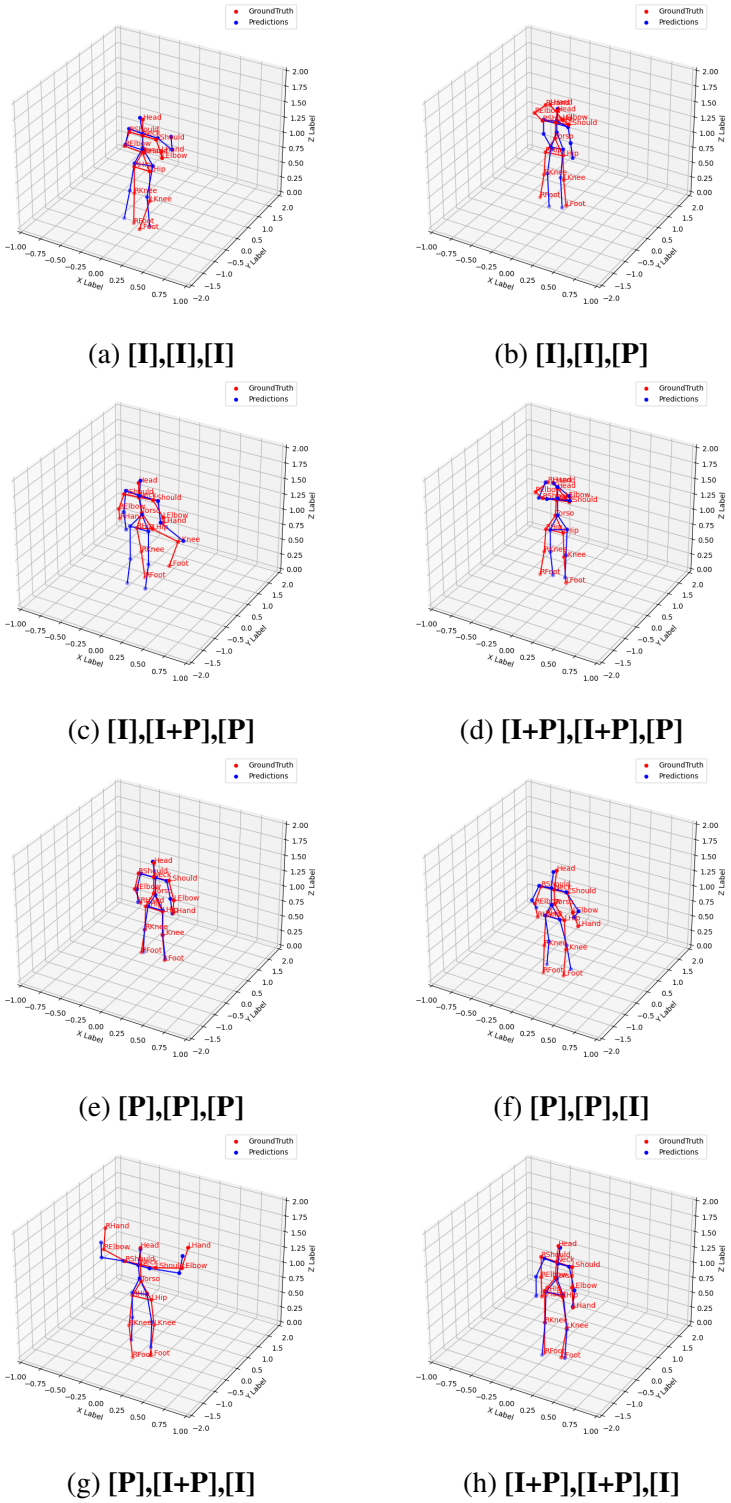
networks in the RGB or depth domain.

Dataset validation on depth images

We have shown how state-of-the-art methods for HPE on RGB images work on the PanopTOP31K dataset, failing in the case of top-view images. In this section, we focus on validating our dataset on depth images. Since human pose estimation already works well enough on front-view images, we focus our attention on a most difficult scenario, namely top-view. We use a vanilla version of the V2V network [133] on depth images for training and validating.

In the remainder of this section and in table 3.4 we use the following notation: **I** and **P** identify the ITOP and our PanopTOP31K dataset respectively. We use **[train],[validation],[test]** to indicate on which datasets the network has been trained, validated and tested.

We perform 8 different cross-validation experiments, from (a) to (h), as shown in Tab. 3.4. In Fig. 3.19 we show some qualitative results for the HPE task, while Tab. 3.4 and Fig. 3.18 report the percentage of correctly detected joints and the mean per-joint error respectively.



(a) **[I],[I],[I]** (b) **[I],[I],[P]**

(c) **[I],[I+P],[P]** (d) **[I+P],[I+P],[P]**

(e) **[P],[P],[P]** (f) **[P],[P],[I]**

(g) **[P],[I+P],[I]** (h) **[I+P],[I+P],[I]**

Figure 3.19:

Qualitative results on multiple **[train],[validation],[test]** data splits, corresponding to quantitative results in Table 3.4. As an example, the notation **[P],[I+P],[I]** means that the network has been trained on PanopTOP31K, validated on both ITOP and PanopTOP31K and tested on ITOP. Each experiment is identified by a letter (**a-h**).

Experiment (a) shows how the network performs well when trained, validated, and tested on the ITOP dataset. At the same time, in (b), the same training and validation split shows a poor ability to generalize on a new dataset. In (c), we provide a diverse validation set, but we achieve a small gain and in some cases, we even worsen the performances on hands and feet estimation, as shown in Tab. 3.4. This happens because the ITOP dataset is not diverse enough and it tends to overfit the training data. Experiment (d) shows how adding our PanopTop31K split to the training set results in a substantial improvement in performances with respect to previous cases (a)(b)(c). The PanopTOP31K dataset is thus suitable for augmenting a real-world dataset and it leads to a performance improvement.

Experiment (e) proves that the network can correctly process our PanopTOP31K dataset with good results. In (f), we obtain much better results than (b), and in some cases, they are also comparable with (a). This proves how the PanopTOP31K dataset is more able to generalize to different data than the ITOP one, without overfitting. Adding the ITOP validation split as in (g) allows the network to improve its performances with respect to (f), thus proving the robustness to the overfitting of the network trained on our dataset. Finally, experiment (h) shows how the best performances on real data are obtained by augmenting a real-world dataset with our semi-synthetic one. Both (f) and (h) validate the ability of our semi-synthetic dataset to provide samples that are realistic enough for the network to generalize well on real-world depth images.

3.3.4 Discussion

We presented PanopTOP, a new method for fully automatic multi-view datasets creation along with PanopTOP31K, the first multimodal RGB and depth dataset exhibiting challenging viewpoints for HPE. Our dataset allows for the training of viewpoint-invariant HPE networks from a manifold of data inputs (RGB images, depth images, point clouds, 3D meshes). Experiments on our semi-

Figure 3.20: **[Better seen in color].** Overview of the proposed solution. Two different views of the same subject are shown for each image: (a) 2D/3D Human Pose Estimation (HPE) and (b) 3D Human Mesh Recovery (HMR) methods achieve good accuracy on the front-view (second row). Changing the viewpoint turns into performance degradation (first row). Our method (c) promotes viewpoint equivariance, showing good results in both the RGB and depth domains.

synthetic PanopTOP31K dataset show promising results on top-view HPE, obtaining comparable results with popular real-world datasets and improving network accuracy when used for data augmentation.

3.4 DECA

Human pose estimation is key for many applications, such as action recognition, animation, gaming, to name a few [86, 159, 169]. State of the art methods [21, 182] that rely on RGB images can correctly localize human joints (e.g. torso, elbows, knees) in images, also in presence of occlusions. However, they tend to fail when dealing with challenging scenarios. The top-view perspective, in particular, turns out to be a difficult task; on the one hand, it causes the largest amount of joints occlusions, and on the other hand, it suffers the scarcity of suitable training data, as shown in Fig. 3.20.

When presented with unseen viewpoints, humans display a remarkable ability to estimate human poses, even in the presence of occlusions and unconven-

tional joints configurations. This is not always true in computer vision. In fact, available methods are trained in relatively constrained settings [85], with a limited variability between different viewpoints. Limited data, especially from the top-viewpoint, along with limited capabilities of modeling the hierarchical and geometrical structure of the human pose, results in poor generalization capabilities.

This generalization problem, known as the *viewpoint problem*, depends on how the network activations vary with the change of the viewpoint, usually after a transformation (translation, scaling, rotation, shearing). Convolutional Neural Networks (CNNs) scalar activations are not suitable to effectively manage these viewpoint transformations, thus needing to rely on max-pooling and aggressive data augmentation [32, 63, 133, 200]. By doing so, CNNs aim at achieving *viewpoint invariance*, defined as

$$f(Tx) = f(x) \tag{3.9}$$

According to this formulation, applying a viewpoint transformation T on the input image x, *does not change* the outcome of the network activations.

However, a more desirable property would be to capture and retain the transformation T applied to the input image x, thus obtaining a network that is aware of the different transformations applied to the input. Being able to model network activations that change in a structured way according to the input viewpoint transformations is also called *viewpoint equivariance* and it is defined as:

$$f(Tx) = Tf(x). \tag{3.10}$$

This is achieved by introducing *capsules*: groups of neurons that explicitly encode the intrinsic viewpoint-invariant relationship existing between different parts of the same object. Capsule networks (CapsNets) can learn part-whole relationships between so-called *entities* across different viewpoints [75, 76, 152],

similarly to how our visual cortex system operates, according to the recognition-by-components theory [13]. Unlike traditional CNNs, which usually retain viewpoint invariance, capsule networks can explicitly model and jointly preserve a viewpoint transformation T through the network activations, achieving *viewpoint equivariance* (Eq. 3.10).

Developing viewpoint-equivariant methods for 3D HPE networks leads to multiple advantages: (i) the learned model is more robust, interpretable, and suitable for real-world applications, (ii) the viewpoint is treated as a learnable parameter, allowing to disentangle the 3D data of the skeleton from each specific view, (iii) the same annotated data can be used to train a network for different viewpoints, thus less training data is required.

In this work, we address the problem of viewpoint-equivariant human pose estimation from single depth or RGB images. Our contribution is summarised as follows:

- We present **a novel Deep viewpoint-Equivariant Capsule Autoencoder architecture (DECA)** which jointly addresses multiple tasks, such as 3D and 2D human pose estimation.

- We show how our network works with limited training data, no data augmentation, and across different input domains (RGB and depth images).

- We show how the feature space organization, defined by routing the input information to build capsule entities, improves when the tasks are jointly addressed.

- We evaluate our method on the ITOP [63] dataset for the depth domain and on the PanopTOP31K [56] dataset for the RGB domain. We establish a new baseline for the viewpoint transfer task and in the RGB domain.

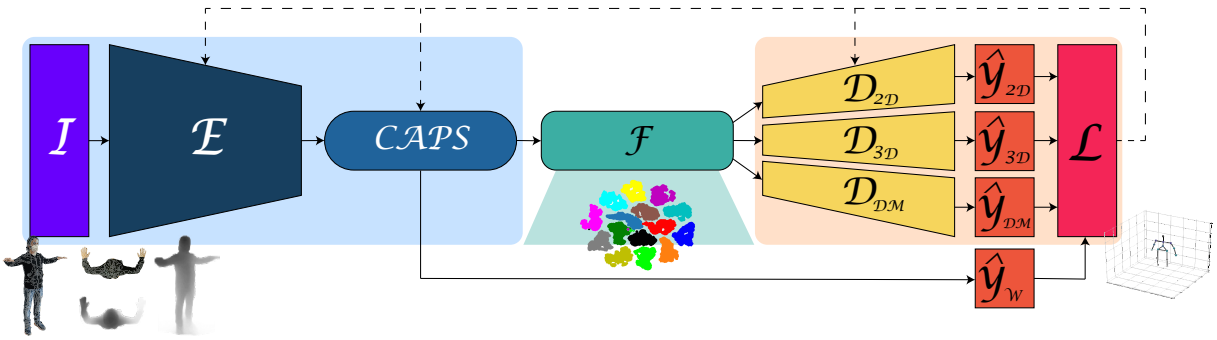


Figure 3.21: **[Better seen in color].** Overview of the proposed architecture. In light blue, the encoding module (Input, CNN encoder, Capsule layers), in green the interpretable feature space with capsule entities, in light orange the decoding module (fully connected decoders with multiple tasks and self-balancing loss).

3.4.1 Method

We now analyze the proposed autoencoder, DECA, starting with the capsule encoder and the multi-task decoders. DECA can be trained end-to-end, without any pre-training or data augmentation, and it works in real-time in the inference phase. An overview of the proposed architecture is shown in Fig. 3.21.

3.4.2 Capsule encoder

The encoding module of the network (light blue in Fig. 3.21) is divided in: (i) an input pre-processor I, (ii) a CNN encoder E and (iii) four layers of Matrix Capsules with Variational Bayes Routing [150].

(i) I is a layer which normalizes the different type of data (RGB images, depth images, top-view, side-view, free-view) in the interval $[0,1]$.

(ii) The normalised input is then forwarded to a CNN encoder E, built using four convolutional layers with inputs $[N_{ch}, 64, 128, 256]$, instance normalisation and GELU activations [69], as shown in Eq. 3.11. N_{ch} is the number of channels, which may vary depending on the input.

$$\text{GELU}(x) \approx 0.5x(1+\tanh\left[\sqrt{\frac{2}{\pi}}(x+0.044715x^3)\right]) \tag{3.11}$$

(iii) The output of the CNN encoder E feeds our capsule layers. It has been shown in previous works [76, 102, 152] that capsules provide a superior understanding of the viewpoint and the relationship between parts and parent objects, thus aiming at true viewpoint equivariance. Given the multiple degrees of freedom of each joint, we adopt the matrix capsules model [76] instead of vector capsules [152], enriching the description of single joints as hierarchically linked capsule entities. We deploy the novel capsule routing based on Variational Bayes (VB) [150], which is proven to speed up the training of our matrix capsules layers, at the same time improving performances. The last iteration of the VB routing is also called *ClassRouting* and it is used to route the highest-level information to the last layer of capsules before the feature space $\mathcal{F}$.

In our CapsNet, we employ four layers: a *primary capsules* layer encapsulates the output features of E into 16-dimensional capsules, two *convolutional capsules* layers refine the capsule features, and a final *class capsules* layer encodes the output into a J-dimensional features in the latent space $\mathcal{F}$, where J is the number of joints, also called *entities*.

Given each lower-level capsule i and the corresponding higher-level capsule j, we define M_i as the proposed lower level pose matrix and $W_{ij} \in \mathbb{R}^{4\times4}$ as a trainable viewpoint-equivariant transformation matrix such that:

$$V_{j|i} = M_i W_{ij} \tag{3.12}$$

where $V_{j|i}$ is the vote coming from lower capsules i for higher capsules j. The voting procedure takes place inside the VB routing and it allows each lower capsule i to route its information to a higher capsule j of its choice, thus allowing to build the hierarchical structure typical of CapsNets.

To promote the viewpoint equivariance in Eq. 3.10, we introduce an inverse matrix $\hat{y}_W$ in the *class capsules*, which aims at satisfying the Inverse Graphics constraint:

$$\hat{y}_W W_{ij} = I \tag{3.13}$$

meaning that the learned inverse matrix $\hat{y}_W$ effectively acts as an approximated inverse of the rendering operation, as it is commonly found in computer graphics [75].

At the output of the encoder, each *entity* corresponding to each joint of the skeleton is defined by a flattened vector of 16 elements, or, in other words, a 4×4 matrix, which is sufficient to grasp the complete pose (translation + rotation) of each joint.

An overview of the capsule encoder is shown in Algorithm 2. In the algorithm, $s_{3\mathcal{D}}, s_{2\mathcal{D}}, s_{\mathcal{DM}}, s_{\mathcal{W}}$ are weights used for the self-balancing of the loss, w_c are the convolutional layer weights, a are the activations of each Capsule layer, and $\{\cdot\}$ represents parameters used only when in the RGB domain.

Multi-task decoders

Starting from the 16-dimensional entities in the capsule feature space $\mathcal{F}$, we design a decoding module (light orange block in Fig. 3.21) that allows us to simultaneously retrieve multiple predictions for different tasks from the same feature space $\mathcal{F}$. Each decoder D_τ in the decoding module is configured as an independent fully connected block, with 0.5 Dropout and GELU activations [69]. We employ no weight sharing or layer sharing across the decoders to enforce the multi-task loss, as explained in section 3.4.2.

We define different tasks (τ) with different objectives:

- $3\mathcal{D}$: minimise the distance between ground truth and predicted 3D joints in 3D space $\hat{y}_{3D}$;

Algorithm 2: Capsule encoder

CapsuleEncoder (x)

inputs : $x = x_0 \dots x_{BS}$, $BS =$ batch size of RGB or depth images

outputs: $\mathcal{F} = J$ 16-dimensional *entities*; $\hat{y}_W =$ trainable Inverse Graphics matrix

$s_{3\mathcal{D}}, s_{2\mathcal{D}}, \{s_{\mathcal{DM}}\}, s_{\mathcal{W}} \leftarrow 1$;

$w_c \leftarrow xavier_{uniform}() \quad \forall c \in ConvLayers$;

foreach $i \in ConvLayers$ **do**

  $x \leftarrow Conv2d_i(x)$;

  $x \leftarrow InstanceNorm2d_i(x)$;

  $x \leftarrow \text{GELU}(x)$;

$a, x \leftarrow PrimaryCapsules(x)$;

foreach $j \in ConvCapsuleLayers$ **do**

  $a, x \leftarrow ConvCapsules_j(a, x)$;

  $a, x \leftarrow VBRouting_j(a, x)$;

$a, x, \hat{y}_W \leftarrow ClassCapsules(a, x)$;

$a, x \leftarrow ClassRouting(a, x)$;

$\mathcal{F} \leftarrow entities(x)$;

return $\mathcal{F}, \hat{y}_W$;

- $2\mathcal{D}$: as above, but without relying on 3D joints predictions, and rather predicting 2D joints $\hat{y}_{2D}$ as seen from the current viewpoint in camera frame coordinates;
- $\mathcal{DM}$: reconstruct the depth map $\hat{y}_{DM}$ of the input RGB image. It is used only in the RGB domain;
- $\mathcal{W}$ Inverse Graphics loss : learn the inverse graphics matrix $\hat{y}_{\mathcal{W}}$ to promote the *de-rendering* of input pixels into isolated capsule entities, as explained in Sec. 3.4.2, Eq. 3.13.

For each task $\tau = 3\mathcal{D}, 2\mathcal{D}, \mathcal{DM}$, a decoder D_τ takes as input the feature space $\mathcal{F}$ and it outputs the predictions $\hat{Y} = [\hat{y}_{3\mathcal{D}}, \hat{y}_{2\mathcal{D}}, \{\hat{y}_{\mathcal{DM}}\}]$ to the loss function. For $\mathcal{W}$, the $\hat{y}_{\mathcal{W}}$ matrix is forwarded to the loss function directly from the encoder.

An overview of the capsule decoders is shown in Algorithm 3.

Algorithm 3: Capsule decoders

CapsuleDecoders (x)
- **inputs :** $\mathcal{F} = J$ 16-dimensional *entities*
- **outputs:** $\hat{Y} = [\hat{y}_{3\mathcal{D}}, \hat{y}_{2\mathcal{D}}, \{\hat{y}_{\mathcal{DM}}\}]$
- $x \leftarrow \mathcal{F}$;
- **foreach** $i \in Y$ **do**
  - $x \leftarrow Dropout_{0.5}(x)$;
  - $x \leftarrow Linear(x)$;
  - $\hat{y}_i \leftarrow \text{GELU}(x)$;
- **return** $\hat{Y} = [\hat{y}_{3\mathcal{D}}, \hat{y}_{2\mathcal{D}}, \{\hat{y}_{\mathcal{DM}}\}]$;

Self-balancing multi-task loss

Tasks are associated to the different input domains, as follows:

Each task is assigned a loss $\mathcal{L}_\tau$, defined as:

- $\mathcal{L}_{2\mathcal{D}}$, $\mathcal{L}_{3\mathcal{D}}$: Mean Square Error (MSE) loss for the $3\mathcal{D}$ and $2\mathcal{D}$ joints prediction tasks.

	$3\mathcal{D}$	$2\mathcal{D}$	$\mathcal{DM}$	$\mathcal{W}$
Depth	✓	✓		✓
RGB	✓	✓	✓	✓

- $\mathcal{L}_{\mathcal{DM}}$: masked L1 loss for the depth estimation task $\mathcal{DM}$, in the RGB domain, where *mask* is a function that applies the L1 loss only on pixels over a certain depth threshold, to promote the depth estimation over non-background areas.

- $\mathcal{L}_{\mathcal{W}}$: inverse graphics loss $\mathcal{W}$, which role is to enforce invertibility for the capsule weight matrices. The notation $\|.\|_F$ defines the Frobenius norm of a matrix.

$$\mathcal{L}_\tau = \begin{cases} \mathcal{L}_{2\mathcal{D},3\mathcal{D}} = \frac{1}{BS}\sum_{i=0}^{BS}(y_i - \hat{y}_i)^2 \\ \mathcal{L}_{\mathcal{DM}} = \dfrac{\sum_{i=0}^{BS}\left[mask\,|y_i - \hat{y}_i| + |y_i - \hat{y}_i|\right]}{2 * BS} \\ \mathcal{L}_{\mathcal{W}} = \left\|\hat{y}_W W_{ij}\right\|_F \end{cases} \tag{3.14}$$

Considering $\mathcal{T}$ as the set of the employed tasks τ, the overall balanced loss for all the tasks is expressed as:

$$\mathcal{L} = \sum_{\tau \in \mathcal{T}} \left(s_\tau + e^{-s_\tau}\mathcal{L}_\tau\right) \tag{3.15}$$

where $s_\tau = [s_{3\mathcal{D}}, s_{2\mathcal{D}}, s_{\mathcal{DM}}, s_{\mathcal{W}}]$ are the trainable weights associated with each loss in $\mathcal{T}$, initialised to 1 in algorithm 2, and $\mathcal{L}_\tau$ is each loss of the enabled decoders, as defined in Eq. 3.14.

3.4.3 Experiments

Datasets

ITOP Dataset of depth images. The ITOP dataset [63] contains depth images from top and front view. The training split and the test split consist of 40k and 10k images, respectively. The depth images display 15 videos of 20 actors in a constrained setting. The dataset is recorded using two Axus Xtion Pro cameras. The 3D skeleton model consists of 15 joints.

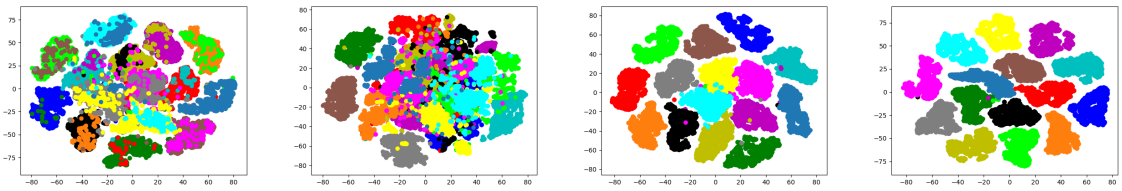
PanopTOP31K dataset of depth and RGB images. The PanopTOP dataset [56] consists of 31k top-view and 31k front view images coming from video sequences of 24 different actors, available both in the RGB and depth domain, for a total of 68k images. The ground truth 3D skeleton consists of 19 joints.

Evaluation metrics

Following the works of [63, 133, 200], we choose the mean average precision (mAP) as the evaluation metric for the depth domain. It is defined as the percentage of all predicted joints which fall in an interval smaller than 0.10 meters. In the RGB domain, we use the Mean Per Joint Position Error (MPJPE) in millimeters as in many HPE works [21, 149, 182].

Implementation details

Our network is trained in an end-to-end fashion using Pytorch Lightning. Input images are normalized in the interval $[0,1]$ with a resolution of 256x256 pixels for depth images and 256x256 pixels for RGB ones. We do not perform any augmentations on the input datasets. The batch size is set to 128 for ITOP and 128 for PanopTOP31K. We initialize the weights with the Xavier initialization [58]. The learning rate is set to $1e^{-5}$, the weight decay is set to 0, and Adam is the optimizer of choice. We train our network for 20 epochs on the ITOP dataset and 15 epochs on PanopTOP31K.



(a) V2V [133] (b) DECA-D1, $\mathcal{T} = [3\mathcal{D}]$ (c) DECA-D2, $\mathcal{T} = [3\mathcal{D}, \mathcal{W}]$ (d) DECA-D3, $\mathcal{T} = [3\mathcal{D}, 2\mathcal{D}, \mathcal{W}]$

Figure 3.22: 2D representation on the 16-dimensional latent space obtained using t-SNE [187]. Each dot corresponds to an entity E_{jt} representing a joint jt of the skeleton from the test set of ITOP [63]. V2V network [133] relies on CNNs, thus is not able to cluster together samples corresponding to the same entity (a). When trained to satisfy only the 3D prediction constraint our DECA-D1 network performs slightly better than V2V (b). The 15 clusters, corresponding to the 15 joints of the skeleton model, are clearly distinguishable in DECA-D2 (c) and DECA-D3 (d), with (d) displaying better cluster separation and fewer outliers.

Feature space entities and ablation study

We report experiments on the top-view of the ITOP dataset [63] to validate the 3D representation provided by our network and to show how the multi-tasks decoder influences the overall performances.

To do so, we deploy 4 configurations, 3 on depth data and 1 on RGB data, with different sets of tasks $\mathcal{T}$ of our method:

- DECA-D1, with $\mathcal{T} = [3\mathcal{D}]$
- DECA-D2, with $\mathcal{T} = [3\mathcal{D}, \mathcal{W}]$
- DECA-D3, with $\mathcal{T} = [3\mathcal{D}, 2\mathcal{D}, \mathcal{W}]$
- DECA-R4, with $\mathcal{T} = [3\mathcal{D}, 2\mathcal{D}, \mathcal{DM}, \mathcal{W}]$

where the letter D or R indicates the depth or RGB domains, and the number defines how many tasks are assigned to the network. Since we are evaluating the performances on the 3D HPE, the $\tau = [3\mathcal{D}]$ is used for all the different configurations.

Loss effectiveness analysis. The results are reported in the last 3 columns of Table 3.5. As shown in the Table, increasing the number of tasks in $\mathcal{T}$ generally leads to an increase in the network's performances. DECA-D1 already achieves similar results to the state-of-the-art, thanks to the CapsNets' capability to interpret the geometrical nature of the input data. When the inverse graphics loss $\mathcal{W}$ is employed (DECA-D2 and DECA-D3), the enforced invertibility of the weights matrix leads to an immediate gain in performances. In DECA-D3, the introduction of the $2\mathcal{D}$ loss leads to an additional improvement in terms of accuracy. Hence, we argue that the network performances improve when more tasks are given because we achieve a better representation of the entities in the latent space.

Latent space analysis. To analyze the latent space, we use the features of the test set extracted after the capsule modules. Each feature $f \in \mathcal{F}$ is linearised to obtain a vector of length L_{feat}. At this stage, each entity E_{jt} corresponding to each joint jt is defined by dividing each feature vector by the number of joints, resulting in vectors of length $\frac{L_{feat}}{\#of\,joints}$. For visualisation purposes, we use *t-SNE* [187] to project the entities on a 2-dimensional space. The results are displayed in Fig. 3.22. We compare our latent space against the publicly available version of the V2V [133] encoder/decoder structure. We show how our DECA network can better cluster and separate each entity E_{jt} with respect to V2V. Our solution provides a better organization of the latent space, with bigger inter-class margins and fewer outliers. The latent space organization improves drastically when we employ the $\tau = \mathcal{W}$ task (DECA-D2), thus enforcing the inverse graphics constraint. In DECA-D3 we add the $\tau = 2\mathcal{D}$ task. The resulting organization of the latent space improves, thus further establishing a correlation between the growing number of tasks and the improvement in performances.

	ITOP front-view								ITOP top-view									
Body part	RF[159]	RTW[206]	IEF[22]	VI [63]	REN9x6x6[62]	V2V[133]	A2J[200]	DECA-D3	RF[159]	RTW[206]	IEF[22]	VI [63]	REN9x6x6[62]	V2V[133]	A2J[200]	DECA-D1	DECA-D2	DECA-D3
Head	63.80	97.80	96.20	98.10	**98.70**	98.29	98.54	93.87	95.40	**98.40**	83.80	98.10	98.20	**98.40**	98.38	94.41	95.31	95.37
Neck	86.40	95.80	85.20	97.50	**99.40**	99.07	99.20	97.90	98.50	82.20	50.00	97.60	98.90	98.91	98.91	98.86	**99.16**	98.68
Shoulders	83.30	94.10	77.20	96.50	96.10	**97.18**	96.23	95.22	89.00	91.80	67.30	96.10	96.60	96.87	96.26	96.12	**97.51**	96.57
Elbows	73.20	77.90	45.40	73.30	74.70	80.42	78.92	**84.53**	57.40	80.10	40.20	**86.20**	74.40	79.16	75.88	76.86	81.67	84.07
Hands	51.30	**70.50**	30.90	68.70	55.20	67.26	68.35	56.49	49.10	76.90	39.00	**85.50**	50.70	62.44	59.35	44.41	45.97	54.33
Torso	65.00	93.80	84.70	85.60	98.70	98.73	98.52	**99.04**	80.50	68.20	30.50	72.90	98.10	97.78	97.82	99.46	**99.70**	99.46
Hip	50.80	80.30	83.50	72.00	91.80	93.23	90.85	**97.42**	20.00	55.70	38.90	61.20	85.50	86.91	86.88	97.84	**97.87**	97.42
Knees	65.70	68.80	81.80	69.00	89.00	91.80	90.75	**94.56**	2.60	53.90	54.00	51.60	70.00	83.28	79.66	88.01	88.19	**90.84**
Feet	61.30	68.40	80.90	60.80	81.10	87.60	86.91	**92.04**	0.00	28.70	62.40	51.50	41.60	69.62	58.34	79.30	**83.53**	81.88
Upper Body	-	-	-	**84.00**	-	-	-	83.03	-	-	-	**91.40**	-	-	-	78.51	80.60	83.00
Lower Body	-	-	-	67.30	-	-	-	**95.30**	-	-	-	54.70	-	-	-	89.96	91.27	**91.39**
Mean	65.80	80.50	71.00	77.40	84.90	88.74	88.00	**88.75**	47.40	68.20	51.20	75.50	75.50	83.44	80.5	83.85	85.58	**86.92**

Table 3.5: Comparison with the state-of the art for ITOP front-view and top-view (metric: 0.1m mAP).

Comparison with state-of-the-art methods

Depth data: ITOP dataset. We compare our DECA against common state-of-the-art method for human pose estimation on depth images [22, 62, 63, 133, 159, 200, 206]. The results are reported in Tab. 3.5. Our DECA outperforms existing methods on the front-view task, improving the accuracy by a wide margin on the more challenging top viewpoint. In general, we consistently perform better than other methods on most of the joints and the average. The gain of our method is particularly large when dealing with the lower body, which is often occluded in the top-view.

Depth data: Viewpoint-equivariant ITOP. We test DECA on the viewpoint transfer task, meaning training on one viewpoint, either top-view or front-view, and testing on the other one, unseen at training time. The comparison against available state-of-the-art methods [22, 63, 159, 206] are reported in Tab. 3.6. We consistently outperform other methods by a wide margin, thus making a step forward toward viewpoint equivariance. While other methods provide only the best subset of viewpoint transfer results (Tab. 3.6), omitting entirely the train on top and test on front scenario, we provide results for all the joints and all the viewpoint transfer combinations in Tab. 3.7. Our DECA achieves better results than the top-most of the other methods on many different joints (e.g. shoulders, lower body). In Tab 3.7, training DECA on top-view or front-view

ITOP Train on front, test on top					
Body part	RF [159]	RTW [206]	IEF [22]	VI [63]	DECA-D3
Head	48.10	1.50	47.90	55.60	46.27
Neck	5.90	8.10	39.00	40.90	**73.14**
Torso	4.70	3.90	41.90	35.00	**85.94**
Upper Body	19.70	2.20	23.90	29.40	**45.00**
Full Body	10.80	2.00	17.40	20.40	**51.85**

Table 3.6: Comparison with the state-of the art for the ITOP viewpoint transfer task (metric: 0.1m mAP). Training on front-view, validating on front-view, testing on top-view (top-view data is unseen in validation).

achieves comparable lower body accuracy. This means that when the network is trained on top view, where the lower body is mostly occluded, it can retrieve the occluded joints from previously unseen front views, and vice versa. This shows how our network has learned the viewpoint as a parameter, and it is thus able to generalize in a similar fashion in all the viewpoint transfer combinations.

RGB data: Viewpoint-equivariant PanopTOP31K. To the best of our knowledge, we are the first to tackle the problem of viewpoint transfer between top-view and front-view in the RGB domain. We report results with training and testing on both seen and unseen viewpoints in Tab. 3.8. The chosen metric is the mean per-joint projection error (MPJPE). We report results with and without the Procrustes alignment [59] of the predicted poses. It is interesting to notice how DECA can reduce the gap between the same viewpoint results and the results of the viewpoint transfer tasks. In the case of viewpoint transfer, we train on viewpoint A, validate on the same viewpoint A and test on viewpoint B.

Qualitative results

In Fig. 3.23 we show some qualitative results from DECA-R4 configuration on RGB data. We deploy our network training and testing on all the possible

	DECA-D3	
Body part	Train on front, test on top	Train on top, test on front
Head	46.27	18.51
Neck	73.14	44.77
Shoulders	69.02	25.18
Elbows	43.87	16.23
Hands	9.41	2.19
Torso	85.94	68.63
Hip	72.15	64.75
Knees	49.31	68.15
Feet	42.46	46.12
Upper Body	45.00	18.81
Lower Body	59.11	60.95
Mean	51.85	38.48

Table 3.7: DECA-D3 complete results for the ITOP viewpoint transfer tasks (metric: 0.1m mAP). Test data is unseen during validation for both the cases.

	DECA-R4							
	Train on front, test on front		**Train on top, test on top**		**Train on front, test on top**		**Train on top, test on front**	
Body part	No Procrustes	Procrustes	No Procrustes	Procrustes	No Procrustes	Procrustes	No Procrustes	Procrustes
Neck	4.02	2.37	4.55	2.51	16.02	4.16	8.21	5.06
Nose	5.66	3.75	6.98	3.89	16.83	7.67	10.72	6.76
Body Center	0.56	4.63	1.23	3.63	1.01	31.20	0.83	11.59
Shoulders	4.56	2.76	5.14	3.07	17.43	5.33	8.51	5.35
Elbows	9.82	7.14	9.64	7.51	29.70	18.52	23.20	15.47
Hands	13.88	10.82	14.02	12.34	47.01	38.29	36.78	28.25
Hips	18.75	4.87	2.71	3.89	5.10	30.07	3.64	10.88
Knees	9.54	5.14	7.59	4.84	52.98	28.65	20.11	9.28
Feet	11.53	5.08	9.83	5.10	69.18	28.75	26.36	11.07
Eyes	6.19	4.00	7.44	3.79	19.33	11.00	11.40	7.45
Ears	5.50	3.73	7.15	3.74	23.56	13.00	11.22	7.16
Upper Body	6.93	5.21	7.66	5.46	23.69	16.56	15.54	11.60
Lower Body	7.65	5.03	6.71	4.61	42.42	29.16	16.71	10.41
Mean	7.16	5.15	7.36	5.19	29.60	20.54	15.91	11.22

Table 3.8: DECA-R4 results on the PanopTOP31K RGB dataset, with and without the Procrustes transformation [59] (metric: MPJPE). Tasks: (i) 3D pose estimation from the front and top viewpoints (ii) viewpoint transfer for both front and top views. Test data is unseen during validation for both the viewpoint transfer tasks.

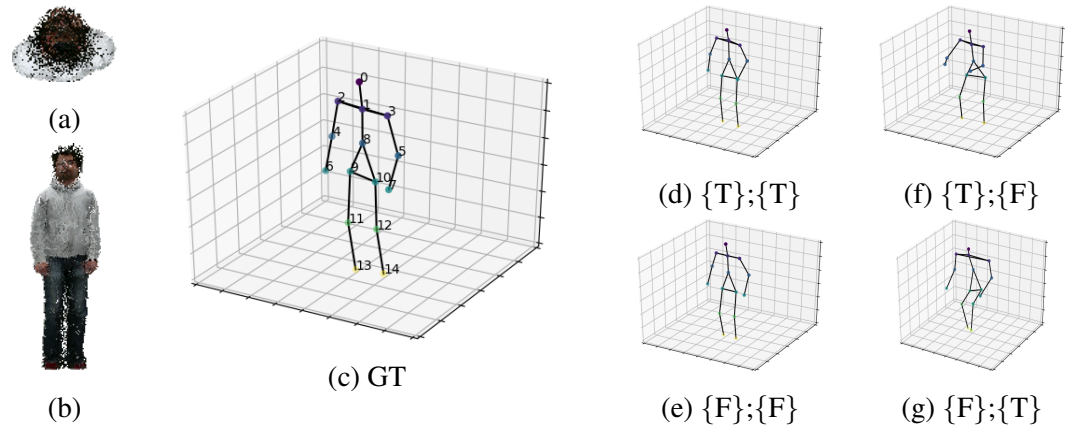
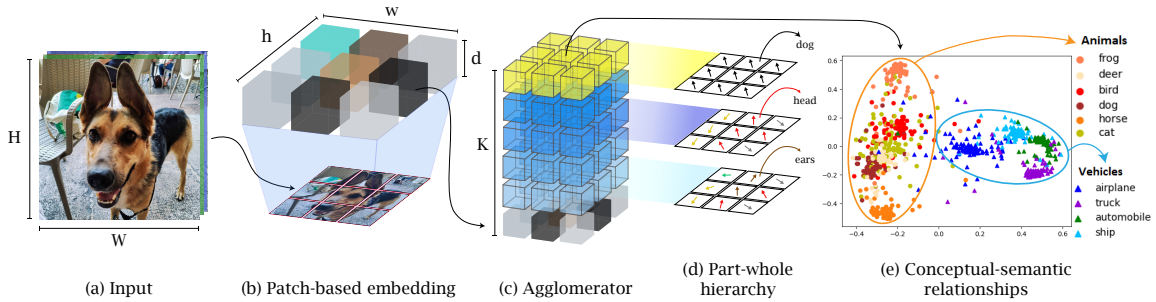


Figure 3.23: DECA-R4 qualitative results on the PanopTOP31K dataset. On the left (a, b) the types of input accepted by DECA (top-view or front-view). DECA can also accept inputs in the depth domain. In the center (c), the corresponding 3D ground truth. On the right, the possible combinations of training/testing experiments. **T** stands for **top** and **F** stands for **front**. As an example, in (f), $\{T\};\{F\}$ means that DECA has been trained exclusively on **top** data and tested on previously unseen (not even at validation time) **front** data.

viewpoint combinations. The network takes as input either the top-view RGB (Fig. 3.23a) image or the front view (Fig. 3.23b) one. When trained and tested on the same viewpoint (Fig. 3.23d, 3.23e), the network produces similar outputs, thus confirming its ability to deal with the challenging top-view scenario. When training on the top view and testing on the front one (Fig. 3.23f), the network can accurately retrieve the positions of the lower body joints. DECA can retrieve parts of the body mostly occluded ad training time, thus displaying its generalization capabilities. When training on the front view and testing on the top one (Fig. 3.23g), the network can retrieve the positions of the upper body joints, which are visible in both images but from different perspectives, proving that DECA can internally model the viewpoint.

3.4.4 Discussion

We presented DECA, a deep viewpoint-equivariant method for human pose estimation on single RGB/depth images using capsule autoencoders. We show



(a) Input (b) Patch-based embedding (c) Agglomerator (d) Part-whole hierarchy (e) Conceptual-semantic relationships

Figure 3.24: **[Better seen in color]**. Overview of the proposed solution. Our Agglomerator is a novel architecture for vision applications, in which column structure (c) mimics hyper-columns typical of the human visual cortex [65]. The input data (a) is fed to the columns using a patch-based embedding (b). The Agglomerator architecture iteratively routes the information across its structure, creating a neural representation of each image, similar to neural fields [127]. In the neural representation, *part-whole* hierarchies (d) emerge at different levels of the columns. The same column can represent the same patch of the image with different levels of abstraction (e.g., the ears, the head, and the dog) corresponding to each level in the column. Neighbor columns agree on a *part* representation (e.g ears, head) at lower levels, ideally representing the same *whole* (e.g. dog) at the top level. The resulting feature space represents the *conceptual-semantic relationships* between data (e) resembling the human hierarchical organization [128]. Samples belonging to the same super-class (e.g., animals, vehicles) are clustered together, with conceptually close categories (e.g., birds and airplanes) represented on the edge of the super-classes.

how CapsNets are better suited to deal with the 3D nature of raw data and how they allow taking a step forward to viewpoint equivariance. We have shown how our method can effectively generalize and achieve state-of-the-art results in both RGB and depth domains, as well as in the viewpoint transfer task.

3.5 Agglomerator

The extensive adoption of neural networks and, more in general, learning models has been raising concerns regarding our chances, as humans, to explain their behavior. Interpretability would be a highly desirable feature for neural networks, especially in those applications like autonomous driving [60], healthcare

[130], and finance [156], where safety, life, and security are at stake.

Deep neural networks have achieved superhuman performances in many domains, from computer vision [67, 105] to natural language processing [41, 190], and data analysis [156]. However, the achieved performances have come at the expense of model complexity, making it difficult to interpret how neural networks work [113]. These neural networks are usually deployed as "black boxes", with millions of parameters to be tuned, mostly according to experience and rule of thumb. Interpreting how a trainable parameter in the network setup directly affects the desired output from a given input has nearly zero chances.

According to the literature, interpretability is defined as *"the degree to which a human can understand the cause of a decision"*[129]. When a machine learning model reaches high accuracy on a task such as classification and prediction, can we trust the model without understanding why such a decision has been taken? The decision process is complex and we tend to evaluate the performance of a system in solving a given task using metrics computed at the end of the processing chain. While single metrics, such as the classification accuracy, reach super-human results, they provide an incomplete description of the real-world task [43]. For example, when dealing with an image classification problem, the learning model can tell the class the represented object belongs to. Thus, we know **what** has been predicted by the network, but we have little understanding about **why** we have obtained such prediction [132]. As humans, when looking at an object that has eyes and limbs, we can infer via reasoning and intuition that these are elements (parts) that belong to the same entity (whole) [13], say an animal, and we can explain and motivate why such decision is taken, generally based on past experiences, beliefs and attitude [2]. Moreover, even in presence of an animal never seen before, we can probably tell from the visual features, our frames of reference [65] and our hierarchical organization of objects in the world [128] whether it is a fish or a mammal. We would like neural networks to display the same behavior, so that objects that are *close* in

the conceptual-semantic and lexical relations are adjacent in the feature space as well (as shown in Fig. 3.24e). By doing so, it would be intuitive to identify hierarchical relations between samples and how the model has learned to build a topology describing each sample. Consequently, we can agree on the definition of interpretability in deep learning as the *"extraction of relevant knowledge from a machine-learning model concerning relationships either contained in data or learned by the model"* [135].

In the image classification field, available techniques, such as transformers [41, 44, 190], neural fields [127], contrastive learning representation [27], distillation [73] and capsules [152], have achieved state-of-the-art performances, introducing a number of novelties, such as powerful attention-based features and per-patch analysis, positional encoding, similarity-based self-supervised pre-training, model compression and deep modeling of part-whole relationships. Taken as standalone, these methods have contributed improving the interpretability of networks, while still lacking to directly emphasize either data relationships [27, 41, 44, 127, 190] (e.g. conceptual-semantic relationships) or model-learned relationships [73, 152] (e.g. part-whole relationships).
In [72], a concept idea on how to represent part-whole hierarchies in neural networks is introduced, which attempts to merge the advantages of the above state-of-the-art frameworks into a single theoretical system (known as *GLOM*). GLOM aims at mimicking the human ability in learning to parse visual scenes. Inspired by the theoretical concepts described in [65, 72], we build a working system, called Agglomerator, which achieves part-whole agreement [74] at different levels of the model (*relationships learned by the model*) and hierarchical organization of the feature space (*relationships contained in data*), as shown in Fig. 3.24.

Our contribution is summarised as follows:

- we introduce a novel model, called Agglomerator[2], mimicking the func-

[2]The code and the pre-trained models can be found at **[see supplementary materials]**.

tioning of the cortical columns in the human brain [66];

- we explain how our architecture provides interpretability of *relationships learned by the model*, specifically part-whole relationships;
- we show how our architecture provides interpretability of *relationships contained in data*, namely the hierarchical organization of the feature space;
- we provide results outperforming or on par with current methods on multiple common datasets, such as SmallNORB [107], MNIST [106], FashionMNIST [199], CIFAR-10 and CIFAR-100 [103];
- we show that our model relies on fewer parameters and can generalize to multiple datasets.

3.5.1 Method

The framework we propose aims at replicating the column-like pattern, similar to hyper-columns typical of the human visual cortex [65]. An overview is shown in Fig. 3.24.

Agglomerator brings together concepts and building blocks from multiple methods, such as CNNs [110], transformers [41, 44, 190], neural fields [127], contrastive learning representation [27], distillation [73], and capsules [152]. In the next paragraph, we introduce the mathematical notation needed to explain the details of the main building blocks of the architecture.

Each input image is transformed into a feature map divided into $N = h \times w$ patches. The n-th patch, with $n \in \{1,\dots,N\}$ is fed to the corresponding column $C_n(h,w)$, spatially located at coordinates (h,w). The subscript n is omitted in the next equations for better readability. As shown in Fig. 3.25, each column $C(h,w)$ consists of K embedding levels $\{l_t^{(h,w),k} \mid k = 0,\dots,K\}$ connected by a stack of auto-encoders at location (h,w) at time $t \in \{0,\dots,t-1,t,t+1,\dots,T\}$. The superscript (h,w) is omitted in the next instances of l_t^k for better readability.

Each level l_t^k of the column is an embedding vector representation of size d. Levels l_t^{k-1} and l_t^k represent consecutive levels; l_t^{k-1} represents a *part* of the *whole* l_t^k. We indicate as $l_t^k \in L_t^k$ all the levels l_t^k in all columns $C(h,w)$ sharing the same k value and belonging to the same layer L_t^k. Being K the last layer of our architecture at the last time step T, it is represented as L_T^K.

Patches embedding

At the embedding stage, as in [110], we apply a convolutional Tokenizer to extract the feature map of each image of size $H \times W$ pixels, which provides a richer representation compared to the original image. Following the implementation in [110], the obtained feature map has size $h \times w \times d$ where $h = H/4$ and $w = W/4$. We then embed each of the n d-dimensional embedding vectors into the bottom levels $l_t^0 \in L_t^0$ at the corresponding coordinates (h,w) of the corresponding column $C(h,w)$. Feeding the n-th each patch to a spatially located column $C(h,w)$ resembles the positional encoding of neural fields [127], where each d-sized embedding l_t^k represents at the same time the sample and its relative observation viewpoint. At each time step t, we embed each image sample into the first layer of the columns, which is represented as the bottom layer L_t^0.

Hypercolumns

Consecutive levels in time and space in a column $C(h,w)$ are connected by an auto-encoder. The auto-encoders are based on an MLP, which allows for model reduction [73] and faster training time. Each auto-encoder computes the top-down contribution of a level l_{t-1}^{k+1} to the value of the level below at the next time step l_t^k using a $N_{TD}(l_{t-1}^{k+1})$ top-down decoder. Similarly, each auto-encoder computes the bottom-up contribution of a level l_{t-1}^{k-1} to the value of the level above at the next time step l_t^k using a $N_{BU}(l_{t-1}^{k-1})$ bottom-up encoder. $N_{TD}(l_{t-1}^{k+1})$ and $N_{BU}(l_{t-1}^{k-1})$ share a similar structure, but for the activation functions, as described in Fig. 3.25(e). The top-down network uses GELU activation functions

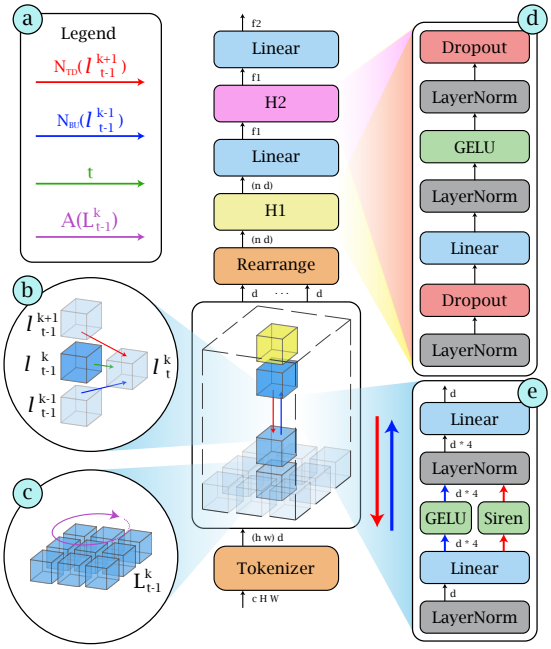


Figure 3.25: **[Better seen in color]**. Architecture of our Agglomerator model (center) with information routing (left) and detailed structure of building elements (right). Each *cube* represents a level l_t^k. **Left:** (a) legend of the arrows in the figure, representing the top-down network $N_{TD}(l_{t-1}^{k+1})$, the bottom-up network $N_{BU}(l_{t-1}^{k-1})$, attention mechanism $A(L_{t-1}^k)$ and time step t. (b) Contribution to the value of level l_t^k given by l_{t-1}^k, $N_{TD}(l_{t-1}^{k+1})$ and $N_{BU}(l_{t-1}^{k-1})$. (c) The attention mechanisms $A(L_{t-1}^k)$ share information between $l_{t-1}^k \in L_{t-1}^k$. **Center:** bottom to top, the architecture consists of the Tokenizer module, followed by the columns $C(h,w)$, with each level l_t^k connected to the neighbors with $N_{TD}(l_{t-1}^{k+1})$ and $N_{BU}(l_{t-1}^{k-1})$. On top of the structure, the contrastive $H1$ and cross entropy $H2$ heads. **Right:** (d) structure of heads $H1$ and $H2$. (e) Structure of the top-down network $N_{TD}(l_{t-1}^{k+1})$ and the bottom-up network $N_{BU}(l_{t-1}^{k-1})$.

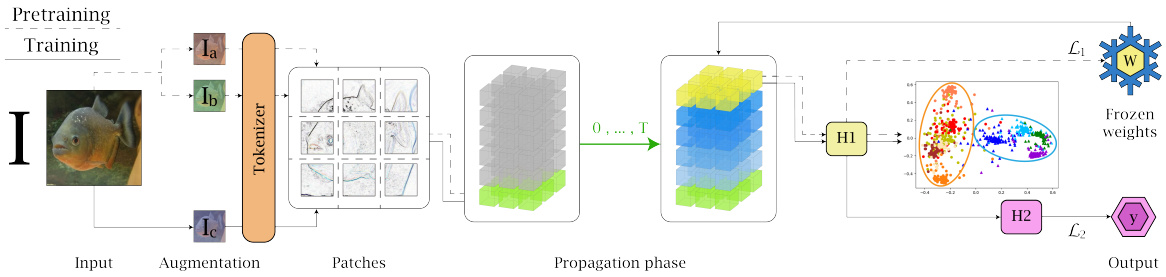


Figure 3.26: **Contrastive pre-training (dashed lines)** and **supervised training (continuous lines)** procedures. During the contrastive pre-training, two images I_a and I_b are produced by applying random data augmentation to the input image I. Through the Tokenizer, we compute feature maps for both I_a and I_b, which are then divided in patches and embedded into the bottom layer of the columns L_t^0. During the *propagation phase*, the information is routed through the Agglomerator architecture to obtain the neural representation L_T^K for each sample. We pre-train the network with the contrastive head $H1$ using a supervised contrastive loss $\mathcal{L}_1$, obtaining weights W. During the supervised training, we first load the frozen weights W in the network. Then, augmentation RandAugment [35] is applied on the input image I to obtain I_c, which follows the same steps as the pre-training phase. The network, with the classification head $H2$, is trained for the classification task by minimizing the cross-entropy loss $\mathcal{L}_2$.

[69], while the bottom up network relies on Siren activation functions [164]. All the $N_{TD}(l_{t-1}^{k+1})$ connecting L_{t-1}^{k+1} to layer L_t^k share the same weights. The same is true for the $N_{BU}(l_{t-1}^{k-1})$ connecting L_{t-1}^{k-1} to layer L_t^k.

Routing

The key element of our architecture is how the information is routed to obtain a representation of the input data where the part-whole hierarchies emerge.

Before computing the loss, we need to iteratively propagate each batch N through the network, obtaining a deep representation of each image. This procedure, *propagation phase*, encourages the network to reach *consensus* between neighbor levels $l_t^k \in L_t^k$. Ideally, this means that all neighbor levels in the last layer $l_t^K \in L_t^K$ should have similar values, representing the same *whole*; neighbor levels at bottom layers $l_t^k \in L_t^k | k \neq K$ should instead share the value among smaller groups, each group representing the same *part*. Group of vectors that

"agree" on a similar value have reached the *consensus* on the image representation at that level, and they are called *islands of agreement*. An example of such representation is shown in Fig. 3.24(d). In capsules-based approaches [152], group of neurons are activated to represent the part-whole hierarchy with limited expressive power. Our d-dimensional layers l_t^k provide a richer representation of the same hierarchy.

To obtain such representation, at time step $t = 0$, we randomly initialise all the values l_0^k and we embed a batch of B samples into the bottom layer L_0^0. Once the values are initialized, we compute the attention $A(L_t^k)$. Instead of the self-attention mechanism used in Transformers [41, 44, 190], a standard attention weighting is deployed as in [201]. Each attention weight ω_n is computed as

$$\omega_n = \frac{e^{\beta \lambda_n \cdot l_t^k}}{\sum e^{\beta \lambda_n \cdot N(\lambda_n)}} \tag{3.16}$$

where λ_n represents each possible level l_t^k belonging to the same layer L_t^k as l_t^k, $N(\lambda_n)$ is an indicator function which indexes all the neighbors levels of λ_n belonging to the same layer L_t^k and β is a parameter that determines the sharpness of the attention.

At each time step $t \mid t \in \{1, \ldots, T\}$, a batch with B samples is fed to the bottom layer L_t^0 network as described in Sec. 3.5.1. We compute the values l_t^k as

$$\begin{aligned} l_t^k = avg(&\omega_l l_{t-1}^k, \omega_{BU} N_{BU}(l_{t-1}^{k-1}), \\ &\omega_{TD} N_{TD}(l_{t-1}^{k+1}), \omega_A A(L_{t-1}^k)) \end{aligned} \tag{3.17}$$

where $avg()$ indicates the arithmetical average, and $\omega_l, \omega_{BU}, \omega_{TD}, \omega_A$ are trainable weights. For layer L_t^K, contribution $N_{TD}(l_{t-1}^{k+1})$ is not included, as L_t^{K+1} does not exist. The *propagation phase* takes T time steps to reach the final representation of each image at each layer L_k^T.

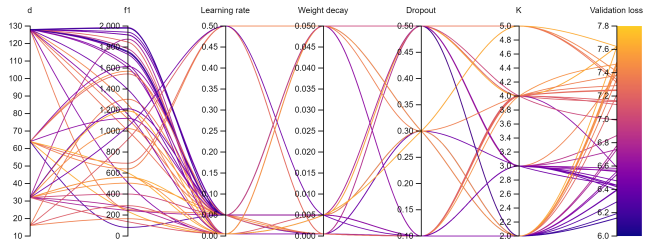


Figure 3.27: Hyper-parameters sweep. Each line represents a combination of parameters setup, with the darker lines representing the models achieving the lowest validation loss. Image obtained with [14].

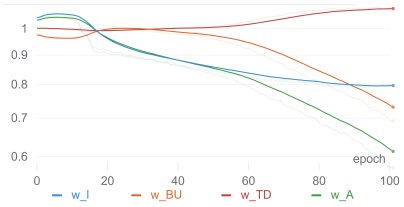


Figure 3.28: Each line represents the variation of weights $\omega_I, \omega_{BU}, \omega_{TD}, \omega_A$ across epochs. Image obtained with [14].

Training

The training procedure of our architecture is shown in Fig. 3.26. It is divided in two steps: (i) a pre-training phase using a supervised contrastive loss function [27] and (ii) a training phase for the image classification using a Cross-Entropy loss.

We first pre-train our network using an image-based contrastive loss [27]. Given a batch with B samples, we duplicate each image I to obtain pairs of samples (I_a, I_b), for a total of $2B$ data points. We then apply data augmentation RandAugment [35] to both (I_a, I_b). Both samples are fed to the network as described in Sec. 3.5.1, and we perform the propagation phase in Sec. 3.5.1 to obtain the representation at the last layer L_T^K. Then we rearrange the n levels $l_T^K \in L_T^K$ to obtain a vector of dimensions $n \times d$, given as input to the contrastive head $H1$, as described in Fig. 3.25. At the output of the contrastive head, each sample is described by a feature vector of dimension $f1$. We take all the possible sample pairs (I_a, I_b) from the batch and we compute the contrastive loss defined as:

$$\mathcal{L}_1 = ContrLoss(I_a, I_b) = -log\frac{e^{sim(I_a,I_b)}}{\sum_{k=1}^{2B}\mathcal{I}_{[k\neq a]}e^{sim(I_a,I_b)}} \tag{3.18}$$

where $sim(u,v) = u^T v / \|u\| \|v\|$ indicates the dot product between the normalised version of u and v, $\mathcal{I}_{[k \neq a]}$ is a indicator function valued 0 if k and a belong to the same class, and 1 otherwise.

Once the network is pre-trained using the contrastive loss, the weights are frozen. We apply augmentation [35] to each sample I_c in a batch of size B, which is then fed to the network for the *propagation phase* to obtain for each sample the representation L_T^K. Then, the cross-entropy head $H2$ is added on top of the contrastive head $H1$. A linear layer resizes $f1$-dimensional features to dimension $f2$, which corresponds to the number of classes to be predicted for each dataset. The new layers are then trained using the cross-entropy function:

$$\mathcal{L}_2 = CE(y, \hat{y}) = -\frac{1}{f2} \sum_{i=1}^{\text{f2}} y_i \log(\hat{y}_i) \tag{3.19}$$

where y is the label of a sample taken from the batch and $\hat{y}$ is the label to be predicted.

3.5.2 Experiments

We perform our experiments on the following datasets:

SmallNorb (S-NORB) [107] is a dataset for 3D object recognition from shape. It consists of roughly 200000 images of size 96×96 pixels of 5 classes of toys.

MNIST [106] and **FashionMNIST** [199], consist of 60000 training images and 10000 test images of grayscale handwritten digits and Zalando's articles of size 28×28 pixels.

CIFAR-10 and **CIFAR-100** [103] both consist of 50000 training images and 10000 test images of size 32×32 pixels, with 10 and 100 classes, respectively.

Our network is trained in an end-to-end fashion using PyTorch Lightning on a single NVIDIA GeForce RTX 3090. Input images for each dataset are

Method	Ref	Backbone	Error %					# of params (Millions)	Training Arch.
			S-Norb	MNIST	F-MNIST	C-10	C-100		
E-CapsNet	[124]	Caps	2.54	0.26	-	-	-	0.2	GPU
CapsNet	[134, 152]		2.70	0.25	6.38	10.6	82.00	6.8	GPU
Matrix-CapsNet	[76]		1.40	0.44	6.14	11.9	-	0.3	GPU
Capsule VB	[150]		1.60	0.30	5.20	11.2	-	0.2	GPU
ResNet-110	[9, 67, 79]	Conv	-	2.10	5.10	6.41*	27.76*	1.7	GPU
VGG	[9, 163]		-	0.32	6.50	7.74*	28.05*	20	GPU
ViT-L/16	[44]	Transf	-	-	-	0.85*	6.75*	632	TPU
ConvMLP-L	[110]	Conv/MLP	-	-	-	1.40*	11.40*	43	TPU
MLP-Mixer-L/16	[181]	MLP	-	-	-	1.66*	-	207	TPU
Ours		Conv/MLP/Caps	0.01	0.30	7.43	11.15	40.97	72	GPU

Table 3.9: Error percentages on the Top-1 accuracy results on datasets SmallNorb (S-Norb), MNIST, FashionMNIST (F-MNIST), CIFAR-10 (C-10), and CIFAR-100 (C-100). The $*$ notation indicates results obtained with networks pre-trained on ImageNet.

normalized using each standard dataset's normalization. We train our network on each dataset's native resolution, except for SmallNorb, which is resized to 32×32 pixels, following the standard procedure as in [76, 150]. The Tokenizer embedding creates $n = H/4 \times W/4$ patches, thus the corresponding number of columns is 8×8 for CIFAR-10, CIFAR-100, and SmallNorb, and 7×7 for MNIST FashionMNIST. During the pre-training, we deploy the following hyper-parameters: 300 epochs, cyclic learning rate [167] in the range $[0.002, 0.05]$, batch size $B = 1024$, levels embedding $d = 128$, number of levels $K = 3$, number of iterations $T = 2K = 6$, dropout value 0.3, contrastive features dimension $f1 = 512$, and weight decay $5e^{-4}$. During the training phase, we resume the network training with the same hyper-parameters, $f2$ being the number of classes corresponding to each dataset.

3.5.3 Quantitative results

We report the quantitative results for each dataset in Tab. 3.9. Capsule-based models [76, 124, 134, 150, 152] can achieve good performances on simple datasets (SmallNorb, MNIST, and FashionMNIST), but they fail to general-

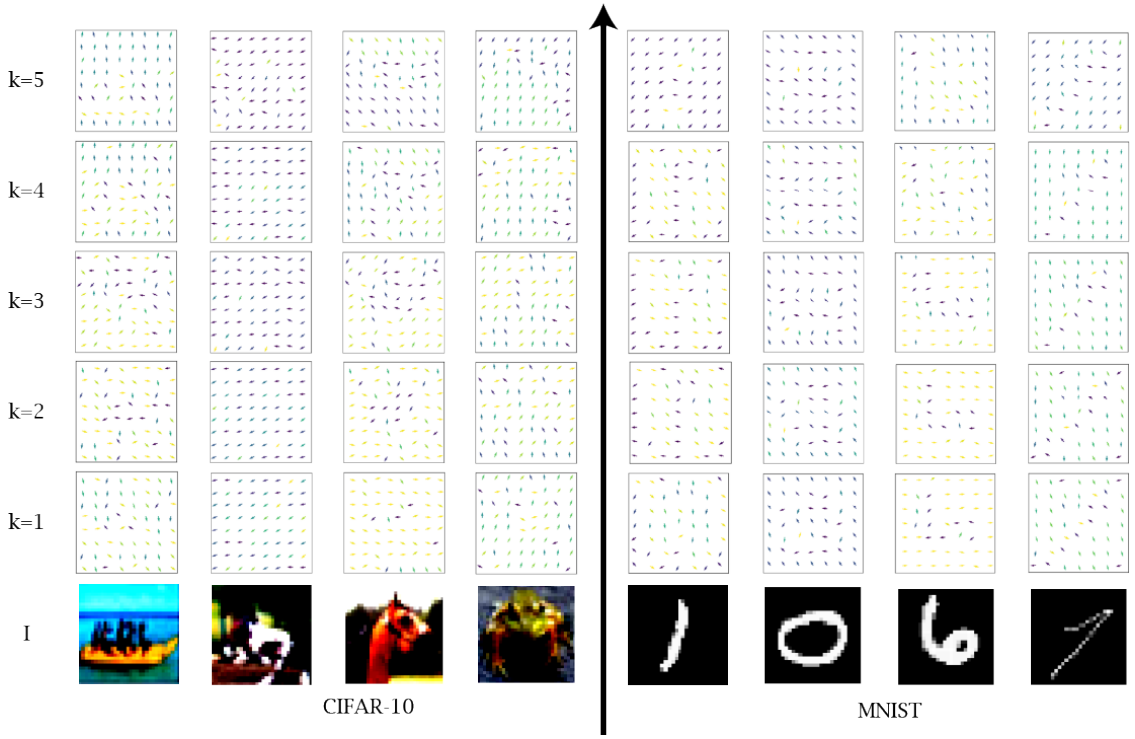
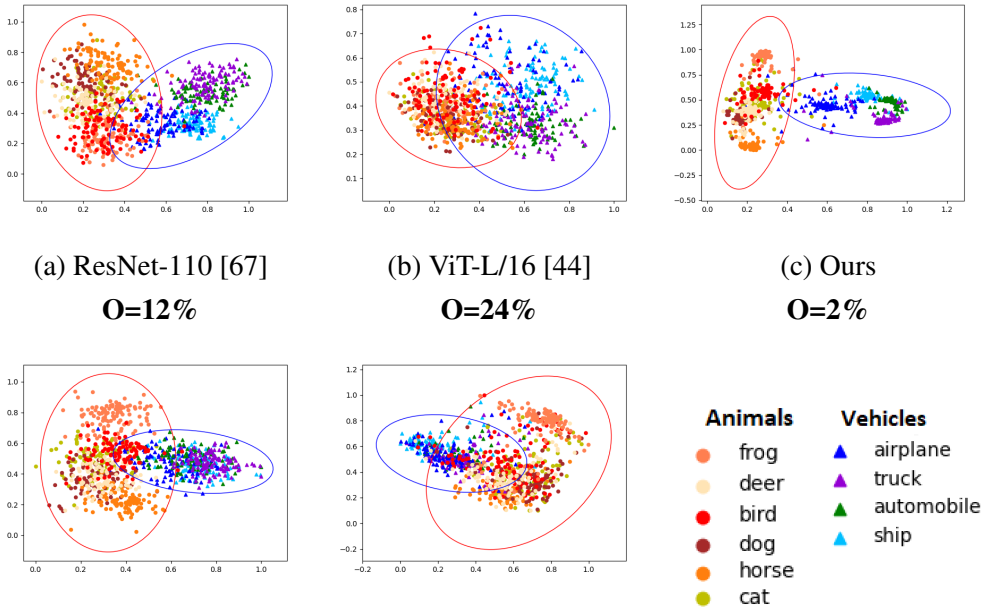


Figure 3.29: Vectorial representation of emerging islands of agreement at different K levels of sample from MNIST and CIFAR-10 datasets.

ize to datasets with a higher number of classes (CIFAR-100). Convolutional-based models [9, 67, 79, 163] can generalize to different datasets, at the expense of weak model interpretability, mainly due to the max-pooling operation. Transformer-based [44] and MLP-based methods [110, 181] are able to achieve the best performances on more complex datasets, but they do not provide tests for smaller datasets. However, to achieve such levels of accuracy they rely on long pretraining (thousands of TPU days) on expensive computational architectures, implementing data augmentation on ImageNet [104] or the JFT-300M [172] dataset, not available publicly. As can be seen, our method performs on par with capsule-based methods on simpler datasets, while achieving better generalization on more complex ones. In addition, our method has fewer parameters than most transformer-based and MLP-based methods, and it can be trained in less time on a much smaller architecture.

Ablation study. We analyze the contribution of the different components of our architecture evaluating their influence on the validation loss. The considered parameters, in descending order of correlation with the validation loss value are: the embedding dimension d, the contrastive feature vector $f1$, learning rate, weight decay, dropout, and the number of levels K. The results are reported in Fig. 3.27. We perform 50 different training on CIFAR-10 with different combinations of parameters. We monitor the validation loss value after 50 epochs.

In Fig. 3.28, we show how the contributions to the values of levels l_t^k in Eq. 3.17 are weighted over the first 100 epochs by observing how the trainable weights $\omega_l, \omega_{BU}, \omega_{TD}, \omega_A$ change. The attention contribution ω_A and the previous level one ω_l decrease over time, with ω_l reaching a plateau. After 15 epochs of weights self-calibration, the top-down contribution ω_{TD} increases over the epochs, as the network can infer useful information from predicted "wholes" at the upper levels. The bottom-up contribution ω_{BU} decreases over the epochs since the bottom-up networks have learned to map efficiently map



(a) ResNet-110 [67] **O=12%**

(b) ViT-L/16 [44] **O=24%**

(c) Ours **O=2%**

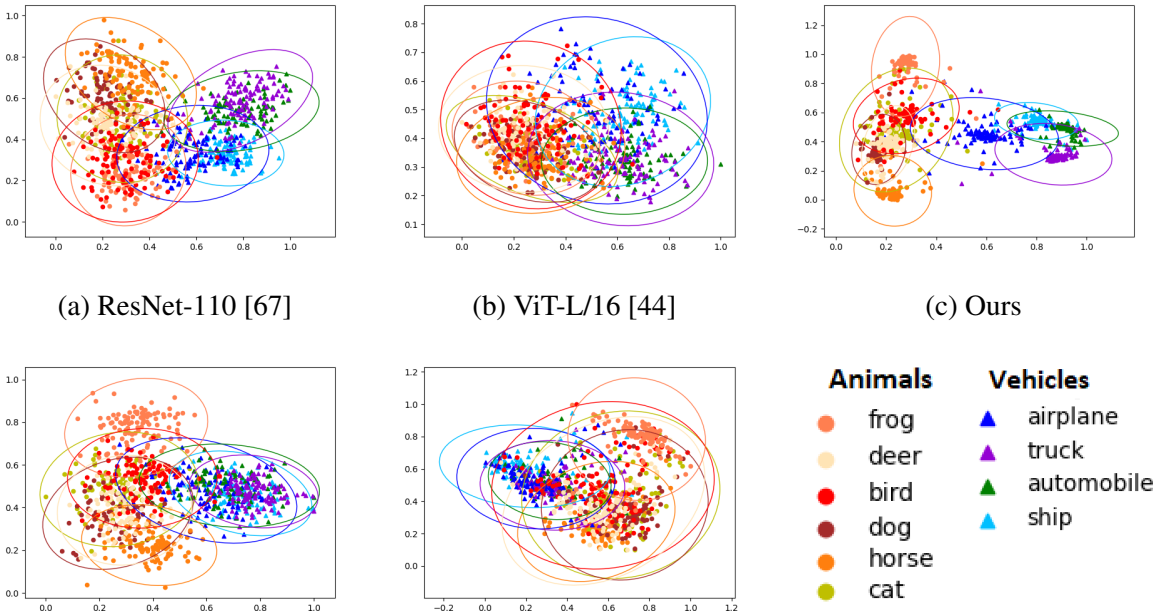
(d) ConvMLP-L [110] **O=12%**

(e) Matrix-CapsNet [76] **O=20%**

(f) Legend

Figure 3.30: 2D representation of the latent space for multiple methods trained only on the CIFAR-10 dataset obtained using Principal Component Analysis (PCA) [196]. The PCA provides a deterministic change of base for the data from a multidimensional space into a 2D space. The legend (f) displays the classes, which are divided between super-classes *Vehicles* and *Animals* following the WordNet hierarchy [128]. The different methods (a,b,c,d,e) are all able to cluster the samples between the two super-classes. However, while (a,b,e) display a latent space where classes are close to each other, the two MLP-based methods (c,d) are able to provide a clearer separation between the super-classes. Both methods show conceptual-semantically close samples on the edge of each superclass, such as airplanes and birds. Inside each superclass, semantically close samples are represented contiguously, such as deers and horses, or cars and trucks. Our method (c) provides better inter-class and intra-class separability. The overlap percentage O is reported for each method. The overlap area is the area where a mistake with a higher hierarchical severity [11] has a higher probability to occur.

the input into the embeddings, especially at lower levels.



(a) ResNet-110 [67] (b) ViT-L/16 [44] (c) Ours

(d) ConvMLP-L [110] (e) Matrix-CapsNet [76] (f) Legend

Figure 3.31: 2D representation of the latent space for multiple methods trained only on the CIFAR-10 dataset obtained using Principal Component Analysis (PCA) [196]. The PCA provides a deterministic change of base for the data from a multidimensional space into a 2D space. The legend (f) displays the classes, which are divided between super-classes *Vehicles* and *Animals* following the WordNet hierarchy [128]. The different methods (a,b,c,d,e) are all able to cluster the samples. However, while (a,b,e) display a latent space where classes are close to each other, the two MLP-based methods (c,d) are able to provide a clearer separation between classes. Both methods show conceptual-semantically close samples on the edge of each superclass, such as airplanes and birds. Inside each superclass, semantically close samples are represented contiguously, such as deers and horses, or cars and trucks. Our method (c) provides better inter-class and intra-class separability. We provide numerical results of the classes overlap in Fig. 3.32.

3.5.4 Qualitative results: interpretability

Our method provides interpretability of the *relationships learned by the model* by explicitly modeling the part-whole hierarchy, and of the *relationships con-*

ResNet-110	Frog	Deer	Bird	Dog	Horse	Cat	Airplane	Truck	Automobile	Ship
Frog	100	32	70	12	3	30	24	0	0	3
Deer	32	100	42	50	22	73	14	1	1	0
Bird	70	42	100	18	7	39	27	3	3	6
Dog	12	50	18	100	39	52	5	0	0	0
Horse	3	22	7	39	100	33	4	4	4	0
Cat	30	73	39	52	33	100	17	3	4	2
Airplane	24	14	27	5	4	17	100	19	21	44
Truck	0	1	3	0	4	3	19	100	70	15
Automobile	0	1	3	0	4	4	21	70	100	18
Ship	3	0	6	0	0	2	44	15	18	100

(a) ResNet-110 [67]

ViT-L/16	Frog	Deer	Bird	Dog	Horse	Cat	Airplane	Truck	Automobile	Ship
Frog	100	75	55	73	59	73	25	18	13	14
Deer	75	100	71	56	53	58	30	18	13	18
Bird	55	71	100	48	56	52	39	22	16	22
Dog	73	56	48	100	65	82	20	19	16	11
Horse	59	53	56	65	100	65	27	28	24	20
Cat	73	58	52	82	65	100	24	23	20	14
Airplane	25	30	39	20	27	24	100	42	35	60
Truck	18	18	22	19	28	23	42	100	71	41
Automobile	13	13	16	16	24	20	35	71	100	35
Ship	14	18	22	11	20	14	60	41	35	100

(b) ViT-L/16 [44]

Ours	Frog	Deer	Bird	Dog	Horse	Cat	Airplane	Truck	Automobile	Ship
Frog	100	2	17	0	0	17	0	0	0	0
Deer	2	100	26	63	14	29	0	0	0	0
Bird	17	26	100	19	0	54	12	0	0	0
Dog	0	63	19	100	14	27	0	0	0	0
Horse	0	14	0	14	100	14	0	0	0	0
Cat	17	29	54	27	14	100	10	0	0	0
Airplane	0	0	12	0	0	10	100	18	11	16
Truck	0	0	0	0	0	0	18	100	19	11
Automobile	0	0	0	0	0	0	11	19	100	44
Ship	0	0	0	0	0	0	16	11	44	100

(c) Ours

ConvMLP-L	Frog	Deer	Bird	Dog	Horse	Cat	Airplane	Truck	Automobile	Ship
Frog	100	6	23	5	0	18	9	0	3	0
Deer	6	100	42	85	33	58	18	0	9	3
Bird	23	42	100	39	8	62	35	5	21	11
Dog	5	85	39	100	30	57	17	0	9	3
Horse	0	33	8	30	100	16	6	0	1	0
Cat	18	58	62	57	16	100	23	1	13	5
Airplane	9	18	35	17	6	23	100	38	61	53
Truck	0	0	5	0	0	1	38	100	59	70
Automobile	3	9	21	9	1	13	61	59	100	66
Ship	0	3	11	3	0	5	53	70	66	100

(d) ConvMLP-L [110]

Matrix-CapsNet	Frog	Deer	Bird	Dog	Horse	Cat	Airplane	Truck	Automobile	Ship
Frog	100	21	5	15	25	50	19	60	73	18
Deer	21	100	28	66	74	16	77	24	28	57
Bird	5	28	100	25	33	6	31	6	8	10
Dog	15	66	25	100	58	10	75	16	21	54
Horse	25	74	33	58	100	22	76	31	32	44
Cat	50	16	6	10	22	100	15	69	49	9
Airplane	19	77	31	75	76	15	100	21	25	48
Truck	60	24	6	16	31	69	21	100	70	18
Automobile	73	28	8	21	32	49	25	70	100	25
Ship	18	57	10	54	44	9	48	18	25	100

(e) Matrix-CapsNet [76]

Figure 3.32: The overlap percentage O between classes in the latent space is reported for each possible class and each method. For each table, the top-left quadrant represents the overlap percentage between classes belonging to the super-class *animals*, while the bottom-right one for the superclass *vehicles*. The top-right and bottom-left quadrants represent the area where a mistake with a higher hierarchical severity [11] is possible. It would be ideal to have the table with zeros for all the values, but the diagonal. That would represent perfect separation between all the classes. Our method provides the best separation between the two superclasses. It is interesting to notice that the highest intra-superclasses correlation is present between the two classes *bird* and *airplanes* which share features like the wings and the ability to fly.

tained in data through the hierarchical organization of the feature space.

Island of agreement as a representation of multi-level part-whole hierarchy. During the *propagation phase*, neighbor levels on the same layer L_t^k are encouraged to reach a *consensus* by forming *islands of agreement*. The *islands of agreement* represent the part-whole hierarchies at different levels. In Fig. 3.29, we provide a few examples of the islands of agreement obtained on MNIST and CIFAR-10 trained with $K = 5$ levels. Each arrow represents the value of a level l_k^t at location (h, w), reduced from d-dimensional to 2D using a linear layer. As k for L_t^k increases, neighbor $l_k^t \in L_t^k$ tend to agree on a common representation of the *whole* represented in the image sample. At lower levels, smaller islands emerge, each representing a part of the *whole*. Samples of MNIST present fewer changes in the islands across levels because the data is much simpler, indicating that fewer levels in the hierarchy can be sufficient to obtain similar results. Our Agglomerator is thus able to represent a patch differently at different levels of abstraction. At the same level, spatially adjacent patches take the same value, agreeing on the representation of parts and wholes.

Latent space organization as the representation of conceptual-semantic relationship in data. Recent networks aim at maximizing inter-class distances and minimizing intra-class distances between samples in the latent space. While the accuracy is high, they provide little interpretability in their data representation. As a result, mistakes are less likely to happen, but the mistake severity, defined as the distance between two classes in WordNet lexical hierarchy [128], does not decrease [11]. As shown in Fig. 3.30, our network provides an organization of the input data, which is semantically closer to the human lexical hierarchy.

3.5.5 Limitations

Our method introduces new types of hyper-parameters in the network structure, such as embedding dimensions, number of levels, and size of patches, which

need to be tuned. We believe a better parameters setting can be found for all the datasets, increasing accuracy while still retaining interpretability. Moreover, a higher number of parameters generally causes architectures to be more prone to over-fitting and more difficult to train. To improve the accuracy of our network, we would need a pre-training on large datasets (e.g., on ImageNet), which requires large computational resources to be performed in a reasonable time frame. While hoping that powerful TPU architectures become publicly available in the future, we are currently investigating efficient pre-training strategies for our network.

3.5.6 Discussion

We presented Agglomerator, a method that makes a step forward towards representing interpretable part-whole hierarchies and conceptual-semantic relationships in neural networks. We believe that interpretable networks are key to the success of artificial intelligence and deep learning. With this work, we intend to promote a preliminary implementation and the corresponding results on the image classification task, and we hope to inspire other researchers to adjust our solution to solve more complex and diverse tasks.

Chapter 4

Human Pose Estimation and Ambient-Assisted Living

Estimating the 3D human pose from images and videos can lead to the development of multiple applications in many different fields. Perhaps one of the most suited application fields for HPE is medicine and in particular Ambient-Assisted Living (AAL) [61]. AAL deals with the recovery of motion abilities after a physical or neurological trauma, which is a long and winding road that is usually supported by medical staff. In particular, occupational therapists (OTs) play a fundamental role in assessing the performances of patients in daily life tasks and suggesting better practices and aids. The goal of OTs is to promote the patients' abilities, fostering their independence with the minimum needed supervision. To this purpose, the possibility of remotely operating while being able to monitor the subject with a sufficient level of detail is highly desirable. The development of advanced tools for patient monitoring and supervision with limited intrusiveness is a scientific challenge that spans different technological areas: sensing, localisation, tracking, measurement, business intelligence. The final goal is to provide medical doctors with a set of meaningful metrics related to the patient's activity. Among them, occupation of living spaces, motion patterns, posture, as well as fine-grained motion-related parameters (e.g., grasping objects, performing elementary tasks of variable difficulty), can be of great help

for the medical staff to assess the degree of independence of the user and to recommend suitable assisting devices, more appropriate organisation of living spaces, effective rehabilitation procedures.

In this chapter, we present two main developments that allow to solve real-world AAL problems with HPE techniques, namely:

1. **[Section 4.2]** A first study on the potential relationship between human pose, trajectory discontinuities and shoulder pain in wheelchair users. We show how fostering the development of best practices aimed at preventing the raise of shoulder-related pathologies, by correcting the user's movements and the wheelchair setup.

2. **[Section 4.3]** A fully automatic expert system that coordinates an augmented reality physical environment supporting occupational therapists and rehabilitation staff in evaluating their patients' performance and therapy activities, as inferred by a network of monocular cameras empowered by HPE networks.

4.1 HPE role in AAL

In this section we will explore the recent literature for Ambient-Assisted Living technologies and the role of Human Pose Estimation in various medical applications related to it.

4.1.1 Shoulder fatigue analysis

Many studies show the high correlation between the setup of the wheelchair and shoulder pain [4, 36, 37, 50, 126]. The literature shows how the force applied to push the wheelchair relates to the specific type of movement performed and also to the presence of obstacles in the user's path, elements that can be well modeled and described through the trajectory performed. Studies carried out in this

area cover different topics. The authors in [36] clearly highlight how the raise of shoulder pain in wheelchairs users during daily activities is a widespread phenomenon, so that the authors, as a final goal outcome of their study, define the Wheelchair User's Shoulder Pain Index (WUSPI). Using this index, they highlight an increase in pain in quadriplegic users than paraplegic users above all, in relation to increasing age [4].

4.1.2 Human detection and trajectory analysis

Most of the human detection approaches focus on purely vision-based approaches, which keep the implementation and deployment costs low, thus not requiring the adoption of expensive wearable sensors. The most straightforward solutions are based on background subtraction techniques for blob detection, followed by tracking [166, 205]. However, background subtraction is generally not robust, and tracking-by-detection algorithms are generally preferable [12]. From the tracking perspective, the analysis is generally conducted on the 2D trajectory. Many different solutions have been proposed. We will adopt the methods proposed by Piotto et al. [145], in which the authors explore the analysis and comparison of trajectory subsets: after subdividing the trajectory into short segments, they look for temporal discontinuities, as well as spatial discontinuities in terms of sharp direction variations.

A recent trend in computer vision is to estimate the human pose through deep networks. Accurately estimating the pose of a person in single images and applying tracking to the detected poses facilitates the task of trajectory analysis. More in depth related work on human pose estimation will be detailed in chapters 2 and 3.

4.1.3 EMG signals analysis

The electromyographic (EMG) signal is used to characterize the activity of a muscle. The activation of a particular muscle can be both voluntary and involuntary [5, 39, 174] These signals are very complex to handle, since they resemble the combination of numerous stimuli and are generally affected by a considerably large amount of noise, thus requiring the adoption of signal processing techniques to highlight the meaningful signal components.

Although the applicable filters may vary depending on the analysis, an overall good method for cleaning a signal is proposed by [5]. This method considers four main steps: filtering, rectification, smoothing and features extraction. The filtering and smoothing steps are signal-dependent, because every signal is affected by different noise sources and can have different ranges, depending on the performed movement, as well as the type of muscle being activated. A more detailed explanation is available in [207], in which the authors have listed possible filtering strategies together with the corresponding references. An additional feature of the EMG signal analysis is that it can be performed both in the time and in the frequency domain, as underlined by [5, 174, 207] depending on the different physical parameters [207] analyzed. There are four fundamental physical parameters that can be extracted from the temporal and frequency analysis, namely joint motion, force, velocity, and muscle fatigue. Those of greater interest, especially with regard to this work, are muscular force and fatigue. The first one, defined as force or muscle tension, can be extracted in the time domain analysis, due to the fact that it can be related to the amplitude of the EMG signal. In addition, for experiments with constant tension exercise, the average value of the signal that has been previously rectified, is a measure of tension. A feature widely used in literature to extract this value is the Root Mean Square. Instead, the muscle fatigue is generally measured analyzing the frequency domain, as it has been found that the fatigue produces changes in the EMG spectrum. In

particular, an increase in fatigue decreases the high-frequency components of the spectrum.

4.1.4 Activity and behavior monitoring

Activity and behavior monitoring of patients affected by cognitive and/or physical diseases is reported in several studies, and could be performed in both clinical settings and daily living environments for different purposes. As far as clinical monitoring is concerned, the literature shows that Intensive Care Units (ICU) are among the most critical hospital wards where an accurate control of activities is of paramount importance. In this case, monitoring is performed mainly using depth cameras and other non-invasive devices [38, 121]. Other studies proposed ambient monitoring systems to assess the fulfillment of safety procedures and to supervise complex equipment [24].

An interesting application of in-hospital monitoring concerns patients' evaluation for the elderly or subjects affected by chronic diseases. In this case, technologies support the clinical staff in both assisting the patients and assessing their needs, with the aim of determining whether they can return to normal life and the possible assignment of human support or aids [18, 170]. The monitoring system is requested to acquire manifold information, including physiological parameters, level of stress, behaviors, but also the ability to perform daily activities. Given the different nature of each disability and their diverse impact on the capability in performing various activities, customisation is a key issue. Various solutions have been reported in literature, mostly based on the adoption of sensor networks. On one side, wearable sensors are effective in measuring physiological parameters (ECG, body temperature, blood pressure [25]), collected while the subject performs daily activities. For instance, MyHeart [141] and MagIC [42] proposed wearable monitoring systems using small sensors attached to everyday clothes, to ensure unobtrusive recording of cardiorespiratory signals or user's movements. On the other side, environmental sen-

sors can collect useful information with lower obtrusiveness. For instance, the use of radar or acoustic sensing devices has been proposed to detect user's activities [23]. A more detailed overview of the solutions and platforms so far proposed to tackle physiological and environmental monitoring can be found in the works by Spasova et al., focusing on fall detection [168], Bonato et al., reviewing wearable sensors based solutions [17] and Acampora et al., focusing on environmental sensor architecture approaches [1].

A major limitation of monitoring systems developed so far lies in the fact that they are usually tailored to the specific needs which they are designed for. For instance, objective assessment of Parkinson disease is achieved through various specific solutions reviewed in [122], rehabilitation accounts a wide and fast growing spectrum of customised systems, such as the recently published [49]. On the contrary, the typical situation when dealing with elderly or disabled people is a unique combination of physical and cognitive limitations and comorbidities that requires a more holistic approach to be fully assessed. Previous attempts to tackle this problem are reported in [120], addressing severely injured people, and in [139] and [64], where different ambient-assisted living (AAL) solutions are reviewed and organised.

4.2 Joint fatigue and trajectory analysis

People with motor disabilities, whether it is a permanent or temporary disability, often require continuous assistance. The project AUSILIA (Assisted Unit for Simulating Independent LIving Activities, *Fig. 4.1*)[1] is a living lab, aimed at improving the user capabilities, thanks to an advanced technological infrastructure able to measure the user's skills and needs, recommending the most suitable instruments to fostering autonomy and independent living. As described in [146], within the premises of AUSILIA, users perform small daily activi-

[1] http://www.ausilia.tn.it

ties in a controlled environment simulating a real fully furnished house. In this way it is possible to observe what difficulties they would be facing once back home. The AUSILIA living lab consists of two main environments: the home apartment and the gym as shown in Fig. 4.1. Both the apartment and the gym are equipped with a multi-sensory infrastructure, necessary to capture the user motion and his/her interactions with the environment. The gym is the reference environment for this work. It is equipped with numerous platforms simulating different pavements and surfaces, in which wheelchair users can be monitored and their capabilities assessed. The goal of this paper is to jointly analyze trajectory and shoulder fatigue targeting paraplegic users, using manual self-propulsion wheelchairs, with different inclinations of the seat and in presence of obstacles in the route. The system is composed of two main modules: on the one hand we rely on a camera network for tracking and trajectory analysis; on the other hand we focus on the muscular activity through the positioning of an electromyograph and an electrogoniometer on the shoulder and arm of the user. The output of the analysis, conducted in synergy with the medical personnel, aims at highlighting a potential correlation between salient points in the trajectories and discontinuities in the shoulder movement. The data, as will be shown in the experiments section, is made available through an ad-hoc user interface, where doctors and caregivers can assess the motion and guiding abilities with reference to the specific wheelchair setup.

At the current stage of development, the presence of the electromyograph and an electrogoniometer is necessary mostly to acquire the ground truth, while in the future it is expected that the analysis is conducted by solely using video cameras and/or, more in general, non wearable sensors, for an improved user experience and limited invasiveness.

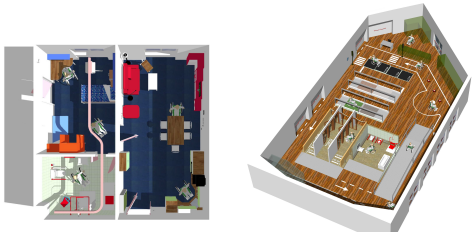


Figure 4.1: Layout of the apartment and gym of the AUSILIA project.

Figure 4.2: Image of a user testing all the described equipment: EMG sensors, electrogoniometer, and wheelchair.

4.2.1 Proposed model

Our focus is to jointly analyze wheelchair users' trajectories and the development of fatigue, mainly at shoulder level, in relation to their posture. The analysis has been conducted in the AUSILIA gym, by detecting sudden speed and direction changes in the path that can be linked to incorrect movements, to the presence of obstacles, such as steps, or to variations in the road inclination. For the user detection phase, we rely on the method described in [21] and [20], combined with a background subtraction motion model, that can support the algorithm is those cases, in which the skeleton detection fails, due to severe occlusions of the body details. For the shoulder and posture analysis instead, we adopt a mixed approach, that links data from the EMG sensor, the user's body pressure distribution, and pose data in terms of joints position. The main steps of the developed method are shown in Fig. 4.3.

Sensors positioning, calibration, and recording

Four cameras have been installed on the gym perimeter so as to guarantee the coverage of the whole area of interest. The approximate coverage on the ground by each camera is shown in Fig. 4.4.

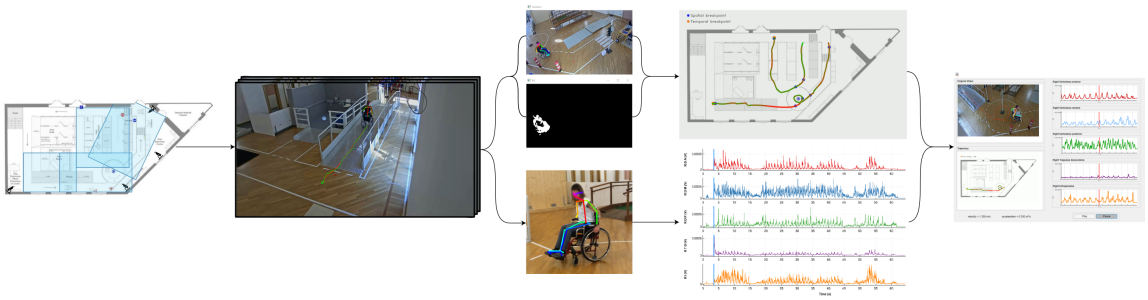


Figure 4.3: Pipeline of the proposed method: sensor positioning, camera calibration, video recording, real-time trajectory analysis, posture and arm fatigue analysis

Figure 4.4: Layout of the AUSILIA gym. The environment is currently covered by four IP cameras, necessary to monitor the wheelchair path. The approximate coverage of the cameras is highlighted in light blue.

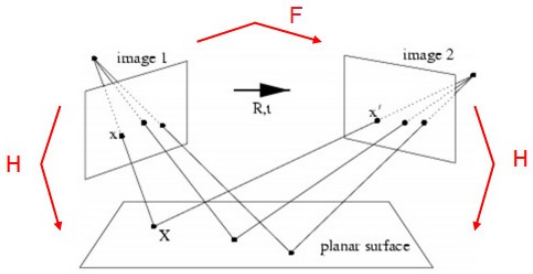


Figure 4.5: Computation of the fundamental and homography matrices

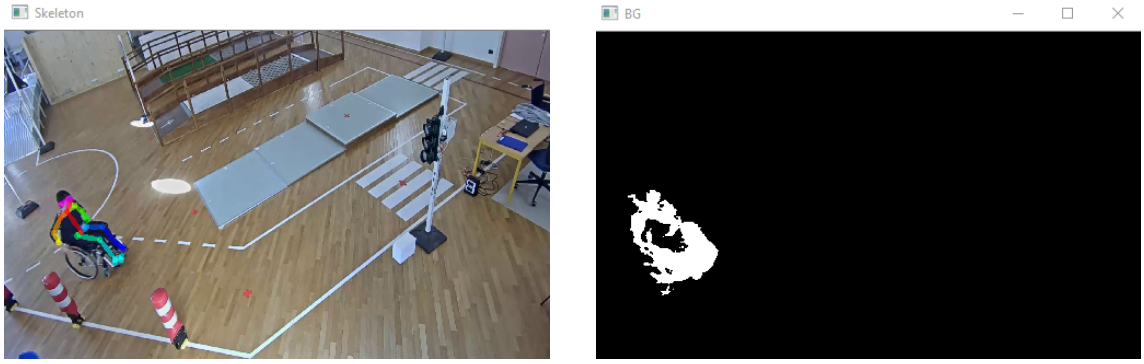
Cameras are subject to distortions of various kinds and therefore require calibration, to determine both the intrinsic and extrinsic parameters. As we are working with a small camera network, it is also necessary to relate each camera to the others in the network, via the computation of the fundamental matrix, as well as linking the cameras with the real world through the computation of the homography matrix.

With the camera network fully calibrated, it is possible to start collecting data. Since it is not convenient, in terms of required storage space, to record all the videos from all the cameras in the gym simultaneously, the goal is to trigger the recording of each camera based on the user's position, automatically

handling the hand-off. The final video consists then of the concatenation of the single views. However, in order to guarantee that no relevant data is lost, the entire videos for each camera are also temporarily stored, until the medical personnel confirms that no salient event has been missed.

Trajectory analysis through skeleton detection

In order to determine the users' trajectory, the first step consists of determining the position of the skeleton. For this phase, we rely on the OpenPose library [20], which allows for a good detection also in presence of partial occlusions. Only in case the skeleton is not found, we backtrack to background subtraction, so as to ensure a continuous tracking over time (Fig. 4.6).



(a) Skeleton extraction (b) Background subtraction

Figure 4.6: Example of skeleton extraction and motion detection through background subtraction.

We can now define a bounding box that resembles the position of the users at a given time instant. At each frame p_G represents the ground point. When the pose estimation succeeds, p_G can be identified as the median point of the 11th and 14th skeleton joints (heels joints). When pose estimation fails, p_G is located at the lower median point of the bounding box obtained via background subtraction.

At this stage, the trajectory is still noisy and uneven. As a consequence, a fil-

tering and smoothing phases are necessary to obtain a more uniform and meaningful trajectory. Therefore, we employ a two-step filtering approach: first, we apply a Kalman filter; we then refine the obtained trajectory with a moving average filter. The outcome of the trajectory filtering is shown in Fig. 4.7.

(a) Smoothed trajectory in the real world

(b) Projected smoothed trajectory

Figure 4.7: Trajectory filtering: the green line represents the output of the Kalman filter, while the red line is the result of the moving average filter.

All the extracted points that contribute to the definition of the trajectory can now be mapped to the ground plane in the real world coordinates thanks to the calibration procedure described above.

Velocity and acceleration are then extracted by computing the values over a one-second window, so as to cope also with minor irregularities that might occur at smaller time scales. In addition, the speed value for each frame is also used to graphically indicate on the map the exact points where a change in speed occurs, using an ad-hoc colormap changing from green (slow) to red (fast). Following the procedure described in [145], the trajectory is analyzed to look for possible temporal breakpoints, defined as sudden stops and re-starts. Such discontinuities happen when the position of the object of interest at the beginning and end of a defined time interval $[t_k, t_{k+1}]$ is within a predefined range Δ. Mathematically, and according to [145], this implies:

$$P^i_{k+1} = P^i_k \pm \Delta \quad \text{where} \quad P^i_k = (x^i_k, z^i_k, t_k)$$

A temporal breakpoint is inserted when, within a second, the position of the wheelchair varies less than 30cm. In [145], also spatial breakpoints are extracted to describe the trajectory, defined as sudden variations of the motion direction. This means computing the angle between two segments of the trajectory in order to search for sharp direction variations:

$$\beta_k = \left| \frac{m_{k-1} - m_k}{1 - m_{k-1} m_k} \right| > \beta_{th}$$

$$\text{where} \quad \begin{cases} r_{k-1}(x) = m_{k-1}x + q_{k-1} \\ r_k(x) = m_k x + q_k \end{cases}$$

In our case, we use a temporal reference of two seconds in order to calculate the angle of rotation performed by the users at a precise instant.

Figure 4.8: Trajectory performed by a test user projected on the gym map showing: speed changes, spatial breakpoints and temporal breakpoints.

Shoulder forces and posture analysis

In addition to the extraction of the trajectory of the wheelchair, our aim is to also study the posture and the arm movement while pushing the wheelchair. In particular, the shoulder movement is analyzed in detail as the prolonged use of wheelchairs with manual self-propulsion causes shoulder disorders in most

users. As a consequence, the pushing force distribution is analyzed to understand if there are discontinuities that, in the long period, can be a potential source of pain. To achieve this second objective, several tools have been used: five EMG sensors to detect the muscular activity of the shoulder, the angle of the elbow extracted from the skeleton, and an electrogoniometer, mostly used as a ground truth to measure the reliability of the OpenPose joint extraction. Each user is asked to test three different types of wheelchairs (Fig. 4.11). The EMG and the electrogoniometer are used to directly analyze the user's movement. The motivation for using three different wheelchairs, instead, is necessary to evaluate the performances while changing on the one hand the weight of the wheelchair itself, and on the other hand the responsiveness of the wheelchair when a force is applied to the wheels. This second difference mostly depends on the different inclination of the seat and, consequently, the different posture of the user. For this reason, in addition to the analysis of the shoulder, the objective is also to understand if such different postures can help in preventing shoulder pain.

The most significant data for the study of user movement, and in particular the shoulder movement, is obtained from the EMG sensors. A common framework to analyze these data is MOKKA [2], which is is provided by the open-source Biomechanical ToolKit (BTK) [3].

The EMG sensors measure the muscle activity, including both voluntary and involuntary movements performed by muscle fibers.

The amplitude of the signal is analyzed in the time domain, as it reflects the force used by the muscle, i.e. its muscle activation. In the frequency domain, it is possible to extract values related to the fatigue.

In our implementation the EMG signals have been analyzed and processed in the time domain through the following steps (*Fig.* 4.9): data extraction, filtering,

[2] http://biomechanical-toolkit.github.io/mokka/

[3] https://code.google.com/archive/p/b-tk/

rectification and RMS (*Root Mean Square*) envelope.

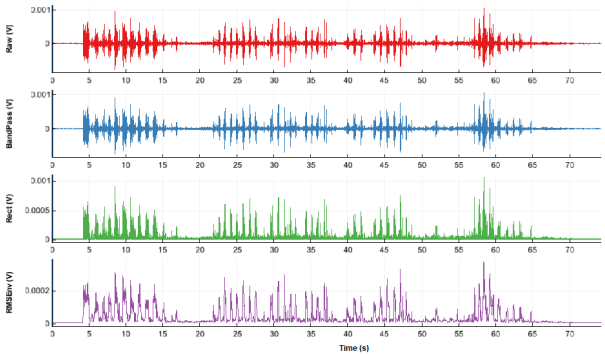


Figure 4.9: EMG signal processing: raw signal (red), band-pass filtered signal (blue), rectified signal (green) and the RMS envelope (violet). The signal amplitude is reported in Volt.

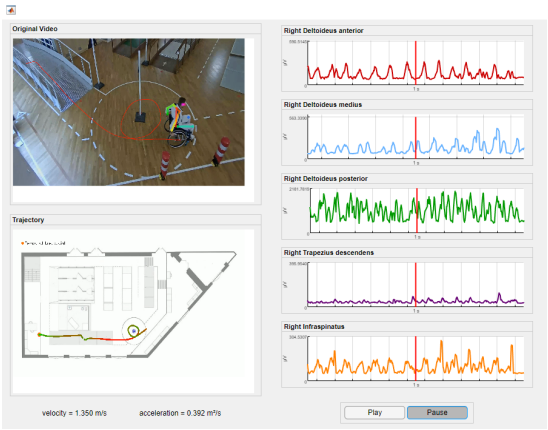
Since the signal analyzed in the time domain is a representation of the muscular force, that is the muscular activation, it is possible to define the mean muscle activation by computing the mean of the signal. This average value is extracted for every signal and for every trial using the different wheelchair. Consequently, by calculating the ratio between the average values of each signal passing from one wheelchair to another, it is possible to understand if and how the muscular activation changes if the seat is changed.

Together with the EMG information, the angle of opening and closing of the elbow is also extracted, to have an additional data on the control on the movement of the arm. However, the electrogoniometer is rather invasive and we preferred using the joints of the skeleton, providing a weak replacement for the electrogoniometer data, for a more comfortable experience for the user, who will not be required to wear invasive sensors during the experiments. From the skeleton extracted at each frame, the positions (x, y) of three joints are extracted: the shoulder, the elbow and the wrist. These positions are stored so that they can be used and processed later to evaluate the variations of the angle at the elbow level. The procedure used for the calculation of the opening and closing of the elbow, relies on the same strategy adopted for the identification of spatial and

temporal breakpoints in the trajectory analysis.

Results correlation and displaying

At this point of the study, the data is processed, and can be viewed by the medical personnel in order to correlate the information coming from the different information sources, bearing in mind that each source has different temporal resolutions requiring an additional effort in synchronization. An overview of the final layout of the interface is shown in Fig. 4.10b.

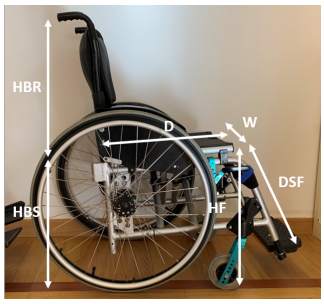


(a) Graphical interface

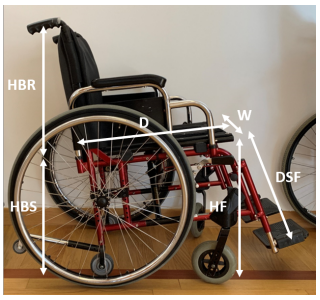
RDA = 219.30 µV	RDM = 184.34 µV	RDP = 819.16 µV
RTD = 106.58 µV	RI = 98.19 µV	
Mean velocity = 0.97 m/s	Max velocity = 2.26 m/s	
Duration = 50.80 s	Time = 28.48 s	

(b) An example of extracted data

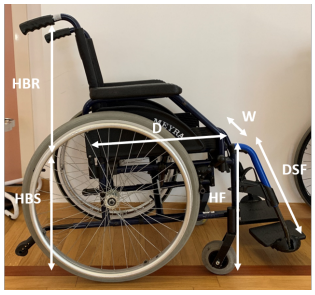
Figure 4.10: Graphical interface developed to display all data at the same time. A play / pause button is present to be able to observe in detail all the signals at a precise moment. The displayed data includes the mean value of muscular activation for every EMG sensor, the mean and maximum velocity of the user as well as the entire duration of the trial.



(a) ”Vega”: super light, responsive.



(b) ”Ministar”: light, responsive.



(c) ”Meyra”: light.

Figure 4.11: Characteristics of the three types of wheelchair, which vary in terms of weight and responsiveness.

4.2.2 Results

The main objective of this study is to find a correlation between the development of shoulder pain in users and the prolonged or incorrect usage of the wheelchair. Considered that the study presented is pioneering in its nature and not a proper clinical study, we involved six healthy users of different age, musculature and gender, as shown in Table 4.1. In the considered environment, users are subject to clinical trials and are constantly monitored by therapists and nursing staff on a daily basis, in order to optimize the choice of the most suitable wheelchair.

In the tests, users were asked to perform an established route three times, each time using a different wheelchair model. At the end of each trial, users were required to fill in a questionnaire to collect their direct feedback about the level of fatigue.

Trajectory analysis

For what concerns the trajectory analysis, the obtained results confirm our initial hypotheses, showing how the speed and the motion properties of users change in presence of obstacles and turns, while maximum speed is reached in presence of straight paths and descents. An example of the salient interest points, repre-

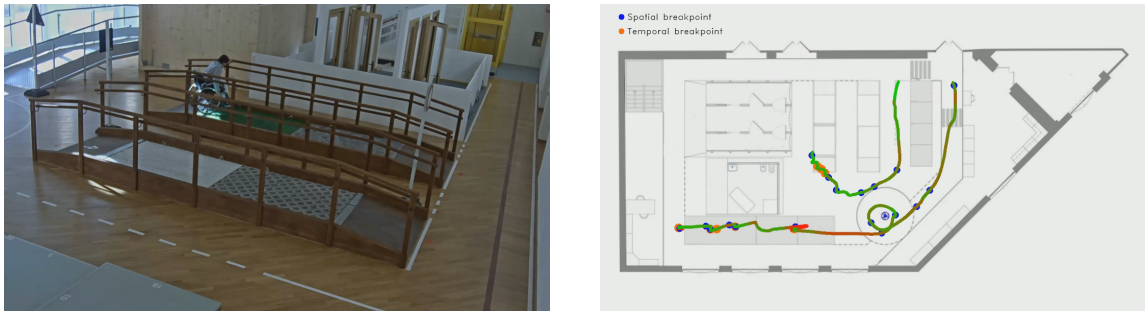
	Gender	**Musculature**	**Experience with wheelchairs**
User 1	male	well-developed	3
User 2	female	normal	2
User 3	female	normal	1
User 4	female	normal	4
User 5	male	normal	2
User 6	female	normal	1

Table 4.1: Users involved in the experiments, with experience level between 1 (beginner) and 5 (expert).

sented through the temporal and spatial breakpoints, can be seen in Fig. 4.12. It is shown how User 6, with a **Level 1 (beginner)** wheelchair usage experience, encountered severe difficulties in facing obstacles in the pre-determined path, as also highlighted by the breakpoints plot in Fig. 4.12b. The user also encounters difficulties while getting on the ramp in Fig. 4.12a. To continue along the path, the user suddenly changes the motion direction to reach for the handrail, which was not in range along the path. We can notice that a set of useful information can be inferred from the trajectory plot in 4.12b, in particular: (i) the speed has decreased significantly (light green) during the ascent, as shown by the presence of temporal breakpoints; (ii) when the user continues the test with the help of the handrail, a spatial interruption is drawn on the map due to an abrupt change in direction. Such breakpoints are successfully detected by our trajectory analysis and they also match the answers that users gave in the questionnaire. Moreover, users identified the climbs, descents and also the different terrain types as the most difficult parts along the path, which is in line with the data observed in our arm movement and posture analysis.

From the analysis it turned out that the results of the trajectory analysis in relation with the different wheelchair models were less relevant than expected.

By changing the wheelchair model from the most responsive to the less responsive one (Fig. 4.11), a progressive increase in the employed force was expected, associated to a decrease in speed. However, this trend was not evident in every test, as can be seen in Fig. 4.13. This might be due to the fact that users have not been pushed to maximum strain, and only a limited number of trials has been performed. However, despite the small number of samples used to perform the analysis, both the quantitative results and the user questionnaires show how the wheelchair weight is not the main factor linked to speed variations along the path. In fact, from such a preliminary analysis, it appears from the trajectory data, that the inclination of the wheelchair along the path has a far more relevant influence than the weight variation of the wheelchair itself.



(a) User 6 that stops on the ascent and continue the path clings to the parapet

(b) Temporal and spatial breakpoints on the map caused by the difficulties encountered

Figure 4.12: Example of difficulty caused by obstacles (user 6).

Arm movement and posture analysis

We conducted a similar analysis also for the shoulder movement and user posture, relating them with the presence of obstacles along the path and the wheelchair type. As regards the comparison with the different types of wheelchair, we calculate the mean value $\mu_p : p \in P$ of each user's muscular activity, as extracted by the EMG sensor, and compare with the rest of the user population P. Then, by calculating the ratio of the mean value in passing from one wheelchair model

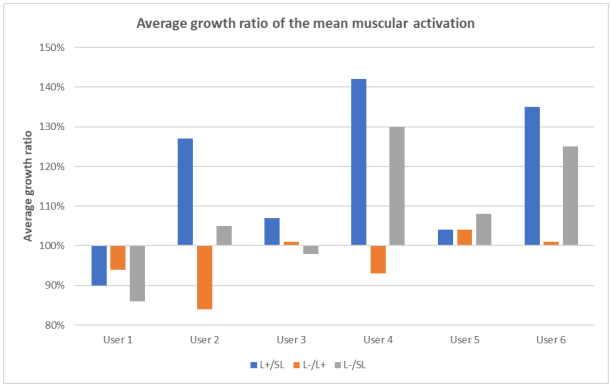


Figure 4.13: Average growth ratio of the mean muscular activation for the 6 users on the three wheelchair models. Notice how the muscular fatigue changes based on user experience and how in our tests it is not strictly linked just to the wheelchair weight.

to the other, it was possible to quantify the change in the force applied Δ_f, as performed by the user. The data obtained from this analysis are shown in Table 4.2. Also in this case the idea is that when the wheelchair changes, the user's force increases proportionally to the difficulty of the path. During the evaluation of these parameters, while speed seems to be more related to the wheelchair inclination, the required force is more related to weight. In fact, an increase in muscle activity occurred for five out of six users, when switching from a superlight to a light wheelchair, but both tilted. Instead, only half of the users in the panel has increased the muscular force by varying the inclination of the seat.

Results correlation

By visualizing and integrating all the data together through the developed graphical interface (Fig. 4.10), it is possible to have a complete and detailed view of the entire trial. Through the video and the projection of the trajectory on the gym map, it is possible to observe when and where the user's difficulties occur, and understand how specific obstacles in the path can lead to an increasing fatigue. At the same time, in addition to visualizing spatial and temporal breakpoints in the path thanks to the trajectory analysis, it is possible to also visually

	L+/SL	L-/L+	L-/SL
User 1	90%	94%	86%
User 2	127%	84%	105%
User 3	107%	101%	98%
User 4	142%	93%	130%
User 5	104%	104%	108%
User 6	135%	101%	125%

Table 4.2: Average growth ratio of the mean muscular activation between pairs of wheelchairs: super light, responsive (*SL*), light, responsive (*L+*), light (*L-*). A 100% ratio indicates that with the second wheelchair the user employed 100% of the average force used with the previous wheelchair, therefore a constant muscle activity (ratio equal to 1). While a 90% ratio indicates that with the second wheelchair only 90% of the previous force was used, i.e. there was a 10% decrease. A value greater than 100% therefore indicates an increase in muscle activity.

correlate the EMG signals of each muscle at every precise moment. Observing the signal during the various phases of the route, it is possible to notice how, in presence of situations considered critical by the users (ascent, as well as different surfaces like grass and stone), muscle activity increases with numerous peaks. This allows the medical personnel to improve their knowledge about the user's fatigue, which would be otherwise difficult to gather, clearly showing the benefits of having a distributed sensing framework like the one available in AUSILIA.

This confirms that the setup we have envisaged and the multi-sensory environment we have defined can indeed be of help in correlating the muscular activity to the trajectory. We believe that the main limitation we have been facing so far is mostly due to the small number of users involved and trials performed. This issue is currently being addressed via the definition of an ad-hoc clinical study.

4.2.3 Discussion

We presented an approach for joint trajectory and fatigue analysis in wheelchair users. We combine the state-of-the-art skeleton-based pose estimation techniques, classic computer vision algorithms and data from wearable sensors to provide the medical personnel with a complete solution that enables them to evaluate an ensemble of parameters jointly, while performing a pre-defined task. Our study shows that a relationship exists between the trajectory data and the EMG signals, both in terms of velocity, and temporal and spatial breakpoints.

4.3 Building HPE-based expert systems

Occupational therapy aims at assessing and evaluating the activities of various types of patients, in order to guide them in regaining or improving their motor or mental skills through rehabilitation. However, fostering independence during and after rehabilitation is a fundamental yet difficult process. In fact, the interaction with fragile subjects requires a special mix of technical competence and sensitivity to personal elements, in a patient work of trust enforcement. From OTs' perspective, the possibility of observing patients performing daily activities allows understanding their weaknesses, tracing down their improvements in the rehabilitation process, and making it possible to identify the most suitable assistive technologies, tools, or clinical paths that can contribute to provide them with a *new normality*. While direct observation is currently the most adopted practice, well-established in the literature with plenty of examples [6, 31, 33, 51, 98], it typically introduce a bias in the analysis, because of the physical presence of the therapist together with the patient. With this work we aim at mitigating this bias as much as possible, by providing the medical staff with a sophisticated environment that allows monitoring the patients in completely realistic situations and in their typical living conditions (alone or supported by a caregiver). To this purpose, the proposed system allows mon-

itoring the person both in real-time and offline, and generates an augmented presentation of the captured information by automatically processing the sensor data, analysing them, generating enhanced views, annotating, overlaying measures and data, and so on. The same data are also structured in reports for better accessibility.

The main goals of the proposed system can be summarised as follows:

- Enriched data and metrics: the system provides a number of information on the observed environment, coming from both the direct sensor observation (cameras, 3D sensors, environment/object sensors, wearable, etc.) and the integrated processing of such data with a set of tools (motion analysis, tracking, localisation, action patterns, space occupation, etc.)

- Fully automatic analysis: the implemented architecture is designed to meet the requirements of non-expert users (OTs' and medical staff's). The user has just to turn-on the system and start collecting trials and analytics. Afterwards, he can access the data through a user-friendly multi-modal interface, to infer the clinical conditions of the patients and assess their progresses.

- Minimum invasiveness: the configuration of the environment is organised in order to hide as much as possible the complexity of the infrastructure. Sensors are embedded into the environment and in objects and equipment placed in remote, so that the user can live a 'feel-at-home' experience.

- Portability and real-time compliance: the system is organised according to a distributed and modular structure, which allows to dynamically configure new environments and port it to other scenarios, upon need. Furthermore, it allows the involved staff to access and visualise the metrics and results in real-time from mobile devices such as tablets and laptops, without waiting for additional post-processing steps.

4.3.1 Context: the AUSILIA Facilities

The presented expert system has been developed to work inside a domotic apartment and a gym with different reconfigurable environments, which are hosted inside the Villa Rosa hospital in Pergine, Trento (Italy). The units of the AUSILIA project represent the spaces where users spend their time carrying out their daily activities. The apartment was designed to accommodate people with different cognitive or physical impairments and therefore with different needs. For such a reason, therapists ask to evaluate their patients in an apartment that allows to cover different type of situations. Given the size of the domotic apartment ($\sim 70m^2$) that did not allow to meet all the therapists' requests and the patients' needs, a gym was realised as a parallel environment of $\sim 400m^2$ where even the most specific solution could be tested. In the gym, environments can be highly reconfigurable such as the door clearances or corridor widths. With these facilities, therapists have the opportunity to have a real test-bench, allowing an effective understanding of their patients' problems and needs. Moreover, both the domotic apartment and the gym have been a perfect test bench for two research laboratories [4] [5] of the University of Trento in order to design, develop and test the proposed expert system in a realistic and fully-fledged AAL environment.

4.3.2 System overview

In this section we review the main building blocks of the system and the way they are integrated and orchestrated to collect and process the data. Our proposed system is depicted in Fig. 4.14. On the sensing side, a set of camera (IP, infrared and stereo), wearable, and environmental sensors are employed. The raw data extracted in realtime by these sensors are processed by the inference

[4] DISI UniTN `https://www.disi.unitn.it`
[5] DII UniTN `https://www.dii.unitn.it`

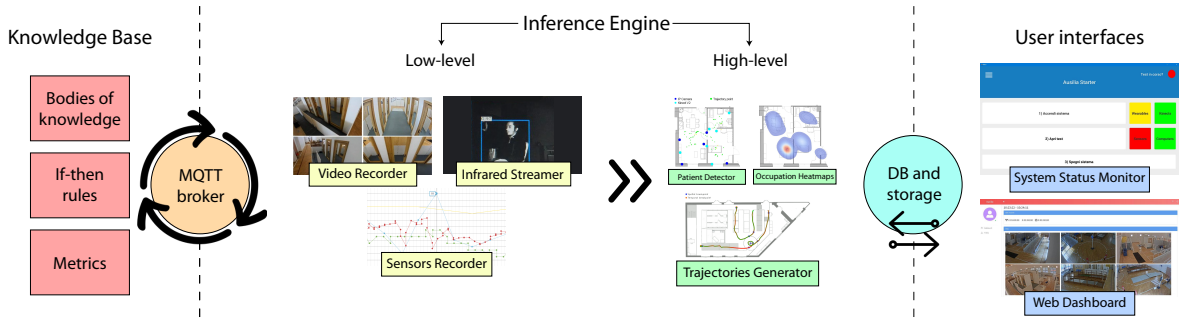


Figure 4.14: An overview of the proposed expert system. The system is enabled to work with different kinds of low-level data such as video streams, environmental and wearable sensors data. The inference engine deals with transforming all the low-level information into high-level features, according to the knowledge base. An MQTT broker serves as orchestrator for all the parts of the system, while all the required features are stored into a structured database linked to a network-attached store. Finally, the end user can access and edit all the required parameters of the system through a system status monitoring console and visualise the system metrics and gathered data from an interactive web dashboard.

engine to obtain high-level features, according to the knowledge base, which is jointly determined by the medical and the technical staff. All the structured information, together with the system status and parameters, is stored and can be accessed by the final user through a set of human-friendly user interfaces. The means of communication between the three main layers is a Message Queue Telemetry Transport (MQTT) orchestrator and a network-attached storage for data collection.

In the following subsections we will detail the role of each communication, inference and presentation module.

Communication modules

The proposed expert system is configured as an ensemble of different inference modules that tackle different automation tasks inside the AAL environment, based on bodies of knowledge. Each module is configured as an always-on service with on-demand processing triggered by specific MQTT messages. The

MQTT broker and the network-attached database and storage are the two main communication modules of the proposed system.

- MQTT broker: it manages topics for different rooms and activities and forwards messages to the proper services. The usage of MQTT as the main orchestrator of the expert system allows for a fine-grained configuration of environments and rooms through MQTT topics, as well as allowing to save computing resources by hibernating services when not needed.
- Database and storage: the proposed expert system interfaces with a MongoDB database to store all the information used by the presentation and metrics evaluation modules, such as timestamps, events and positions inside the environment. Although the system is intrinsically distributed, all the big data such as videos and sensors outputs must reside in a unique Network-Attached Storage (NAS), which is a necessary component, especially when multiple clients are allowed to obtain access to specific data, even simultaneously. More information on the role of the database and storage for data processing and presentation can be found in sections 4.3.4, 4.3.5.

Inference modules

The inference engine in the proposed system transforms the knowledge base into concrete instructions, both low-level and high level. The inference services of the proposed AAL expert system are configured as follows:
Low-level:

- Video Recorder: it synchronises and records IP camera streams through the RTSP protocol in real-time, aided by GPU acceleration.
- Infrared Streamer: it synchronises and locally streams infrared streams via the UDP protocol.

- Sensors Recorder: it synchronises, filters and records both wearable and environmental sensors data in real-time.

High-level:

- Patient Detector: it processes and assembles infrared streams to detect the patient and caregivers position and rooms inside the AAL environment
- Trajectories Generator: it generates and filters human poses [21] from both infrared and RGB streams to infer and save both the patient and the caregivers trajectories
- Occupancy Heatmaps Generator: it processes the trajectories to infer a 2D occupancy map for the different areas (Fig. 4.22)
- Automatic Activity Annotator: it collects additional data about the activities (e.g., force sensors, pressure sensors, etc.) and links and integrates this data with the activities themselves for future presentation.

Presentation modules

The user interface modules communicate with the network-attached storage and database to obtain structured information about the system status and the past and ongoing tests in the environment. Through these interfaces, a registered user is also allowed to edit some of the parameters of the system's configuration. The presentation modules are the following:

- System Status Monitor: it allows OTs and the medical staff to easily check for the status of the ensemble of sensors and environments, enable or disable modules, as well as allowing them to start and stop new tests and automatically record the needed data
- Web Dashboard: it collects all the data from the OTs tests and presents it in an organised, everywhere accessible web interface

- Test viewer: it allows medical staff to review tests performed and saved in the database. This functionality shows the video of the IP cameras and the audio recordings of the tests. It also allows to reproduce the registrations starting from a specific time point.

- 3D Visualiser: this tool is dedicated to visualise the tests in a 3D immersive environment. OTs must wear augmented reality glasses and headphones to watch the recorded tests. In this context, user can interact with the environment by virtually moving inside different views around the virtual space or by reproducing a certain dataset in slow motion, fast forward or by freezing the reproduction at a specific frame. Clinicians can also interact with the output features data on a graph in order to clearly focus on a specific frame and analyse the desired features (Fig. 4.19).

4.3.3 Sensing subsystem

The system exploits a large sensing infrastructure to achieve evidence of all the necessary parameters related to the activities performed by the user. This includes the recording of the action itself, in 2D and 3D, the acquisition of additional information coming from the environment and the manipulated objects (e.g., the pressure on the floor, the force and acceleration impressed to a given object, the tremor of the hands, etc.), and the physiological parameters of the patient when performing that activity (e.g., pulsations, breath, stress, etc.). All the information captured by the sensing system is adequately synchronised and timestamped, and then recorded into a network-attached storage (NAS) composed of 4$\times$ 8TB hard drives in ZFS (Zettabyte File System) RAID. At the same time, the saved data are indexed into a MongoDB database for later access by the high-level and presentation layers. In this section, we review the sensors adopted in the system to capture the relevant information.

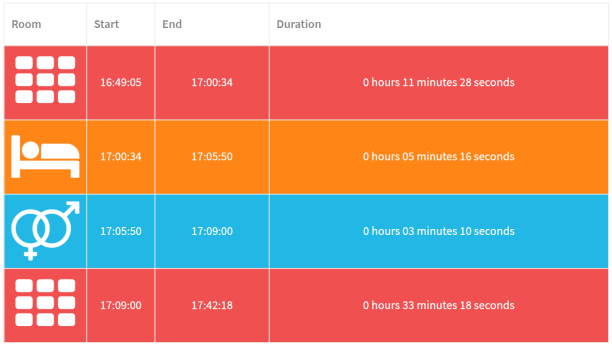


Figure 4.15: Automatic videos hand-off and cutting through global user position monitoring.

IP cameras

A big set of metrics of the proposed expert system are being extracted from simple video streams from common AXIS IP cameras [6]. Once a set of simple IP cameras is installed for each environment in the living lab, the proposed expert system takes care of:

- Synchronisation, simultaneously distributing the RTSP stream processing through the nearest computing units
- Video acceleration on GPU using Nvidia CUDA, whenever possible, to enable fast multi-view processing and saving
- Automatic views handoff through human pose estimation, to limit the video recordings to the moments in which the user is interacting with the environment (Fig. 4.15)
- Fast automatic camera network re-calibration through human mesh recovery, as thoroughly described in [55]
- Real-time 3D users position based on IP and depth cameras joint calibration (an example of output is shown in Fig. 4.16)

[6] AXIS M1125 IP camera `https://www.axis.com/products/axis-m1125`.

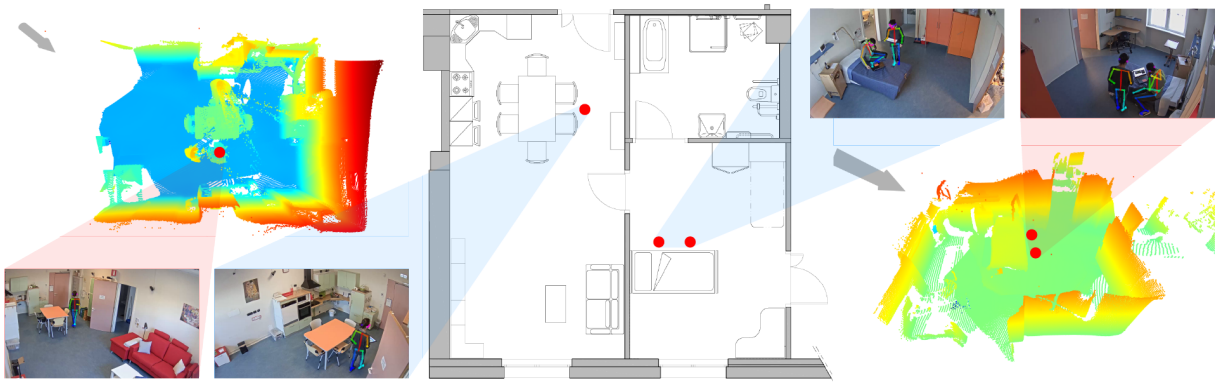


Figure 4.16: 3D users position detection through IP and RGBD cameras joint calibration.

A big advantage of our system architecture is that thanks to the fast and fully automatic camera network calibration module, each environment can be easily and dynamically re-configured, without the need of an expert technician for calibration or re-calibration and neither calibration instruments such as checkerboards. The system only requires walking humans in order to re-calibrate itself. Consequently, cameras could be even plugged-in, removed or moved between test sessions, without losing functionalities in the entire system.
Additionally, having a continuously calibrated system of both 2D and 3D vision sensors renders the environment an excellent test bench for both research and working conditions.

ZED cameras

The proposed expert system also allows for the capture and recording of streams coming from more complex cameras, such as stereo cameras. As an example, in the AUSILIA gym environment we employ several Stereolabs ZED [7] and ZED 2 [8] cameras that allow the system to obtain additional useful information, such as depth maps, high frame rate and resolution videos, refined 3D human poses and meshes. In Fig. 4.17 you can see an example of the output of a top-

[7] Stereolabs ZED stereo camera `https://www.stereolabs.com/zed/`.
[8] Stereolabs ZED 2 stereo camera `https://www.stereolabs.com/zed-2/`.

Figure 4.17: An example of stereo frame extracted from a ZED 2 camera inside the AUSILIA environment, for fine hands motor skills evaluation.

view ZED 2 stereo camera inside the AUSILIA gym, used for fine motor skills evaluation.

RGBD cameras

A set of Microsoft Kinect V2 [9] Time of Flight (ToF) cameras is installed in the domotic apartment to monitor the users' actions and interactions with the objects in the environment. Each Kinect V2 sensor is assigned to a processing unit, which takes care of the image processing and streaming aspects. The configuration of the Kinect multi-camera system ensures a coverage of most of the monitored area. 3D data information is integrated with 2D data to track user position in the environment. Kinect cameras can detect the patient's silhouette starting from the acquired point cloud. To avoid the recognition of unwanted objects in the scene, it is necessary to undergo an initial calibration phase, in which a fused 3D background point-cloud snapshot is saved in order to be subtracted later on to isolate the human silhouettes.
Additionally, at each instant of time, an infrared frame is captured by each Kinect V2 in the environment and broadcasted through UDP within the local network. The multiple UDP infrared streams will be used for data processing and fusion, and in particular for the users' identification and tracking, as ex-

[9] Kinect V2 Time of Flight camera `https://developer.microsoft.com/en-us/windows/kinect/`.

Figure 4.18: Virtual bathroom environment with 2 zones where viewer can move

plained in section 4.3.4.

Environment and object-mounted sensors

In addition to patient tracking, an evaluation of the user residual manual skills is provided by the analysis of tremors and trajectories during daily activities (brushing teeth, drinking etc.) using LP-Research accelerometers fastened to different objects (toothbrush, glass, etc.). Each room in the domotic apartment is supplied by a set of specific sensors according to the different activities which usually take place in a certain room.

In particular, in the bathroom six Adafruit Water Flow sensors are installed to detect water flow distribution through the shower, bathtub and sink (measuring both cold and hot water flow in *ml/s*). At the base of the sink Sa.Ni., *Ultra-sensor* pressure matrices can detect the distribution of patient's weight while performing standing tasks. Furthermore, six FSR 400 Interlink local pressure sensors installed under the seat, monitor movements and stability of the patient while standing on the toilet board (Fig. 4.19).

The kitchen and the living room have different sensors installed: five Seed-Studio Non-Invasive AC Current Sensor (15A) are responsible for measuring the level of electric current flowing through the household appliances (oven, microwave, induction hob, dishwasher and television) during different user ac-

tivities. The data relative to the force applied by the user on a table are given by four load cells fastened on each table leg.

In the bedroom, two sensors are placed for the detection of the user in bed. The first is a a 100Kg type S Load Cell which is responsible for measuring the force applied by the user when grabbing the trapeze bar over the bed. The second is a SeedStudio Non-Invasive AC Current Sensor (15A) which is fastened on the motor of the adjustable electric bed to measure the current flow.

Biometric sensors

A set of wearable sensors are fastened to patients' body to obtain continuous monitoring of physiological data. Breath rate and one lead ECG are detected using a sensorised Smartex T-shirt while the Blood Volume Pulse (BVP) and Electrodermal activity (EDA) raw data are measured with the Empatica wristband device. A Myo sensor provides measurements on the muscular activity of the forearm, which are useful for understanding the capabilities of the patient in manipulating objects. Wearable sensors allow a non-invasive detection of the physiological parameters, but it is necessary to be careful while applying these devices to the patient. The sensorised T-Shirt should be the right fit, in order to avoid loss of signal due to a wider size or an uncomfortable sensation due to a tighter size.

4.3.4 Processing subsystem

We have seen how the highly configurable properties of the proposed system allows for fast and structured data collection by sensors and cameras across all the environments. However, many times raw data signals may need to be filtered and re-organised according to the specifications of the task. For this reason, it is possible to rely on the underlying MQTT broker to write custom modules that take as input raw sensors data, process it and return custom outputs

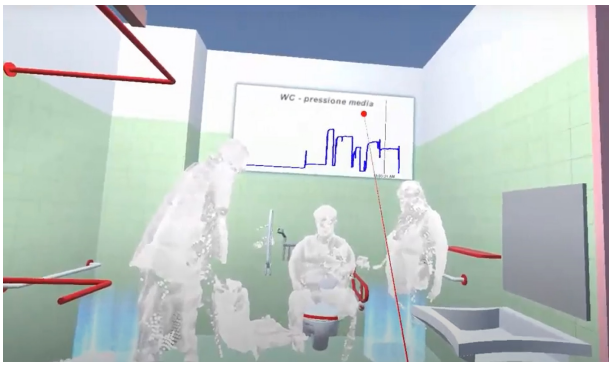


Figure 4.19: Example of ambiental sensors output: average force applied on the sensorised WC and point clouds of patient and therapists

that are going to be saved as a new record in the database and in the attached storage. Some use cases of custom data processing modules implemented inside the AUSILIA domotic apartment are presented below.

Users identification and tracking

In ambient-assisted living scenarios, many metrics derive from the interactions of the user with the environment. Therefore, it is of fundamental importance to be able to identify and track users and caregivers separately. Commercial indoor tracking devices exist for that purpose, but they may not be ideal to adopt in certain scenarios, such as:

- Large coverage needs
- Complex indoor topology
- Need for non-intrusive technology
- Restricted budget

Additionally, having a complex and already calibrated camera network at disposal makes it easier to build a vision-based identification and tracking system. We propose a vision-based approach for users and caregivers identification and tracking, which relies both on a real-time people detection network working on

infrared images (based on [15]) and a simple tracking-by-detection algorithm. An overview of how the proposed detection pipeline can be seen in algorithm 4. To sum it up, we take as input the infrared images I from the Kinect V2 sensors, streamed via UDP, and the set of camera matrices K. We apply an off-the-shelf infrared people detector (Fig. 4.20) and project the detection on to the global point cloud of the environment to obtain a 3D coordinate for each detected person. Subsequently, we apply a threshold on the infrared image, to detect specific reflective patterns that can be found on the occupational therapists' uniforms. By doing so, we can assign a label 0 (caregiver) or 1 (user) to each coordinates triplet p_1. Finally, we obtain the set of unique labeled 3D positions P and return it as output. Algorithm 4 is repeated for each instant of time.

Algorithm 4: Find a non-repeating set `P` of people containing their global 3D position and a label that refers to a caregiver (0) or a patient (1)

PatientDetector (I,K)

inputs : A set $I = i_0 \dots i_n$ of infrared images; a set $K = k_0 \dots k_n$ of camera matrices

output: The labeled 3D positions $P \quad | \quad p_i \in P$ of the user and caregivers, where $p_i = x,y,z,label$

$P^* \leftarrow \emptyset$;

$P^* \leftarrow InfraredDetector(I)$;

foreach *person* $p_i \in P^*$ **do**

  $t_{p_i} \leftarrow threshold(p_i)$;

  if $t_{p_i} \geq 0$ **then**

    $p_i[label] \leftarrow 0$;

  else

    $p_i[label] \leftarrow 1$;

$P \leftarrow merge(P,K)$;

return P;

Figure 4.20: People identification through IR (infrared) sensors (left) and tracking-by-detection (right).

Motion trajectories extraction

Human trajectories are of extreme importance in computer vision. Research fields such as trajectory prediction count increasing numbers of publications in literature and trajectory datasets are ever more requested. However, annotating a trajectory dataset is still a cumbersome process, that needs to be supervised in many cases, mostly due to occlusion and re-identification issues. Having a calibrated multi-camera system at disposal makes it easier to automatically annotate trajectories. As an example, in the AUSILIA gym we apply a mixed background subtraction and human pose estimation in order to obtain filtered and smooth trajectories, even detecting spatial and temporal breakpoints. Further details about the process are explained in [154] and an example of a resulting trajectory can be seen in Fig. 4.21.

Occupancy heat-map generation and habits tracking

Often times, occupational therapists have to scout through up to hours of video footage in order to estimate the occupation of environments by certain users. This approach is indeed not optimal in terms of time optimisation, and it is prone to errors. As an alternative, heatmaps can be built starting from raw po-

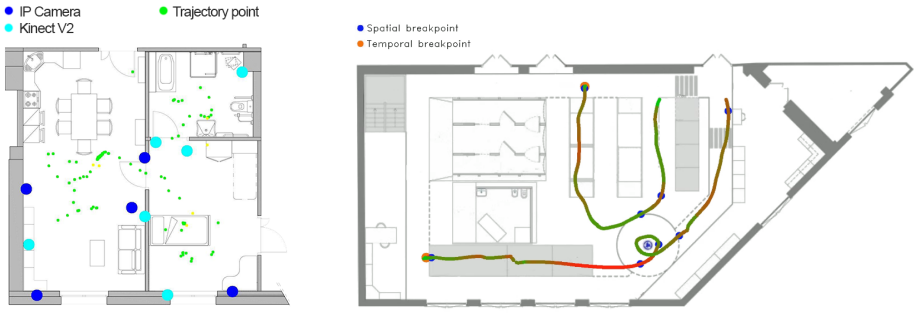


Figure 4.21: Raw user trajectory data points (green) (left, apartment) and final trajectory with metrics (right, gym).

sitional data. As an example, in Fig. 4.22 a heatmap of the daily occupation of the AUSILIA domotic apartment is shown. Other possible visualisation can be built from this type of data, such as heatmap representing interactions between patients and caregivers, or danger zones for social distancing, which in these days is a hot topic indeed [109].
By combining both the room occupancy and the wearable and environmental sensor data, it is possible to keep track of the users' habits, and even automatically detect and label them once they occur. Many other works, such as [3], tackle this problem, although with some manual intervention for labeling.

Sensor data integration

User identification, location, and tracking information are integrated with other wearable and environmental sensor data. All collected data, previously filtered, recorded, and synchronised, allow a final presentation with all the information acquired. It provides a detailed overview of all the needed information to better estimate user activities. These notions are then displayed simultaneously through a dedicated web page and an immersive interface.

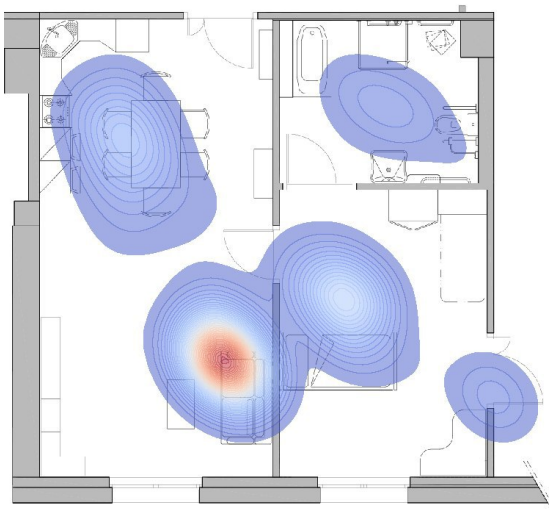


Figure 4.22: An example of the generated occupancy heat-map for the AUSILIA domotic apartment.

4.3.5 Presentation subsystem

From a data presentation viewpoint, the proposed expert system provides three possible levels of access to different type of data, namely:

- Debugging sensors and modules signal through the system management console
- Accessing quantitative data, as well as video streams and metrics for each environment through a web-based dashboard
- Accessing qualitative data, in the form of 3D reconstructions of the environment and users actions, through the usage of a HTC Vive [10] head-mounted display (HMD).

Every level of data access is further detailed in the following sections.

[10] Htc Vive head-mounted display `https://www.vive.com/us/`.

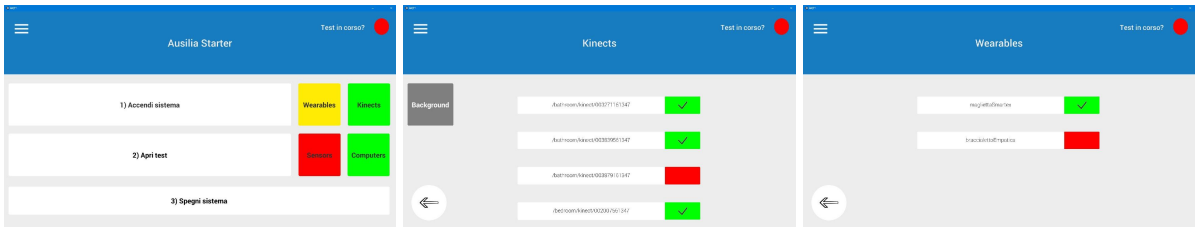


Figure 4.23: Example screenshots from the Unity dashboard to turn on/off the system and its components.

System management console

Each room and scenario of the AAL environment can be accessed through a system management console, which has been written using the Unity game engine, to maximise its portability through devices and operating systems, including WebGL frames. This makes it possible to run the system management console even from a browser or a simple smartphone device. The main purpose of the console is to check the status of each sensor and camera inside the environment, as well as providing additional crafted functionalities, such as preventing the usage of an environment if there are ongoing tests or if critical components are not functional or not responding. An example of the system management console dashboards can be seen in Fig. 4.23.

Web-based data presentation

While the system management console can be used to check the status of the entire system, including sensors and cameras for each environment, the web-based interface can be used to access data coming from the different tests, for each user in the system (Fig. 4.24). Each member of the medical staff can access, through a login screen, to different pools of patients, based on their assignments. Patient data can be shared among different members of the medical staff if required. An example of possible output data that the users have access to while navigating the web dashboard is reported in table 4.3.

Datum	Explanation
R_p	The patients' p registry
T_p	A list of tests for each patient p
$E_{p,t}$	A list of environments used for each test t
$V_{p,e,t}$	A list of video sequences for each utilised environment e in each test t
$S_{p,e,t}$	A list of sensor data sequences (wearable and environ-mental) for each environment e in each test t
$H_{p,e,t}$	A occupational heatmap for each environment e in each test t
$other$	Any kind of data that requires additional visualisation

Table 4.3: Example of possible output data that can be visualised through the web-based dashboard

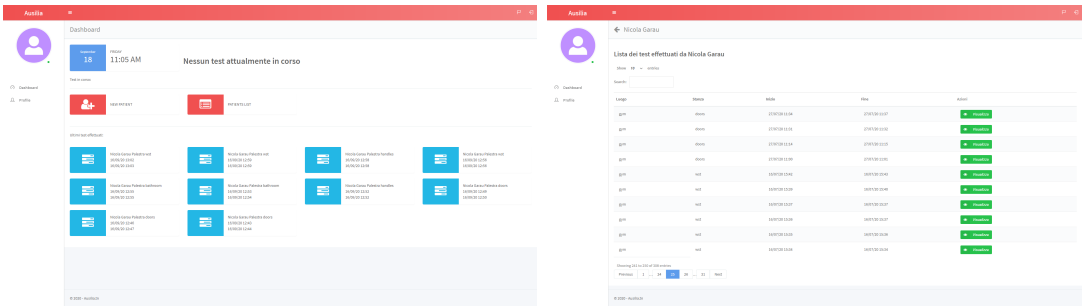


Figure 4.24: Home screen and list of tests from the web dashboard.

HMD-based data presentation

The HMD used by the therapists can be seen as an enhanced display screen in front of their eyes. This device allows them to enter a digital world and fully immerse themselves in it with all the information need to increase the objectivity and effectiveness of their clinical observation.

From the scale of Mixed Reality (MR), which refers to the fusion of the real and virtual world, our context is in Augmented Virtuality (AV) field. The patient's real data and the therapists themselves are projected into a virtual world and participate in it. The virtual environment is a simulated model of the real environment.

The main advantage of AV lies in the possibility for patients to be independent and for therapists to assess the patients' status from an augmented view. Indeed, it avoids the possibility of distraction and influencing the patient's behaviour due to the presence of an external observer. Furthermore, in addition to the evidence provided by the human eye, a large amount of data can be provided to the therapist.

The therapist's augmentation can be related to the patient's motion in 3D, their interaction with the environment and their physiological parameters with the data acquired by the multiple 3D cameras, the different environmental sensors and the wearable devices we discussed in section 4.

The use of innovative technologies using MR methodologies, once the device and what parameters of interest to show were selected, make it necessary to provide data with appropriate feedback to the final user. Only an intuitive, effective and user-friendly interface makes these technologies a useful tool, without affecting the therapist's mental effort. Therapists have control of the entire virtual scene aided by the use of the HTC Vive controller. In addition, it is possible to navigate within the environment with natural gestures (e.g. looking around, moving, etc.). In particular, they can decide for each action of the patient, which

parameters to display, at what time and from what position and point of view.

In the 3D domain, navigation is much more effective in terms of speed and truthfulness. It is possible to directly estimate the 3D shape of objects in the scene and link the corresponding information. Contextualising the measured quantities with the area in which these data were collected allows easy interpretation due to the natural link between cause and effect. Some data are displayed through 2D panels within the scene with a pre-set graphic style. These panels contain static diagrams but with the possibility to enables exploration over time.

The medium used to provide information is not only related to visual feedbacks, but further possibilities are 3D sound recreated for the patient's heartbeat and breath or haptic feedback driven by the 3D interaction between the controller and the objects displayed in the virtual scene. The combination of these stimuli suggests to the therapists a natural correlation of different quantities with respect to the patient's status.

4.3.6 Experimental results

Objectively assessing the patients performances solely on the basis of pre-defined metrics (e.g. room occupancy, required OTs assistance) is not a common practice, because each patient presents different needs and variable degrees of independence, to name a few. For this reason, we argue that testing expert systems, such as the ones proposed in this work, requires additional supervision by the experts, namely the medical personnel (in this case mainly doctors and OTs).

Living lab setup and experiments configuration

In order to assess the capabilities of the presented expert system, we designed a series of experiments which involved different actors, each presenting a different role and set of interactions, as shown in Tab. 4.4.

All the tests have been conducted in a blind way, meaning that only one of

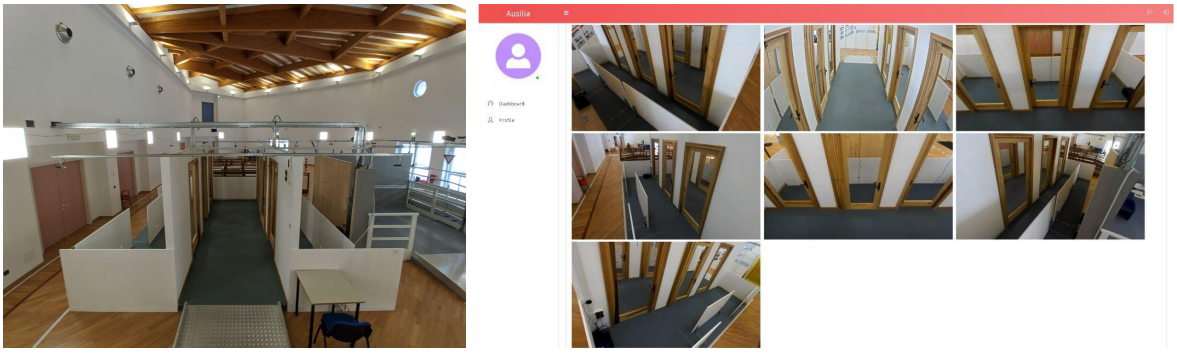


Figure 4.25: Ausilia's Doors environment.

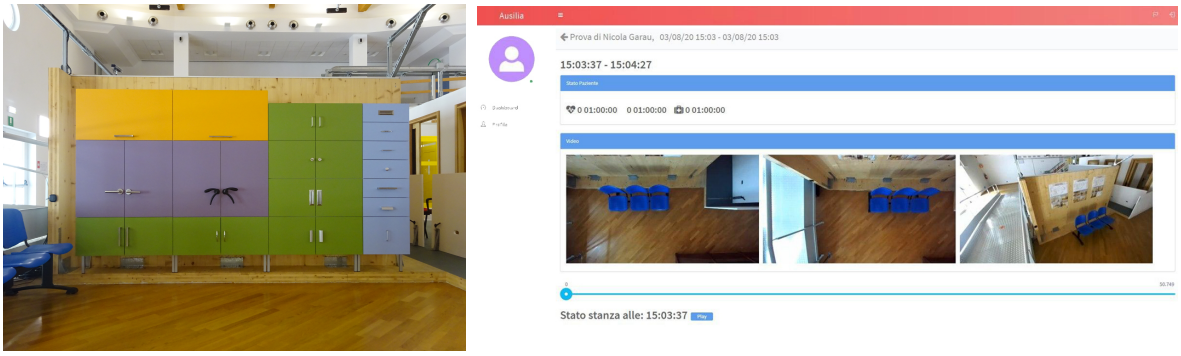


Figure 4.26: Ausilia's Handles environment.

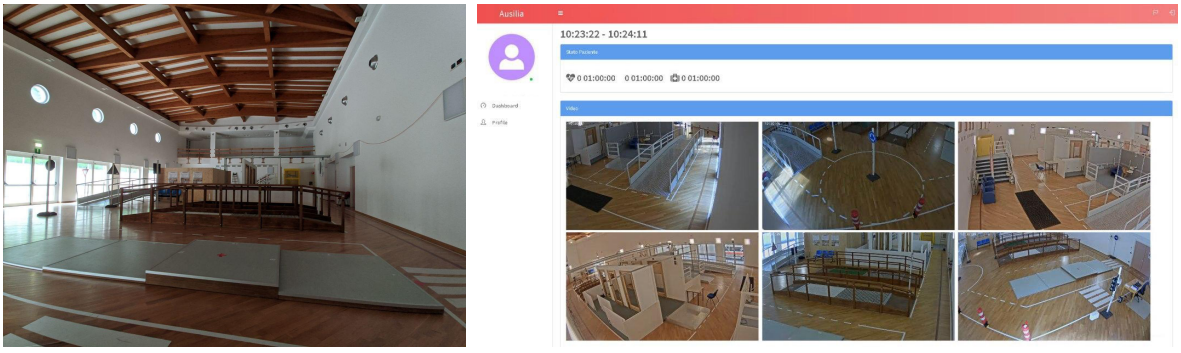


Figure 4.27: Ausilia's WST environment.

	Description	Interacts with
P	The patient P is assisted, if needed, by the occupational therapist OT1 during its stay inside the domotic apartment DA.	[OT1, DA]
OT1	The occupational therapist OT1 is the only person that can directly assist the patient P inside the domotic apartment DA.	[P, DA]
OT2	The occupational therapist OT2 has a blind role, meaning that their only knowledge of the patient P is provided by the expert system ES.	[ES]
DA	The domotic apartment DA hosts the patient P and the occupational therapist OT1, simultaneously interfacing with the expert system ES.	[P, OT1, ES]
ES	The expert system ES interfaces with the domotic apartment DA and the occupational therapist OT2, collecting and organising information from the former, which can be then accessed by the latter	[OT2, DA]

Table 4.4: Actors and their roles in the conducted experiments

the two occupational therapists (*OT1*) was allowed to interact with the patient *P* inside the domotic apartment *DA*, while the other (*OT2*) had to stay outside the whole environment, and monitor everything by only using the web dashboard and the 3D virtual reality visualisation provided by our expert system *ES*. At the end of each test, they were required to fill a standard report describing the performances of the user for each test. The desired outcome of the conducted test was to minimise the distance between the evaluations of patient *P* by the occupational therapists *OT1* and *OT2*, despite the drastically different knowledge sources (directly provided by [*P*, *DA*] for *OT1*, indirectly provided by *ES* for *OT2*).

Preliminary results

Results showed that the proposed expert system can be of great help in aiding the medical staff to objectively evaluate the patients' performances. Moreover, having the possibility of re-evaluating tests after their completion in an offline fashion made it possible for *OT2* to take into considerations some little behavioural details that *OT1* was unable to grasp in person.

A very important factor of the possibility of remotely evaluating patients performances is fostering the complete independence of the user inside the AAL environment, without any help from any occupational therapist, thus accurately reproducing a realistic standard home scenario.

Data collection and quantitative results

The proposed expert system has been used inside and tailored for the AUSILIA project [146], inside the Villa Rosa hospital in Pergine, Italy. During the past years, we worked alongside the medical personnel in order to refine the required metrics, inputs and outputs. During the first year of the system's experimental phase, 52 volunteer patients have been interacting with the system. Each patient stayed in the AAL environment for a total of 2-3 days, mainly during daytime day hospital hours. For this reason, most of the data has been recorded in the living room, kitchen and bathroom. As a result, until today the presented expert system was able to automatically collect and annotate the following data, that is available for consultation by the medical personnel for additional analysis:

- Over 5K video sequences, annotated by zone, test type and patient present in the video
- Over 5K pose and trajectory sequences (Fig. 4.21), for both the patients and for the OTs
- Corresponding occupancy maps (Fig. 4.22) for each conducted test
- More than 5K annotated biometric and environmental data sequence
- A 3D point cloud sequence for each conducted test

In total, approximately 5 terabytes of data have been collected during the past year of testing. All the past tests can be accessed at any time by the authorised medical personnel through the web dashboard, in order to study, evaluate and

re-evaluate certain conditions through quantitative metrics that were previously impossible to objectively verify. The presented expert system can thus be used as an additional occupational therapy tool for understanding how to provide the best possible help to new patients, based on previous similar cases.

Scalability and replicability

The proposed expert system is currently adopted by the AUSILIA [11] living lab (TN, Italy). The AUSILIA's domotic apartment is currently fully operational and self-sustaining. Similarly, the AUSILIA's gym (Figs. 4.25, 4.26, 4.27) has been also finalized from both an architectural and sensorial viewpoint. It is worth recalling that the AUSILIA framework is scalable by design, and it is conceived to allow the integration of new modules, based on the environment and clinical requirements. At the same time, it allows deploying only a subset of technologies and analytics upon need.

As to the possibility of recreating the proposed infrastructure, it has to be pointed out that the proposed systems is fully replicable. The hardware setup is mostly made of general-purpose off-the-shelf components, widely available on the market. The software architecture is custom, but it is built upon established open-source frameworks and libraries. The most complex part is the design of the physical environments, which requires adequate expertise in architecture, design and ergonomy. The following points have to be carefully taken into account:

- The location should be carefully selected to guarantee safe and controlled access to patients and medical doctors.
- The environment should be largely reconfigurable: fixed spaces would not allow generating custom settings for a specific patient that are both realistic and challenging. This is crucial for a significant simulations.

[11] Project AUSILIA (Assisted Unit for Simulating Independent LIving Activities) `http://ausilia.tn.it`.

- It is not necessary to implement all the living environments, the selection may be based on the scope of the analysis. For instance, kitchen and bathroom are active and risky areas in the home environment, and may provided sufficient information for many user studies.
- The sensing infrastructure should be carefully planned according to the goals of the analysis: a preliminary study with medical doctors is necessary to identify the significant parameters to be extracted, and to select the most appropriate sensors accordingly (tables provided in Sect. 5 could be used as a reference).
- Data visualisation tools are of paramount importance for the therapists: the quantity of data collected by the system is overwhelming, therefore it is necessary to select, contextualise and simplify the generated information flow through appropriate interfaces and devices.

4.3.7 Discussion

We presented an expert system targeting ambient-assisted living scenarios, which is fully automatic and capable of monitoring, recording, annotating and storing the users' daily activities. The performance level of each activity is expressed in the form of objective metrics, automatically calculated from a combination of raw video and sensor data. The metrics are specifically tailored to target the pre-filling of standard user cards, which the OTs are usually required to compile by hand. The proposed expert system allows therefore to significantly reduce the patients' performance evaluation efforts. At the same time, it can be used as an assistive technology for helping in prescribing technical aids specifically tailored to the different needs of each patient.

Chapter 5

Conclusions and Future Research

In this thesis we focused on the problem of human pose estimation from images and videos. We showed how to use it to build dynamic camera networks calibration pipelines that exploit 3D human poses and meshes to re-calibrate each camera at run-time. We focused on improving existing camera pose estimation networks by exploiting full pinhole camera geometry for the joints prediction instead of weak perspective camera models.

We also showed how to design better HPE networks by using capsule networks that can explicitly model the viewpoint information and infer it directly from data and not as an additional task. Furthermore, we proposed Agglomerator, a new learning paradigm inspired by the cortical columns in the brain that can learn to build parse trees of part-whole hierarchies in an unsupervised way. We demonstrate how these networks can dynamically parse so called ”islands of agreement” from unlabeled input data. We compare our results with state-of-the-art capsule networks to show how the same good properties of capsule networks, including viewpoint-equivariance, can be retained.

We believe that Agglomerator-like models might constituite the next logical step to build neural networks that can better mimic the functioning of the human brain. Accordingly, we believe that human pose estimation could benefit a lot from this kind of architectures. Being able to parse articulated human bodies

in an unsupervised way with part-whole hierarchies could drastically simplify the network architecture, as well as reduce the amount of training data and implicitly learning the relationships between the viewpoint and the observed scene.

As for possible future research directions, a natural continuation of the presented thesis could be the one of building HPE networks that can mimic the respective category-selective regions in the brain. These networks should present a GLOM-like structure able to dynamically parse multiple people in a scene as well as building part-whole hierarchies of body parts and joints for each detected person.

Bibliography

[1] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos. A survey on ambient intelligence in healthcare. *Proceedings of the IEEE*, 101(12):2470–2494, 2013.

[2] Dolores Albarracin and Robert S Wyer Jr. The cognitive impact of past behavior: influences on beliefs, attitudes, and future behavioral decisions. *Journal of personality and social psychology*, 79(1):5, 2000.

[3] Hande Alemdar, Halil Ertan, Ozlem Durmaz Incel, and Cem Ersoy. Aras human activity datasets in multiple homes with multiple residents. In *2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*, pages 232–235. IEEE, 2013.

[4] Marie Alm, Helena Saraste, and Cecilia Norrbrink. Shoulder pain in persons with thoracic spinal cord injury: prevalence and characteristics. *Journal of rehabilitation medicine*, 40(4):277–283, 2008.

[5] Leandro Ricardo Altimari, José Luiz Dantas, Marcelo Bigliassi, Thiago Ferreira Dias Kanthack, Antonio Carlos de Moraes, and Taufik Abrão. Influence of different strategies of treatment muscle contraction and relaxation phases on emg signal processing and analysis during cyclic exercise. In *Computational Intelligence in Electromyography Analysis-A Perspective on Current Applications and Future Challenges*. IntechOpen, 2012.

[6] Michalis Anastasopoulos, Dirk Niebuhr, Christian Bartelt, Jan Koch, and Andreas Rausch. Towards a reference middleware architecture for ambient intelligence systems. In *ACM conference on object-oriented programming, systems, languages, and applications*. San Diego, USA, 2005.

[7] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1014–1021, 2009.

[8] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3395–3404, 2019.

[9] Filipe Assunção, Nuno Lourenço, Penousal Machado, and Bernardete Ribeiro. Denser: deep evolutionary network structured representation. *Genetic Programming and Evolvable Machines*, 20(1):5–35, 2019.

[10] Daniel Berman, Nonie Finlayson, and Julie Golomb. Depth preferences of category-selective regions in human visual cortex. *Journal of Vision*, 16(12):517–517, 2016.

[11] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12506–12515, 2020.

[12] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.

[13] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

[14] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[15] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[16] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 561–578. Springer International Publishing, October 2016.

[17] P. Bonato. Wearable sensors and systems. *IEEE Engineering in Medicine and Biology Magazine*, 29(3):25–36, 2010.

[18] Isidro III Butaslac, Alessandro Luchetti, Edoardo Parolin, Yuichiro Fujimoto, Masayuki Kanbara, Mariolino De Cecco, and Hirokazu Kato. The feasibility of augmented reality as a support tool for motor rehabilitation. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pages 165–173. Springer, 2020.

[19] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[20] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

[21] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[22] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742, 2016.

[23] Jianfeng Chen, Jianmin Zhang, Alvin Kam, and Louis Shue. An automatic acoustic bathroom monitoring system. pages 1750–1753, 01 2005.

[24] Junyang Chen, James Cremer, Kasra Zarei, Alberto Segre, and Philip Polgreen. Using computer vision and depth sensing to measure healthcare worker-patient contacts and personal protective equipment adherence within hospital rooms. *Open Forum Infectious Diseases*, 3:ofv200, Dec 2015.

[25] Shih Lun Chen, Ho-Yin Lee, Chiung-An Chen, Hong-Yi Huang, and Ching-Hsing Luo. Wireless body sensor network with adaptive low-power design for biometrics and healthcare applications. *Systems Journal, IEEE*, 3:398–409, Jan 2010.

[26] Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3d human pose estimation in videos. *arXiv preprint arXiv:2002.10322*, 2020.

[27] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[28] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S.

Huang, and Lei Zhang. Bottom-up higher-resolution networks for multi-person pose estimation. *CoRR*, abs/1908.10357, 2019.

[29] Yu Cheng, Bo Yang, Bo Wang, and Robby T Tan. 3d human pose estimation using spatio-temporal networks with explicit occlusion training. *arXiv preprint arXiv:2004.11822*, 2020.

[30] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2262–2271, 2019.

[31] Philippa Clarke, Victor Marshall, Sandra E Black, and Angela Colantonio. Well-being after stroke in canadian seniors: findings from the canadian study of health and aging. *Stroke*, 33(4):1016–1021, 2002.

[32] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

[33] Adelina Comas-Herrera, Raphael Wittenberg, Joan Costa-Font, Cristiano Gori, Alessandra Di Maio, Concepcio Patxot, Linda Pickard, Alessandro Pozzi, and Heinz Rothgang. Future long-term care expenditure in germany, spain, italy and the united kingdom. *Ageing & Society*, 26(2):285–302, 2006.

[34] James M Coughlan and Alan L Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In *Advances in Neural Information Processing Systems*, pages 845–851, 2001.

[35] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[36] KA Curtis, KE Roach, E Brooks Applegate, T Amar, CS Benbow, TD Genecco, and J Gualano. Development of the wheelchair user's shoulder pain index (wuspi). *Spinal Cord*, 33(5):290, 1995.

[37] Kathleen A Curtis, George A Drysdale, R David Lanza, Morey Kolber, Richard S Vitolo, and Ronald West. Shoulder pain in wheelchair users with tetraplegia and paraplegia. *Archives of physical medicine and rehabilitation*, 80(4):453–457, 1999.

[38] Anis Davoudi, Kumar Rohit Malhotra, Benjamin Shickel, Scott Siegel, Seth Williams, Matthew Ruppert, Emel Bihorac, Tezcan Ozrazgat-Baslanti, Patrick J. Tighe, Azra Bihorac, and Parisa Rashidi. Intelligent icu for autonomous patient monitoring using pervasive sensing and deep learning. *Scientific Reports*, 9(1):8020, May 2019.

[39] Carlo J De Luca, L Donald Gilmore, Mikhail Kuznetsov, and Serge H Roy. Filtering the surface emg signal: Movement artifact and baseline noise contamination. *Journal of biomechanics*, 43(8):1573–1579, 2010.

[40] Kevin Desai, Balakrishnan Prabhakaran, and Suraj Raghuraman. Skeleton-based continuous extrinsic calibration of multiple rgb-d kinect cameras. In *Proceedings of the 9th ACM Multimedia Systems Conference*, pages 250–257. ACM, 2018.

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[42] Marco Di Rienzo, Francesco Rizzo, Gianfranco Parati, Gabriella Brambilla, Maurizio Ferratini, and Paolo Castiglioni. Magic system: a new textile-based wearable device for biological signal monitoring. applicability in daily life and clinical setting. *Conference proceedings : ...*

Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference, 7:7167–9, 02 2005.

[43] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[44] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[45] Kaiwen Duan, Lingxi Xie, Honggang Qi, Song Bai, Qingming Huang, and Qi Tian. Location-sensitive visual recognition with cross-iou loss. *CoRR*, abs/2104.04899, 2021.

[46] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[47] Andrea D'Eusanio, Stefano Pini, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Refinet: 3d human pose refinement with depth maps. 01 2021.

[48] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *European Conference on Computer Vision (ECCV)*, 2018.

[49] Gert Faber, Idsart Kingma, Chien-Chi Chang, Jack Dennerlein, and Jaap Van Dieen. Validation of a wearable system for 3d ambulatory l5/s1

moment assessment during manual lifting using instrumented shoes and an inertial sensor suit. *Journal of Biomechanics*, 102:109671, 01 2020.

[50] Margaret A Finley and Mary M Rodgers. Prevalence and identification of shoulder pathology in athletic and nonathletic wheelchair users with shoulder pain: A pilot study. *Journal of Rehabilitation Research & Development*, 41, 2004.

[51] Lars Fuglsang, Anne Vorre Hansen, Ines Mergel, and Maria Taivalsaari Røhnebæk. Living labs for public sector innovation: An integrative literature review. *Administrative Sciences*, 11(2):58, 2021.

[52] Michael Fürst, Shriya T. P. Gupta, René Schuster, Oliver Wasenmüller, and Didier Stricker. HPERL: 3d human pose estimation from RGB and lidar. *CoRR*, abs/2010.08221, 2020.

[53] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1050–1059. JMLR.org, 2016.

[54] Nicola Garau and Nicola Conci. Unsupervised continuous camera network pose estimation through human mesh recovery. In *Proceedings of the 13th International Conference on Distributed Smart Cameras*, ICDSC 2019, New York, NY, USA, 2019. Association for Computing Machinery.

[55] Nicola Garau, Francesco G. B. De Natale, and Nicola Conci. Fast automatic camera network calibration through human mesh recovery. *Journal of Real-Time Image Processing*, Sep 2020.

[56] Nicola Garau, Giulia Martinelli, Piotr Bròdka, Niccoló Bisagno, and Nicola Conci. Panoptop: a framework for generating viewpoint-invariant human pose estimation datasets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2021.

[57] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster. Automatic camera and range sensor calibration using a single shot. In *2012 IEEE International Conference on Robotics and Automation*, pages 3936–3943, 2012.

[58] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. JMLR Workshop and Conference Proceedings.

[59] Colin Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(2):285–321, 1991.

[60] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[61] Andrea Grisenti, Giandomenico Nollo, Michela Dalprá, Francesco De Natale, Mariolino De Cecco, Andrea Francesconi, Alberto Fornaser, Paolo Tomasin, Nicola Garau, Luca Guandalini, Barbara Bauer, Barbara Gasperini, Patrizia Ianes, Francesco Pilla, and Giovanni Guandalini. Technological infrastructure supports new paradigm of care for healthy aging: The living lab ausilia. In Andrea Monteriù, Alessandro Freddi, and Sauro Longhi, editors, *Ambient Assisted Living*, pages 85–99, Cham, 2021. Springer International Publishing.

[62] Hengkai Guo, Guijin Wang, Xinghao Chen, and Cairong Zhang. To-

wards good practices for deep 3d hand pose estimation. *arXiv preprint arXiv:1707.07248*, 2017.

[63] 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*, pages 160–177. Springer, 2016.

[64] Panu Harmo, Tapio Taipalus, Jere Knuuttila, J. Vallet, and Aarne Halme. Needs and solutions - home automation and service robots for the elderly and disabled. pages 3201 – 3206, 09 2005.

[65] Jeff Hawkins. A thousand brains: A new theory of intelligence, 2021.

[66] Jeff Hawkins, Subutai Ahmad, and Yuwei Cui. A theory of how columns in the neocortex enable learning the structure of the world. *Frontiers in neural circuits*, 11:81, 2017.

[67] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[68] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[69] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[70] Antonio Hernandez-Vela, Nadezhda Zlateva, Alexander Marinov, Miguel Reyes, Petia Radeva, Dimo Dimov, and Sergio Escalera. Graph cuts optimization for multi-limb human segmentation in depth maps. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 726–732, 2012.

[71] Gines Hidalgo, Yaadhav Raaj, Haroon Idrees, Donglai Xiang, Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Single-network whole-body pose estimation. *arXiv preprint arXiv:1909.13423*, 2019.

[72] Geoffrey Hinton. How to represent part-whole hierarchies in a neural network. *arXiv preprint arXiv:2102.12627*, 2021.

[73] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[74] Geoffrey E Hinton. Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, 46(1-2):47–75, 1990.

[75] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International conference on artificial neural networks*, pages 44–51. Springer, 2011.

[76] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with EM routing. In *International Conference on Learning Representations*, 2018.

[77] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matthew Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. A perceptual measure for deep single image camera calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2354–2363, 2018.

[78] Kjell Jørgen Hole and Subutai Ahmad. A thousand brains: toward biologically constrained ai. *SN Applied Sciences*, 3(8):1–14, 2021.

[79] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

[80] Weiting Huang, Pengfei Ren, Jingyu Wang, Qi Qi, and Haifeng Sun. AWR: adaptive weighting regression for 3d hand pose estimation. *CoRR*, abs/2007.09590, 2020.

[81] Ryo Inomata, Kenji Terabayashi, Kazunori Umeda, and Guy Godin. Registration of 3d geometric model and color images using sift and range intensity images. In *International Symposium on Visual Computing*, pages 325–336. Springer, 2011.

[82] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[83] Aaron S. Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric CNN regression. *CoRR*, abs/1703.07834, 2017.

[84] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015.

[85] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[86] M Esat Kalfaoglu, Sinan Kalkan, and A Aydin Alatan. Late temporal modeling in 3d cnn architectures with bert for action recognition. In *European Conference on Computer Vision*, pages 731–747. Springer, 2020.

[87] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018.

[88] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Computer Vision and Pattern Regognition (CVPR)*, 2018.

[89] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5614–5623, 2019.

[90] Isinsu Katircioglu, Bugra Tekin, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Learning Latent Representations of 3D Human Pose with Deep Neural Networks. *International Journal of Computer Vision*, 126(12):1326–1341, 2018.

[91] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. 26(3):24–es, July 2007.

[92] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), July 2013.

[93] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021.

[94] H. Kim and K. S. Hong. Practical self-calibration of pan-tilt cameras. *IEE Proceedings - Vision, Image and Signal Processing*, 148(5):349–355, 2001.

[95] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[96] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. VIBE: video inference for human body pose and shape estimation. *CoRR*, abs/1912.05656, 2019.

[97] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[98] Lena von Koch, Annica Wohlin Wottrich, and Lotta Widén Holmqvist. Rehabilitation in the home versus the hospital: the importance of context. *Disability and rehabilitation*, 20(10):367–372, 1998.

[99] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. *arXiv preprint arXiv:1909.12828*, 2019.

[100] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. *CoRR*, abs/1909.12828, 2019.

[101] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019.

[102] Adam Roman Kosiorek, Sara Sabour, Yee Whye Teh, and Geoffrey Hinton. Stacked capsule autoencoders. 2019.

[103] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[104] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[105] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[106] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[107] Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–104. IEEE, 2004.

[108] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR, 2019.

[109] Joseph A Lewnard and Nathan C Lo. Scientific and ethical basis for social-distancing interventions against covid-19. *The Lancet. Infectious diseases*, 20(6):631, 2020.

[110] Jiachen Li, Ali Hassani, Steven Walton, and Humphrey Shi. Convmlp: Hierarchical convolutional mlps for vision. *arXiv preprint arXiv:2109.04454*, 2021.

[111] Daniele Liciotti, Marina Paolanti, Emanuele Frontoni, Adriano Mancini, and Primo Zingaretti. *Person Re-identification Dataset with RGB-D*

Camera in a Top-View Configuration, pages 1–11. Springer International Publishing, Cham, 2017.

[112] Jiahao Lin and Gim Hee Lee. Trajectory space factorization for deep video-based 3d human pose estimation. *arXiv preprint arXiv:1908.08289*, 2019.

[113] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2021.

[114] J. Liu, H. Ding, A. Shahroudy, L. Duan, X. Jiang, G. Wang, and A. C. Kot. Feature boosting network for 3d pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):494–501, 2020.

[115] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[116] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):248, 2015.

[117] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014.

[118] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[119] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[120] Natalia López, Sergio Ponce, David Piccinini, Elisa Berenguer, and Martin Roberti. From hospital to home care: Creating a domotic environment for elderly and disabled people. *IEEE Pulse*, 7:38–41, 05 2016.

[121] Andy J. Ma, Nishi Rawat, Austin Reiter, Christine Shrock, Andong Zhan, Alex Stone, Anahita Rabiee, Stephanie Griffin, Dale M. Needham, and Suchi Saria. Measuring patient mobility in the icu using a novel noninvasive sensor. *Critical care medicine*, 45(4):630–636, Apr 2017.

[122] Walter Maetzler, Josefa Domingos, Karin Srulijes, Joaquim Ferreira, and Bas Bloem. Quantitative wearable sensors for objective assessment of parkinson's disease. *Movement disorders : official journal of the Movement Disorder Society*, 28, 10 2013.

[123] Manuel J. Marín-Jiménez, Francisco J. Romero Ramírez, Rafael Muñoz-Salinas, and Rafael Medina Carnicer. 3d human pose estimation from depth maps using a deep combination of poses. *CoRR*, abs/1807.05389, 2018.

[124] Vittorio Mazzia, Francesco Salvetti, and Marcello Chiaberge. Efficient-capsnet: Capsule network with self-attention routing. *arXiv preprint arXiv:2101.12491*, 2021.

[125] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. Xnect: Real-time multi-person 3d motion capture with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 39(4):82–1, 2020.

[126] Jennifer L Mercer, Michael Boninger, Alicia Koontz, Dianxu Ren, Trevor Dyson-Hudson, and Rory Cooper. Shoulder joint kinetics and pathology in manual wheelchair users. *Clinical Biomechanics*, 21(8):781–789, 2006.

[127] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[128] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[129] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[130] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.

[131] Shogo Miyata, Hideo Saito, Kosuke Takahashi, Dan Mikami, Mariko Isogawa, and Akira Kojima. Extrinsic camera calibration without visible corresponding points using omnidirectional cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(9):2210–2219, 2018.

[132] Christoph Molnar. *Interpretable Machine Learning*. 2019.

[133] 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 *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018.

[134] Rinat Mukhometzianov and Juan Carrillo. Capsnet comparative performance evaluation for image classification. *arXiv preprint arXiv:1805.11195*, 2018.

[135] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable

machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[136] Alejandro Newell and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *CoRR*, abs/1611.05424, 2016.

[137] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. *CoRR*, abs/1603.06937, 2016.

[138] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. Ieee, 2004.

[139] Davide Ortenzi, Flavia Benetazzo, Francesco Ferracuti, Alessandro Freddi, Andrea Giantomassi, Sabrina Iarlori, Sauro Longhi, and Andrea Monteriù. *AAL Technologies for Independent Life of Elderly People*, volume 11. 05 2015.

[140] Barış Özcan, Furkan Kınlı, and Furkan Kıraç. Quaternion capsule networks. *arXiv preprint arXiv:2007.04389*, 2020.

[141] M. Pacelli, G. Loriga, N. Taccini, and R. Paradiso. Sensing fabrics for monitoring physiological and biomechanical variables: E-textile solutions. In *2006 3rd IEEE/EMBS International Summer School on Medical Devices and Biosensors*, pages 1–4, 2006.

[142] M. Paolanti, R. Pietrini, A. Mancini, E. Frontoni, and P. Zingaretti. Deep understanding of shopper behaviours and interactions using rgb-d vision. *Machine Vision and Applications*, 31(7-8), 2020.

[143] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from

videos. In *SIGGRAPH Asia 2018 Technical Papers*, page 178. ACM, 2018.

[144] Huy Hieu Pham, Houssam Salmane, Louahdi Khoudour, Alain Crouzil, Sergio A Velastin, and Pablo Zegers. A unified deep framework for joint 3d pose estimation and action recognition from a single rgb camera. *Sensors*, 20(7):1825, 2020.

[145] Nicola Piotto, Nicola Conci, and Francesco GB De Natale. Syntactic matching of trajectories for ambient intelligence applications. *IEEE Transactions on Multimedia*, 11(7):1266–1275, 2009.

[146] Tommaso Pisoni, Nicola Conci, Francesco GB De Natale, Mariolino De Cecco, Giandomenico Nollo, Antonio Frattari, and Giovanni MA Guandalini. Ausilia: assisted unit for simulating independent living activities. In *2016 IEEE International Smart Cities Conference (ISC2)*, pages 1–4. IEEE, 2016.

[147] Liliana [Lo Presti] and Marco [La Cascia]. 3d skeleton-based human action classification: A survey. *Pattern Recognition*, 53:130 – 147, 2016.

[148] Varun Ramakrishna, Daniel Munoz, Martial Hebert, James Andrew Bagnell, and Yaser Sheikh. Pose machines: Articulated pose estimation via inference machines. In *European Conference on Computer Vision*, pages 33–47. Springer, 2014.

[149] Iván Ramírez, Alfredo Cuesta-Infante, Emanuele Schiavi, and Juan José Pantrigo. Bayesian capsule networks for 3d human pose estimation from single 2d images. *Neurocomputing*, 379:64 – 73, 2020.

[150] Fabio Ribeiro, Georgios Leontidis, and Stefanos Kollias. Capsule routing via variational bayes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:3749–3756, 04 2020.

[151] Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[152] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3859–3869, Red Hook, NY, USA, 2017. Curran Associates Inc.

[153] Marta Sanzari, Valsamis Ntouskos, and Fiora Pirri. Bayesian image based 3d pose estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 566–582, Cham, 2016. Springer International Publishing.

[154] Maddalena Sebastiani, Nicola Garau, Francesco De Natale, and Nicola Conci. Joint trajectory and fatigue analysis in wheelchair users. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[155] Y. Seo and K. S. Hong. Theory and practice on the self-calibration of a rotating and zooming camera from two views. *IEE Proceedings - Vision, Image and Signal Processing*, 148(3):166–172, 2001.

[156] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90:106181, 2020.

[157] Saurabh Sharma, Pavan Teja Varigonda, Prashast Bindal, Abhishek Sharma, and Arjun Jain. Monocular 3d human pose estimation by generation and ordinal ranking. *CoRR*, abs/1904.01324, 2019.

[158] Saurabh Sharma, Pavan Teja Varigonda, Prashast Bindal, Abhishek Sharma, and Arjun Jain. Monocular 3d human pose estimation by gen-

eration and ordinal ranking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2325–2334, 2019.

[159] 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 *CVPR 2011*, pages 1297–1304. Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR) 2011, 2011.

[160] 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 *CVPR 2011*, pages 1297–1304, 2011.

[161] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[162] Kyle Simek. Pinhole camera diagram, dissecting the camera matrix. `http://ksimek.github.io/pinhole_camera_diagram/`, 2013. Accessed: 2019-04-26.

[163] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[164] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.

[165] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *arXiv preprint arXiv:1906.01618*, 2019.

[166] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1442–1468, 2013.

[167] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017.

[168] Velislava Spasova and Ivo Iliev. A survey on automatic fall detection in the context of ambient assisted living systems. *International Journal of Advanced Computer Research*, 4:94, 04 2014.

[169] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019.

[170] Michele Stocco, Alessandro Luchetti, Paolo Tomasin, Alberto Fornaser, Patrizia Ianes, Giovanni Guandalini, J Flores Ty, Sayaka Okahashi, Alexander Plopski, Hirokazu Kato, et al. Augmented reality to enhance the clinical eye: The improvement of adl evaluation by mean of a sensors based observation. In *International Conference on Virtual Reality and Augmented Reality*, pages 291–296. Springer, 2019.

[171] Lucas Stoffl, Maxime Vidal, and Alexander Mathis. End-to-end trainable multi-instance pose estimation with transformers. *CoRR*, abs/2103.12115, 2021.

[172] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[173] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. *CoRR*, abs/1902.09212, 2019.

[174] Motion Lab Systems. Motion lab systems- a software user guide for emg graphing and emg analysis. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11–45. Motion Lab Systems, 2009.

[175] Zheng Tang, Jenq-Neng Hwang, Yen-Shuo Lin, and Jen-Hui Chuang. Multiple-kernel adaptive segmentation and tracking (mast) for robust object tracking. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1115–1119. IEEE, 2016.

[176] Zheng Tang, Yen-Shuo Lin, Kuan-Hui Lee, Jenq-Neng Hwang, and Jen-Hui Chuang. Esther: Joint camera self-calibration and automatic radial distortion correction from tracking of walking humans. *IEEE Access*, 7:10754–10766, 2019.

[177] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3941–3950, 2017.

[178] Bugra Tekin, Xiaolu Sun, Xinchao Wang, Vincent Lepetit, and Pascal Fua. Predicting people's 3d poses from short sequences. 04 2015.

[179] Yan Tian, Wei Hu, Hangsen Jiang, and Jiachen Wu. Densely connected attentional pyramid residual network for human pose estimation. *Neurocomputing*, 347:13 – 23, 2019.

[180] Zhi Tian, Hao Chen, and Chunhua Shen. Directpose: Direct end-to-end multi-person pose estimation. *CoRR*, abs/1911.07451, 2019.

[181] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

[182] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2500–2509, 2017.

[183] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[184] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659, 2013.

[185] Shashank Tripathi, Siddhant Ranade, Ambrish Tyagi, and Amit Agrawal. Posenet3d: Unsupervised 3d human shape and pose estimation. *arXiv preprint arXiv:2003.03473*, 2020.

[186] Toon Van de Maele, Tim Verbelen, Ozan Catal, and Bart Dhoedt. Disentangling what and where for 3d object-centric representations through active inference. *arXiv preprint arXiv:2108.11762*, 2021.

[187] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[188] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.

[189] Francisco Vasconcelos, Joao P Barreto, and Edmond Boyer. Automatic camera calibration using multiple sets of pairwise correspondences. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):791–803, 2018.

[190] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[191] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7782–7791, 2019.

[192] K. Wang, L. Lin, C. Jiang, C. Qian, and P. Wei. 3d human pose machines with self-supervised learning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 42(05):1069–1082, may 2020.

[193] Keze Wang, Shengfu Zhai, Hui Cheng, Xiaodan Liang, and Liang Lin. Human pose estimation from depth images via inference embedded multi-task learning. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1227–1236, 2016.

[194] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh.

Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[195] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. *CoRR*, abs/1602.00134, 2016.

[196] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[197] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[198] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. *CoRR*, abs/1804.06208, 2018.

[199] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[200] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 793–802, 2019.

[201] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[202] Tianxu Xu, Dong An, Yuetong Jia, and Yang Yue. A review: Point cloud-based 3d human joints estimation. *Sensors*, 21:1684, 03 2021.

[203] Mao Ye, Xianwang Wang, Ruigang Yang, Liu Ren, and Marc Pollefeys. Accurate 3d pose estimation from a single depth image. In *2011 International Conference on Computer Vision*, pages 731–738, 2011.

[204] Mao Ye, Xianwang Wang, Ruigang Yang, Liu Ren, and Marc Pollefeys. Accurate 3d pose estimation from a single depth image. In *2011 International Conference on Computer Vision*, pages 731–738, 2011.

[205] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

[206] Ho Yub Jung, Soochahn Lee, Yong Seok Heo, and Il Dong Yun. Random tree walk toward instantaneous 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2467–2474, 2015.

[207] T. N. S. Tengku Zawawi, A. R. Abdullah, M. H. Jopri, T. Sutikno, N. M. Saad, and R. Sudirman. A review of electromyography signal analysis techniques for musculoskeletal disorders. *Indonesian Journal of Electrical Engineering and Computer Science*, 11(3):1136–1146, 2018.

[208] Gaopeng Zhang, Hong Zhao, Yang Hong, Yueyang Ma, Jing Li, and Huinan Guo. On-orbit space camera self-calibration based on the orthogonal vanishing points obtained from solar panels. *Measurement Science and Technology*, 29(6):065013, 2018.

[209] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000.

[210] Fangda Zhao, Toru Tamaki, Takio Kurita, Bisser Raytchev, and Kazufumi Kaneda. Marker-based non-overlapping camera calibration methods with additional support camera views. *Image and Vision Computing*, 70:46–54, 2018.

[211] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. *CoRR*, abs/1904.03345, 2019.

[212] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.

[213] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, 2018.

[214] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4966–4975, 2016.

[215] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019.