Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.1120000

D-SPDH: Improving 3D Robot Pose Estimation in Sim2Real scenario via Depth Data

ALESSANDRO SIMONI [1], GUIDO BORGHI [2], LORENZO GARATTONI [3], GIANPIERO FRANCESCA [3], and ROBERTO VEZZANI [1]

[1]Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Italy (e-mail: {name.surname}@unimore.it)
[2]Dipartimento di Educazione e Scienze Umane, University of Modena and Reggio Emilia, Italy (e-mail: {name.surname}@unimore.it)
[3]Toyota Motor Europe

Corresponding author: Guido Borghi (e-mail: guido.borghi@unimore.it).

ABSTRACT In recent years, there has been a notable surge in the significance attributed to technologies facilitating secure and efficient cohabitation and collaboration between humans and machines, with a particular interest in robotic systems. A pivotal element in actualizing this novel and challenging collaborative paradigm involves different technical tasks, including the comprehension of 3D poses exhibited by both humans and robots through the utilization of non-intrusive systems, such as cameras. In this scenario, the availability of vision-based systems capable of detecting in real-time the robot's pose is needed as a first step towards a safe and effective interaction to, for instance, avoid collisions. Therefore, in this work, we propose a vision-based system, referred to as D-SPDH, able to estimate the 3D robot pose. The system is based on double-branch architecture and depth data as a single input; any additional information regarding the state of the internal encoders of the robot is not required. The working scenario is the Sim2Real, *i.e.*, the system is trained only with synthetic data and then tested on real sequences, thus eliminating the time-consuming acquisition and annotation procedures of real data, common phases in deep learning algorithms. Moreover, we introduce SimBa^{++}, a dataset featuring both synthetic and real sequences with new real-world double-arm movements, and that represents a challenging setting in which the proposed approach is tested. Experimental results show that our D-SPDH method achieves state-of-the-art and real-time performance, paving the way a possible future non-invasive systems to monitor human-robot interactions.

INDEX TERMS Human-Machine Interaction, Human-Robot Interaction, Collaborative Robots (Cobots), Robot Pose Estimation, Deep Learning, Computer Vision, Depth Maps

I. INTRODUCTION

We are steadily advancing toward an epoch wherein humans and machines, and in particular robot systems, will coexist within various spatial and temporal contexts throughout the day, encompassing social and occupational settings. The integration of non-invasive camera surveillance in conjunction with accurate computer vision algorithms, exemplified by Robot Pose Estimation (RPE) [1] and Human Pose Estimators (HPE) [2], represents pivotal and facilitative technologies essential for ensuring the secure interaction between humans [3] and robots. For instance, the awareness about 3D positions can be used for collision detection and avoidance, or to raise alarms about imminent and unexpected events. This interaction includes a wide range of activities and robot applications, ranging from object grasping [4], robot manipulation [5], and motion planning [6].

In the context of Industry 4.0 [7], experts agree that co-operation between humans and intelligent agents [8], [9], rather than the complete removal of humans, will be a key enabler for the advancement in manufacturing [10]. In this context, safe interaction between humans and cobots is a crucial element to be investigated [11]. Moreover, in future generations of manufacturing, robots, and operators will share the workspace and have physical contact, raising new aspects related to social and physical coordination between coworkers [12], [13]: topics that are analyzed also through the robot's pose. An additional conceivable application context is exemplified by home automation, wherein robots should have the capability to execute tasks while also engaging in interactions with human occupants.

Therefore, in this paper, we focus on the development of D-SPDH, a vision-based method able to predict the robot joint positions in the 3D world relying only on depth maps as input (see Figure 1). In other words, we propose a system that is

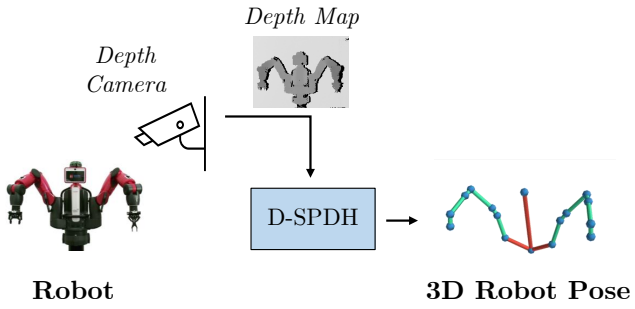


FIGURE 1: Given as input a depth image of a robot, the proposed D-SPDH method recovers the full 3D pose. This kind of non-invasive system is a key technology for safe human-machine interaction, in environments in which, for instance, workers and machines will share the same workplace.

completely agnostic about the robot's internal state, in terms of encoders, communication interfaces, and other electromechanical components, and that outputs 3D coordinates of the robot skeleton.

Specifically, with the term D-SPDH we refer to a Double-branch Semi-Perspective Decoupled Heatmaps (SPDH) [14] based on a Convolutional Neural Network (CNN) architecture trained only on depth data. We believe that using depth data only can lead to two main advantages: i) it provides information for a more accurate estimation of real-world 3D coordinates [15]; ii) it avoids illumination issues that typically affect systems based, for instance, on RGB data [16].

Each branch of the model is specialized in one of the Semi-Perspective Decoupled Heatmaps (SPDH) [14] representation, which has been proven to be effective in encoding the information to address the robot pose estimation task.

D-SPDH is developed in the Sim2Real scenario [17], *i.e.* the model is trained only on synthetic depth data, easily obtainable with simulators, and tested on real sequences. We observe that this scenario limits the difficulties in acquiring a large amount of varied and labeled depth data usually required to train deep learning-based systems [18], [19]. Besides, in this manner, the training procedure and then the method deployment are not tied to a specific acquisition depth device and its technology, which can introduce artifacts in the depth data [20] limiting generalization capabilities [21]. Furthermore, the use of depth data tends to reduce the domain gap between synthetic and real data without domain randomization or similar techniques [22].

The proposed method is evaluated on an extended version of the SimBa dataset [14] that we created called SimBa^{++}. SimBa^{++} consists of both synthetic sequences collected through Gazebo simulator [23] and real data acquired through the second version of the Microsoft Kinect depth device. This evolution of the dataset includes new challenging real-world scenes with double-arm movements.

Summarizing, the contributions of this paper are:

- We propose D-SPDH, a double-branch architecture capable of predicting reliable 3D world locations of robot joints using only images, specifically depth data. The input depth map is converted into a different depth representation (*i.e.* XYZ image, see Sect. III-A), which is fed into a backbone connected to two branches for the prediction in two different joint spaces through the SPDH representation (Sect. III-C). A visual summary of the architecture of the proposed system is given in Figure 2.
- We release SimBa^{++}, an extended version of the previous SimBa dataset [14], which is used for training, experimental evaluations, and comparisons with several baselines and competitors. As one of the first datasets in its category featuring both synthetic and real depth data with 3D annotations, we describe and analyze SimBa^{++} in detail in Sect. IV-A.
- Being aware that the presented task is quite novel in the literature, we present a comprehensive experimental evaluation of various methodologies addressing the challenge of 3D RPE. Our investigation initiates with 2D RPE, progresses to 2D to 3D projection, and culminates in a comprehensive examination of full 3D pose estimation (refer to Sect. V). Through this study, we aim to identify the main challenges and highlight prospects for future research endeavors within the domain of 3D RPE.

II. RELATED WORK

Analyzing an object's 3D location from an external camera is complex. Various approaches, shaped by recent advances in computer vision and deep learning, have emerged. Our overview of current literature on robot pose estimation categorizes works into two groups:

- *Hand-eye calibration* divided into marker-based and learning-based methods;
- *Rendering-based approaches*, which use rendering methods to project a 3D synthetic robot model into the scene and predict the pose.

Finally, a section is dedicated to currently available datasets for robot pose estimation.

A. HAND-EYE CALIBRATION

In robotics, the common approach to estimate the absolute pose of a robot with respect to the camera is the Hand-Eye Calibration [24], [25]. This approach, for instance used in [26]–[29], consists of attaching a fiducial marker (*e.g.* ArUco [30], ARTag [31], AprilTag [32]) to the end effector that is tracked through multiple frames. These algorithms exploit forward kinematics and multiple frames to solve an optimization problem using 3D-to-2D correspondences to get the camera-to-robot transform. However, these methods require physical markers on the manipulator, which is not always a feasible solution depending on the working scenario.

Nonetheless, with recent advances in human pose estimation [36], many works have been proposed to estimate the camera-to-robot pose using CNNs. We divide these approaches into two groups, depending on the input image type:

TABLE 1: Datasets available in the literature for the Robot Pose Estimation task. Further details are reported in Section II-C.

Dataset	Year	Robots	Synth	Real	Data type	Frames	Notes
CRAVES [33]	2019	1	✓	✓	RGB	5.5k	
DREAM [1]	2020	3	✓	✓	RGB	357k	
WIM [34]	2022	7	✓		RGB	140k	
CHICO [35]	2022	1		✓	RGB	≈ 1M	
SimBa [14]	2022	1	✓	✓	RGB-D	370k	Single-arm only
SimBa^{++}	2023	1	✓	✓	RGB-D	380k	Double-arm

depth-based and *RGB-based*. The first group is a minor subset in which depth data is used to predict the robot's pose. [37], taking inspiration from [38], applies a random forest classifier to the depth images to segment the links of the robot arm from which the skeleton joints are estimated. Similarly, the method described in [39] directly regresses the joint angles without the segmentation prior. However, it is noted that these methods solely retrieve the joint angles without recovering the absolute pose relative to the camera. The recent work presented in [14] proposes a new representation, referred to as SPDH, useful to predict the robot's pose through a CNN. In our work, we propose a different architecture, based on the idea of specializing each representation through a double-branch network. The superiority of our approach is highlighted in the experimental evaluation carried out both on sequences with single and double arm movements.

On the other hand, RGB-based methods represent the large majority. Lambrecht *et al.* proposes in [40] a method that combines synthetic and real data to train a keypoint localization network that predicts 2D robot joints. Computing the 3D joint configuration from the forward kinematics, a Perspective-n-Point (PnP) [41] [42] algorithm retrieves the 3D robot pose in camera coordinates. Similarly, Zuo *et al*. [33] also presents a keypoint detector but is trained on synthetic data only. Instead of PnP, they use a non-linear optimization to regress the camera pose and joint angles of a small low-cost manipulator. Recently, the work [1] demonstrates that learning-based approaches could replace classic marker-based calibration also for standard manipulators They exploit synthetic data for training, feeding RGB images into an encoder-decoder network that predicts the 2D pixel coordinates of the robot joints. The pose is computed via PnP, given the camera's intrinsic and joints' angle configuration. Moreover, Tremblay *et al*. [43] extended the previous work to retrieve the camera-to-object transform. Their pipeline consists of two networks, one for the camera-to-robot pose [1] and one for the camera-to-object pose, with the main goal of improving the grasping performance of the robot.

In contrast to the works discussed, our D-SPDH method performs direct regression of the camera-to-robot pose using a 3D pose heatmap representation. This design allows great compatibility with methods based on heatmaps, such as neural networks designed for 2D human pose estimation. Through our direct 3D regression, the proposed approach avoids the use of any PnP algorithm, making it agnostic to the robot state, including joints and angles, which may be unknown in certain instances.

B. RENDERING-BASED APPROACH

Recent works [44] [34] propose approaches based on rendering. With the growing interest in neural rendering techniques [45] [46], the goal is to use synthetic robot models and optimize the camera-to-robot pose estimation by projecting them into the scene. Labbe *et al*. [44] paves the way to this field of research presenting the first method for robot pose estimation based on the *render&compare* paradigm. The optimization algorithm refines the initial robot state, involving joint angles and the anchor part's pose relative to the camera. During testing, it can handle unknown joint angles, but with a significant drop in performance [44]. Furthermore, inspired by [46], the work of [34] proposes a self-supervised method exploiting both an explicit rough approximation of the robot body and an implicit refinement of it. To compensate for the lack of 3D pose supervision, the approach uses multi-view sequences of a moving robot with annotated masks.

Similarly, the proposed D-SPDH is a supervised learning approach. However, differently from the aforementioned methods, it operates without requiring knowledge of the robot state and does not necessitate multiple views. Furthermore, D-SPDH demonstrates notable speed advantages compared to volume rendering-based methods, as reported in Sect. VI-B). These features facilitate deployment in real-world scenarios, where information reliability is variable, and rapid processing is imperative.

C. DATASETS FOR ROBOT POSE ESTIMATION

Datasets are essential in computer vision, especially for training deep learning architectures. Unfortunately, collecting 3D annotated data for robot pose estimation in the real world is costly. An emerging solution to this problem is the use of simulators to generate synthetic data. As summarized in Table 1, only four datasets are currently available for our task and they contain exclusively RGB images: CRAVES [33], DREAM [1], WIM [34], and CHICO [35].

CRAVES is a synthetic and real dataset for the pose estimation of an OWI-535 low-cost manipulator. It contains 5k synthetic RGB images generated with Unreal Engine 4 (UE4) and background domain randomization, and 537 real RGB images with annotations for 2D keypoints and visibility. DREAM is a more complex dataset covering three robots, *i.e*. Franka Emika Panda, Kuka LBR with Allegro, and Rethink

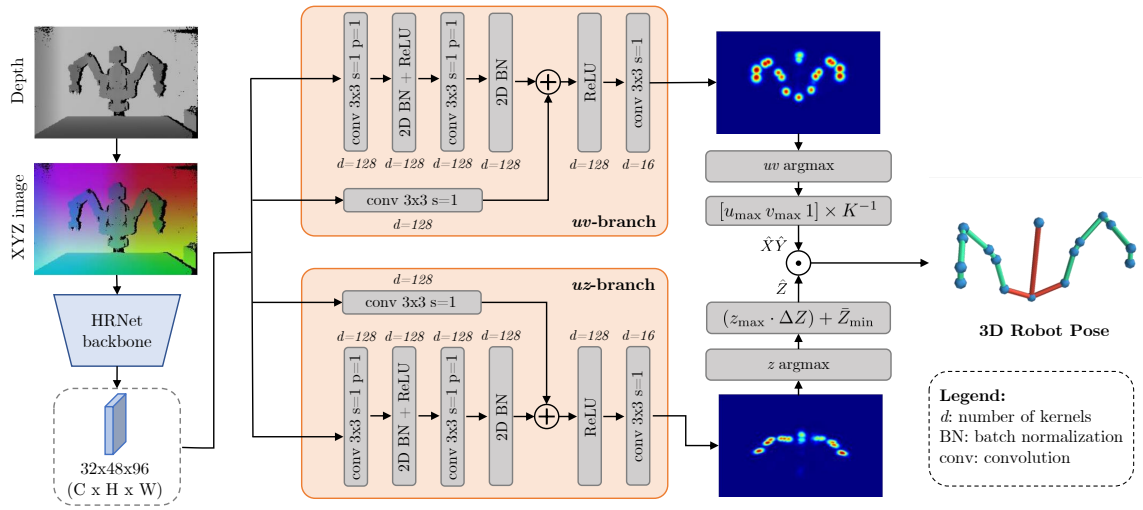
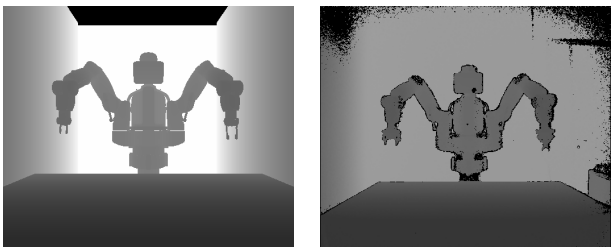


FIGURE 2: Overview of the proposed D-SPDH method: an initial depth map is firstly converted in the XYZ representation (Sect. III-A) and then used as input for the HRNet-32 [47] backbone that extracts a set of visual features elaborated separately by two branches, *i.e.* *uv*-branch and *uz*-branch (Sect. III-C). The output of each branch consists of an SPDH representation that is finally converted into the 3D robot skeleton.



(a) Synthetic Depth Image (b) Real Depth Image

FIGURE 3: Visual comparison of two depth maps depicting the same scene but acquired in two different ways: on the left is the synthetic image, obtained through the use of the Gazebo simulator, while on the right is the real depth image, acquired through the second version of the Microsoft Kinect device. As shown, these two images are visually different, for instance presenting different levels of noise (black dots) and depth accuracy.

Baxter. The synthetic data are generated with UE4 using domain randomization [48] and consist of 100k RGB images for each robot with 2D/3D keypoint locations and robot joint angles as annotations. The real data covers only the Franka Emika Panda and consists of two sets of images: Panda-3Cam containing 17k annotated RGB images collected with 3 different cameras and Panda-Orb that handles a variety of camera poses with 40k annotated RGB images collected from a RealSense camera. WIM is a smaller synthetic-only dataset generated with the python-based renderer NViSII [49] together with PyBullet [50] for animations. It contains 1k RGB frames of a synchronized video with 20 viewpoints for 7 different robots. Finally, CHICO is a real dataset for human-robot collaboration with contact and represents a benchmark for human pose forecasting and collision. It contains 240 RGB HD sequences in which 20 human operators work together with a 7-DoF KUKA LBR robot in a shared workspace.

SimBa^{++} is the sole dataset in the literature with both RGB and depth data, featuring synthetic and real sequences of a Rethink Baxter robot. It provides annotated 3D keypoint locations and camera positions, making it the first dataset suitable for RGB and depth-based approaches. It includes 350k synthetic images and 30k real images, allowing for effective domain adaptation comparisons.

III. PROPOSED METHOD

The proposed system is depicted in Figure 2 and described in the following. From a general point of view, we divide the description of the method into three parts, analyzing the Sim2Real working scenario, preparation of input depth data, and focusing on the new D-SPDH representation.

A. SIM2REAL WORKING SCENARIO

The proposed system uses depth data only. Indeed, we observe that the use of depth devices, especially if based on infrared light, represents an effective and low-priced solution to acquire 3D data robust to light changes and variations in background textures [51]. Moreover, we work in the Sim2Real scenario: during training, the input is a synthetic depth map, while in test mode it is acquired by a real depth sensor. A visual comparison between the synthetic and real depth maps is shown in Figure 3. Our objective is to operate within the demanding Sim2Real scenario, mitigating the need for labor-intensive acquisition and annotation procedures. This aims to foster the development of a system that remains independent of the specific type of depth sensor utilized. Notably, the quality of depth data, encompassing factors such as accuracy,

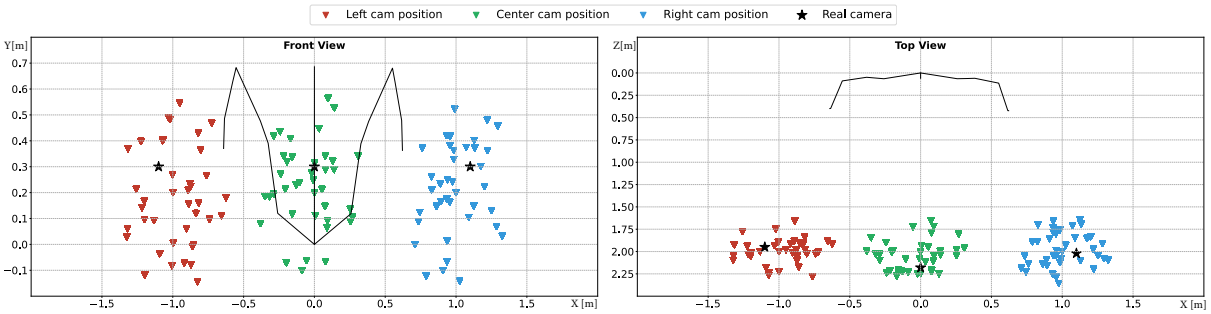


FIGURE 4: Visualization of the camera positions exploited during the acquisition procedure of the SimBa^{++} dataset with respect to the robot location (here represented through its skeleton). Different views, front-view (left) and top-view (right), of the acquisition scenes are reported, highlighting differences between the synthetic and real collection procedures. In both plots, each axis is reported in meters.

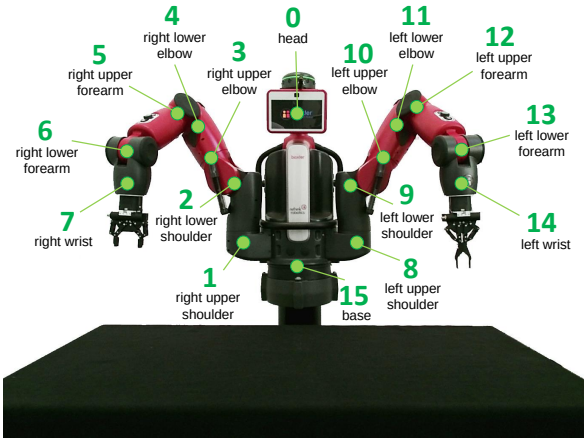


FIGURE 5: Joint locations on the Rethink Baxter robot available in the SimBa^{++} dataset and used in the experimental evaluation.

format, and resolution, is heavily influenced by the acquisition device [20]. This influence can impact the performance and generalization capabilities of vision-based systems. This is especially evident when these systems are trained and tested on images obtained from diverse depth devices or employing distinct technologies [21]. A solution consists of acquiring a great variety of depth data with a new depth sensor every time and finetuning the model on the new sequences: this unpractical and time-consuming approach leads us to investigate the Sim2Real scenario.

B. PROCESSING OF INPUT DEPTH DATA

From a formal point of view, a depth map can be defined as $D_M = \langle D, K \rangle$ where D is the measured matrix of distances d_{ij} between the acquisition device and the points in the scene, and K is the perspective projection matrix, obtained as the intrinsic parameters of the depth camera. It is worth noting that the maximum acquisition range of a real depth map relies on the technology used, mainly based on Structured Light (SL) or Time-of-Flight (ToF) [52], and on the specific sensor quality and resolution. d_{ij} is defined in the range $[r, R]$, where r and R are respectively the minimum and the maximum measurable ranges.

The input depth map is converted into an XYZ image $I_{\mathrm{D}}^{1\times H\times W} \rightarrow I_{\mathrm{XYZ}}^{3\times H\times W}$, *i.e.* a 2D representation formally defined as follows:

$$I_{\mathrm{XYZ}} = \pi(D \cdot K^{-1}) \tag{1}$$

where π is the projection in the 3D space of every value d_{ij} through the inverse of the projection matrix K. The intuition behind XYZ representation is to have an input image that limits the above-mentioned differences between synthetic and real depth data, improving the performance of the adopted model in the Sim2Real scenario. This consideration is confirmed by the experimental results reported in Sect. V.

C. INTERMEDIATE POSE REPRESENTATION THROUGH D-SPDH

To estimate the 3D pose of the robot, we first regress an intermediate representation referred to as Semi-Perspective Decoupled Heatmaps (SPDH) [14]. This representation decomposes the 3D space into two bidimensional spaces where the robot joint locations, organized as in Figure 5, are represented as heatmaps: (i) the uv space corresponding to the camera image plane, and (ii) the uz space, containing quantized values of the Z dimension of the 3D real world. The first space represents the front view of the scene in which the heatmaps H^{uv} are computed with a perspective awareness of the distance of the joints with respect to the camera: we obtain smaller Gaussians for the farthest joints to force the network to focus on those locations that are usually more difficult to predict. On the other hand, the latter space is a bird-eye view of the scene in which the heatmaps H^{uz} are obtained from a quantized portion of the Z plane of size defined as:

$$z = \frac{\bar{Z}_{max} - \bar{Z}_{min}}{\Delta Z} \tag{2}$$

where $\bar{Z} = \{\bar{Z}_i \in Z;\ \bar{Z}_{min} <= \bar{Z}_i <= \bar{Z}_{max}\}$.

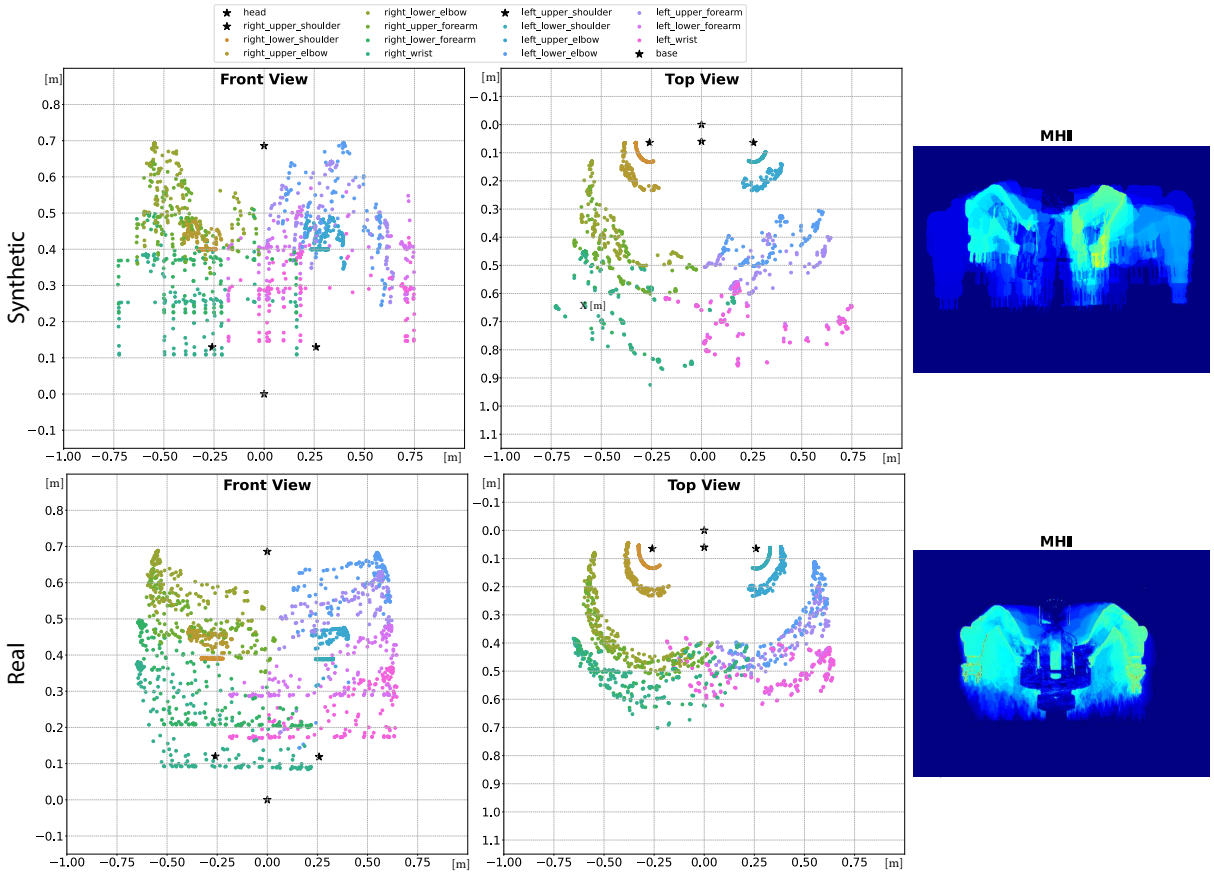


FIGURE 6: Visualization of the joints' movements in synthetic and real sequences contained in the SimBa^{++} dataset with the same camera position. The first two graphs depict each joint location through the sequences from the front and top view of the scene. In each plot, each axis is reported in meters. On the right, a Motion History Image (MHI) [53] of the robot's movements on the same sequences is presented: in this representation, brighter colors denote a high level of motion with respect to blue areas

.

The proposed approach utilizes an HRNet-32 [47] backbone, specifically leveraging the four stages while excluding the final layer. Consequently, the backbone generates visual features, serving as input for both the *uv* and *uz* branches. Each branch comprises a residual block with 128-dimensional convolutional layers, Batch Normalization (BN), and Rectified Linear Unit (ReLU) activations. A final convolutional layer within each branch predicts heatmaps, reducing the channel dimension to 16, corresponding to the number of robot joints, as illustrated in Figure 5.

The output of each branch has the same spatial dimension of $384 \times 216 \times 1$ and it is finally processed to compute the 3D robot skeleton. For each heatmap H^{uv} in the *uv* space, we compute the argmax and then the coordinates $\hat{X}\hat{Y}$ multiplying the pixel values of the peak and the inverse of the camera intrinsics K. On the other hand, for each heatmap H^{uz} in the *uz* space, we compute the argmax of the z coordinate and convert it into a continuous value in metric space defined as $\hat{Z} = (z_{\max} \cdot \Delta Z) + \bar{Z}_{\min}$. Finally, we multiply the $\hat{X}\hat{Y}$ coordinates from the *uv* space and the $\hat{Z}$ coordinate from the *uz* space, obtaining a 3D point $\hat{P}$ for each robot joint.

IV. EXPERIMENTAL SETUP

A. SIMBA^{++} DATASET

To assess the validity of the proposed approach, we have expanded the existing SimBa dataset [14] by incorporating additional real sequences where the *Rethink Baxter* robot performs movements with both of its arms. This novel test dataset presents a great challenge, as the synthetic training dataset exclusively contains data for single-arm movements. In the subsequent sections, we provide comprehensive descriptions of the synthetic and real data, which are utilized for training and testing, respectively.

Synthetic data are collected in a virtual environment using Gazebo for physics simulation and Robot Operating System (ROS) for operating the synthetic robot model. The simulation consists of two runs with different random initializations containing 20 different camera poses from which the robot is recorded at 10 fps while performing 10 pick-n-place motions. The recordings are taken from three anchor cameras that are

TABLE 2: 2D Robot Pose Estimation results (see Sect. V-A) on SimBa^{++} synthetic and real sequences with single-arm movements

			Synthetic test set		Real test set			
			PCK (%) ↑		PCK (%) ↑			
Input	**Network**	**Params** (M)	**2.5px**	**Avg Error** (px) ↓	**2.5**px	**5**px	**10**px	**Avg Error** (px) ↓
RGB	FPM (MobileNet) [54]	0.16	88.23	1.71	15.97	**55.41**	**92.17**	**10.77**
	FPM (SqueezeNet) [54]	0.36	92.42	1.60	15.36	42.65	81.16	14.17
	SH (1 stack) [55]	14.8	99.44	0.67	0.46	10.66	17.56	68.95
	SH (2 stacks) [55]	26.8	99.41	0.66	0.13	5.59	10.21	95.65
	HRNet-32 [47]	28.5	99.58	0.65	**18.69**	53.51	71.69	22.48
	HRNet-48 [47]	63.6	99.62	0.62	2.67	8.99	17.99	70.87
	TransPose-R-A4 [56]	6.08	99.54	0.63	2.13	15.57	25.63	55.10
	Uniformer-B [57]	53.5	99.18	0.70	11.32	46.43	84.54	11.08
RGB-D	FPM (MobileNet) [54]	0.16	91.26	1.67	1.39	9.58	17.13	84.53
	FPM (SqueezeNet) [54]	0.36	92.17	1.63	**10.23**	**30.88**	**57.94**	**37.84**
	SH (1 stack) [55]	14.8	99.38	0.68	1.01	8.30	16.15	77.75
	SH (2 stacks) [55]	26.8	99.52	0.65	0.41	10.49	14.65	82.88
	HRNet-32 [47]	28.5	99.44	0.67	5.39	14.66	21.39	103.74
	HRNet-48 [47]	63.6	99.66	0.61	2.58	13.09	16.66	118.33
	TransPose-R-A4 [56]	6.08	99.59	0.65	2.21	13.59	25.15	58.92
	Uniformer-B [57]	53.5	99.29	0.68	5.54	22.22	36.66	58.29
DEPTH	FPM (MobileNet) [54]	0.16	88.43	1.75	33.83	72.32	95.51	6.28
	FPM (SqueezeNet) [54]	0.36	91.58	1.62	44.79	87.57	99.59	3.03
	SH (1 stack) [55]	14.8	99.41	0.68	43.85	87.94	92.28	7.35
	SH (2 stacks) [55]	26.8	99.62	0.64	47.99	93.73	98.44	4.02
	HRNet-32 [47]	28.5	99.51	0.67	48.35	88.57	93.31	6.84
	HRNet-48 [47]	63.6	99.65	0.61	50.16	95.37	99.12	2.85
	TransPose-R-A4 [56]	6.08	99.61	0.66	**57.41**	**96.48**	99.15	**2.66**
	Uniformer-B [57]	53.5	99.22	0.70	49.53	94.58	**99.73**	2.68
XYZ	FPM (MobileNet) [54]	0.16	88.37	1.71	37.03	70.29	93.61	6.73
	FPM (SqueezeNet) [54]	0.36	92.40	1.60	49.67	89.64	99.74	2.87
	SH (1 stack) [55]	14.8	99.55	0.66	39.67	91.31	97.15	5.09
	SH (2 stacks) [55]	26.8	99.50	0.69	43.32	90.68	96.29	4.69
	HRNet-32 [47]	28.5	99.54	0.67	50.29	96.96	**99.88**	2.66
	HRNet-48 [47]	63.6	99.61	0.66	49.42	95.23	99.08	2.83
	TransPose-R-A4 [56]	6.08	99.63	0.63	**51.89**	**97.42**	99.84	**2.59**
	Uniformer-B [57]	53.5	99.19	0.69	47.93	94.52	99.61	2.94

randomly positioned within a sphere of 1m diameter, as depicted in Figure 4. The movements cover most of the working space at the front of the robot, as illustrated in Figure 6 (top). This guarantees enough variation of the joints' positions for the training phase. The synthetic data contains a total of 400 sequences and 350k RGB-D frames with annotations for 16 joints, pick-n-place locations, and camera positions.

Real sequences are acquired through the *Microsoft Kinect One* ToF sensor, using ROS for recording the robot's movements. The camera is placed in three anchor positions (center, left, right), as depicted in Figure 4, so that they are within the space of the synthetic cameras, but not at the same exact location. As an extension to the original dataset [14], we introduce new sequences and divide the dataset into two groups: (i) *single-arm movements* and (ii) *double-arm movements*. The first test set remains the same as in the original dataset, containing 20 sequences at 15 fps from each camera position with pick-n-place motions with either the left or the right arm. The latter is the extension to the original dataset and consists of 10 additional sequences at 15 fps from each camera position with both robot arms moving. Sequences with both robot arms moving are not available in the synthetic dataset, thus incrementing the challenges in the domain shift operation. SimBa^{++} contains a total of 30k real RGB-D frames with annotations for 16 joints and 3 camera positions.

B. MODEL TRAINING

The model is trained for 30 epochs on the SimBa^{++} synthetic dataset using $L2$ loss on the heatmaps, batch size 16, *Adam* [58] optimizer, and learning rate $1e^{-3}$ with decay factor 10 at 50% and 75% of training. We follow the same training split of [14] to enable a direct result comparison.

We apply a 3D data augmentation on the point cloud computed from the depth map D_M. In particular, 3D points are rotated of $[-5°, +5°]$ on XY axes and translated of $[-8\text{cm}, +8\text{cm}]$ on XZ axes. We further translate the points on the XZ axes, changing implicitly the camera position. In addition to this geometric augmentation, we introduce a pixel-wise pepper noise and a random dropout of portions of the depth map, simulating respectively the noise of the real sensor and the holes caused by light on reflective surfaces (*e.g.* metallic objects or screens) that usually produce invalid depth

TABLE 3: 3D Robot Pose Estimation results (see Sect. V-B) on SimBa^{++} synthetic and real sequences with single-arm movements, exploiting 2D to 3D projection from depth data considering the surface-to-joint displacement

		Synthetic test set						Real test set					
		mAP (%) ↑						mAP (%) ↑					
	Network	2cm	4cm	6cm	8cm	10cm	ADD (cm) ↓	2cm	4cm	6cm	8cm	10cm	ADD (cm) ↓
DEPTH	FPM (MobileNet) [54]	21.36	62.35	75.99	78.16	80.65	10.29 ±6.18	7.30	24.02	55.07	74.28	81.93	13.49 ±10.93
	FPM (SqueezeNet) [54]	23.10	63.38	75.81	78.15	80.49	10.35 ±6.34	6.73	32.98	67.76	81.14	85.16	8.74 ±5.85
	SH (1 stack) [55]	32.73	68.35	75.17	77.71	80.01	10.45 ±6.29	8.78	36.44	68.66	77.31	79.80	11.37 ±7.72
	SH (2 stacks) [55]	33.48	68.57	75.59	78.18	80.60	9.46 ±5.41	9.62	40.69	72.35	82.42	84.69	8.17 ±5.27
	HRNet-32 [47]	33.57	68.56	75.64	78.02	80.38	9.83 ±5.68	9.01	37.31	68.41	78.33	80.02	11.88 ±8.85
	HRNet-48 [47]	33.34	68.79	75.64	78.23	80.50	9.46 ±5.44	9.81	39.36	70.74	83.08	85.99	7.19 ±4.32
	TransPose-R-A4 [56]	33.17	68.51	75.38	77.96	80.49	9.91 ±5.87	9.27	43.46	75.56	83.44	85.46	7.02 ±4.23
	Uniformer-B [57]	33.95	68.36	75.27	77.79	80.27	9.73 ±5.59	9.79	39.42	73.52	82.67	85.60	7.43 ±4.71
XYZ	FPM (MobileNet) [54]	21.70	62.67	75.49	77.83	80.28	10.86 ±6.57	4.83	24.21	49.39	68.64	78.99	17.32 ±14.15
	FPM (SqueezeNet) [54]	23.38	63.50	75.79	78.12	80.49	10.52 ±6.28	7.67	37.89	72.86	83.04	86.96	7.91 ±5.26
	SH (1 stack) [55]	33.05	68.48	75.43	77.97	80.30	9.67 ±5.55	8.91	39.34	71.61	80.81	84.42	8.99 ±5.63
	SH (2 stacks) [55]	34.05	68.61	75.61	78.13	80.37	9.69 ±5.58	6.79	40.04	72.02	80.42	83.06	8.70 ±5.40
	HRNet-32 [47]	33.02	68.24	75.55	77.97	80.37	9.53 ±5.38	8.71	39.55	72.55	83.17	86.98	7.03 ±4.50
	HRNet-48 [47]	33.43	68.78	75.60	78.13	80.38	9.47 ±5.39	9.27	40.84	73.52	82.59	85.13	7.03 ±4.20
	TransPose-R-A4 [56]	32.59	68.48	75.35	77.88	80.20	9.95 ±5.91	9.02	45.50	76.57	84.62	87.47	7.24 ±5.00
	Uniformer-B [57]	32.91	68.03	75.03	77.65	80.22	10.14 ±5.94	8.78	39.38	73.44	83.10	85.91	7.59 ±4.95

measurements. The pepper noise is introduced for $10-15\%$ of the pixels and the random dropout consists of rectangular areas of different dimensions where pixels are set to 0 value.

C. METRICS

For the quantitative evaluation of the proposed method and the competitors, we used 2D and 3D metrics already introduced in the literature for similar tasks.

For the 2D RPE, we use the Percentage of Correct Keypoints [59] (PCK) metric, *i.e.* the percentage of predicted joints that are within a certain distance threshold with respect to the ground truth. We compute PCK with a confidence threshold of 0.5 and a margin error of 2.5 pixels for the synthetic dataset and $\{2.5, 5, 10\}$ pixels for the real dataset. Moreover, we also compute the average pixel error over all robot joints.

For the 3D RPE, we use the average distance metric (ADD) [1], [60]: this metric measures the average distance of 3D model points between the ground truth pose and the predicted pose. In other words, it is the mean $L2$ distance expressed in centimeters of all 3D robot joints to their ground truth positions. This value (the lower the better) is useful to condense the error related to the translation and rotation in the 3D world. In addition, a *mean average precision* (the higher the better) is used as the accuracy on the ADD using different thresholds of $\{2, 4, 6, 8, 10\}$ centimeters. In this way, results can be evaluated at different distances from the ground truth, giving more interpretability to the actual performance of the methods.

V. EXPERIMENTAL RESULTS

Given the recent emergence of the 3D Robot Pose Estimation task from depth data, we take advantage of the opportunity to systematically examine the challenges within this research domain. This analysis is conducted in tandem with the evaluation of the proposed D-SPDH method.

We initiate our investigation in the 2D domain, with a particular focus on exploring the feasibility of employing approaches that have been introduced for the 2D HPE task. Furthermore, we evaluate the complexities inherent in the Sim2Real context, assessing the performance of our methods on both synthetic and real data. Subsequently, we transition towards the estimation of 3D pose. Our analysis begins with a simple approach involving the direct sampling of depth values from depth data and proceeds to more advanced techniques, including the regression of the complete 3D pose in world coordinates.

A. 2D ROBOT POSE ESTIMATION

In this task, we compare several literature approaches explicitly developed for the human pose estimation task, ranging from lightweight models [54] to recent Transformer-based architectures [56], [57]. These methods, originally based on the RGB domain, are tested on different input modalities, *i.e.* RGB, RGB-D (channel-wise stacked), depth, and XYZ (see Sect. III-A) images, belonging to both synthetic and real data.

Experimental results are shown in Table 2, in terms of PCK and average pixel error. As expected, good performances are visible on the synthetic data, while the difference between each input modality rises when testing on the more challenging real sequences. In particular, without applying any domain adaptation technique during training, depth and XYZ inputs overcome the RGB and RGB-D modalities, probably due to the fact that RGB data introduces a significant visual gap between the synthetic and real domains. On the other hand, since the depth and XYZ image representations contain fewer visual details (in particular no details about textures), the trained models tend to better generalize to the real domain, proving that the domain gap on these input types is reduced.

TABLE 4: Experimental comparison on 3D RPE (see Sect. V-C) between the proposed method (D-SPDH) and different baselines and competitors available in the literature. Results are obtained on SimBa^{++} real sequences with single-arm movements

		mAP (%) ↑					
Method	**Network**	**2**cm	**4**cm	**6**cm	**8**cm	**10**cm	**ADD** (cm) ↓
3D regression	ResNet-18 [61]	0.57	9.40	19.99	27.06	44.44	12.20 ±4.12
2D to 3D lifting	[62] *	**13.70**	26.96	37.98	48.40	58.33	10.03 ±3.53
Volumetric heatmaps	[63]	3.35	18.15	42.24	61.60	86.15	7.11 ±0.65
SPDH	TransPose-R-A4 [56]	2.58 ±0.77	43.45 ±3.00	73.56 ±1.95	89.15 ±1.24	93.99 ±0.52	5.89 ±1.69
SPDH	Uniformer-B [57]	11.58 ±0.98	40.48 ±1.80	68.90 ±1.97	85.01 ±1.11	91.43 ±0.45	5.61 ±1.79
SPDH	HRNet-32 [47]	7.31 ±2.48	48.61 ±5.33	79.88 ±8.76	91.65 ±7.97	96.79 ±3.55	4.65 ±1.00
D-SPDH	HRNet-32 [47]	10.62 ±5.69	**53.82** ±9.46	**85.43** ±4.36	**96.17** ±3.82	**98.88** ±1.36	**4.14** ±0.77

* relative joint positions

TABLE 5: Experimental comparison on 3D RPE (see Sect. V-C) between the proposed method (D-SPDH) and the competitor with three different backbones. Results are obtained on SimBa^{++} real sequences with double-arm movements

		mAP (%) ↑					
Method	**Network**	**2**cm	**4**cm	**6**cm	**8**cm	**10**cm	**ADD** (cm) ↓
SPDH	TransPose-R-A4 [56]	3.88 ±0.94	43.51 ±2.86	72.47 ±0.47	87.33 ±1.23	93.39 ±0.57	5.89 ±1.80
SPDH	Uniformer-B [57]	**11.58** ±1.03	43.29 ±1.86	71.85 ±1.55	88.00 ±0.84	93.21 ±0.56	5.36 ±1.97
SPDH	HRNet-32 [47]	6.71 ±1.38	49.54 ±6.84	80.39 ±8.16	91.75 ±7.84	96.73 ±3.73	4.65 ±0.96
D-SPDH	HRNet-32 [47]	10.26 ±4.97	**54.58** ±8.28	**85.04** ±4.76	**95.69** ±3.77	**98.86** ±1.41	**4.14** ±0.69

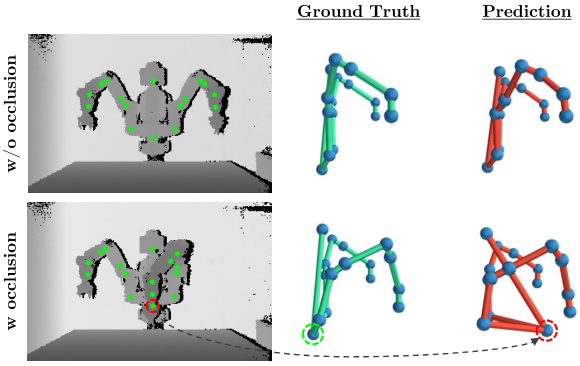


FIGURE 7: Example of the influence of self-occlusions on the predicted 3D pose using the 2D to 3D projection approach. With respect to a frame with all visible robot joints (first row), the occlusion caused by the left arm (second row) results in a large error for the robot base prediction.

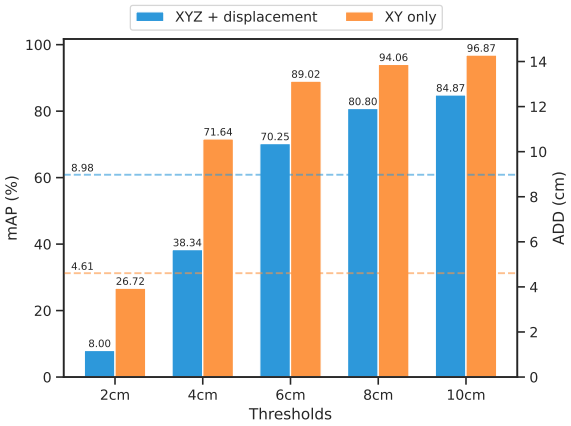


FIGURE 8: Evaluation comparison of the 2D to 3D projection from depth data (see Sect. V-B) in terms of mAP (barplot) and ADD (horizontal lines), considering XYZ with displacement or XY axes only. The trend is computed as an average over all the networks trained for the 2D pose estimation.

B. 2D TO 3D PROJECTION FROM DEPTH DATA

Once experimentally defined the depth-based inputs for the 2D estimation in the previous analysis, a simple approach to obtain a 3D joint prediction would be to take the 2D predicted coordinates and project them into the 3D space using the camera intrinsics and the corresponding depth value. However, we observe this projection would always lay on the surface of the robot, and therefore be incorrect, as the goal is to predict the central location of the robotic joint. Besides, the magnitude of the error would depend on the robot's model, shape, and pose (*e.g.* self-occlusions).

To mitigate this issue, we still sample the Z value from the depth map, but introducing also a fixed displacement (computed through the robot model), to reduce the distance between the prediction and the ground truth joint location. In other words, this displacement tries to move the sampled point from the surface to the proper position of the joint inside the robot. In this experiment, we compare the same networks trained on the 2D pose estimation in terms of mAP and ADD.

As shown in Table 3, the performance in the 3D domain looks satisfying, but especially at low mAP thresholds, the limitations of this approach arise. Indeed, since using the sampled Z coordinate from the depth produces ADD errors higher

than 8 centimeters, the mAP scores at low thresholds become unreliable. In addition, when testing on the real domain, the method is highly influenced by the quality and accuracy of the depth sensor since Z is sampled at a specific point. Another problem is the presence of self-occlusions, which leads to a sampled Z coordinate that is too distant from the inner joint of the robot (Fig. 7). Moreover, we report the results considering only the projected XY coordinates of the 3D space. As shown in Figure 8, it is worth noting that both mAP scores and ADD metric drop significantly when considering the Z values, proving that the sampling from the depth map is not reliable enough for precise 3D joint location.

C. 3D ROBOT POSE ESTIMATION

Given the shortcomings of the 2D to 3D projection analyzed in the previous section, we now consider the 3D robot pose estimation as a direct prediction from the input images. As shown in Table 4, we compare the proposed D-SPDH method with the 3D pose estimation literature.

1) Direct 3D regression

one of the most common approaches is to regress directly the 3D joint coordinates from an image using CNNs. We empirically select a ResNet-18 [61] backbone that is adapted and trained on the synthetic data to regress the 3D robot joint positions. However, as widely demonstrated for the human pose estimation case [36], this approach does not lead to good results, proving that estimating the 3D absolute pose of an articulated object with respect to the camera is not trivial.

2) 2D to 3D lifting

another widely used approach is predicting the 3D pose starting from a 2D pose. The main feature of this approach is the need for a relative joint position with respect to a specific root (*e.g.* the robot base), so the absolute 3D pose is computed with a post-processing fitting of the pose with respect to the camera position. For the comparison, we evaluate the method proposed by [62], in which a sequence of different Multi-Layer Perceptron (MLP) networks are trained to predict the 3D joints relying on their 2D positions. From the reported results, we observe that this method is prone to overfitting on the synthetic data obtaining low results on the prediction of the relative 3D pose.

3) Volumetric heatmaps

the third solution is based on volumetric heatmaps, a specific representation to encode 3D joint locations in a sampled 3D volume. We train the state-of-the-art method of [63] which outputs a volume of size $d \times w' \times h'$, with $d = 64$, $w' = \frac{w}{4}$, and $h' = \frac{h}{4}$. We observe that this method obtains good results but does not perform as well as all SPDH-based approaches. Moreover, the main problem with volumetric heatmaps is memory usage, especially if the goal is to obtain precise 3D joint locations. Indeed, the memory footprint increases exponentially with the size of the volumetric heatmap, limiting its resolution and leading to quantization errors. In our experiments, this approach leads to a heavy GPU memory requirement of $\approx$ 16GB, which is considerably higher than all other methods.

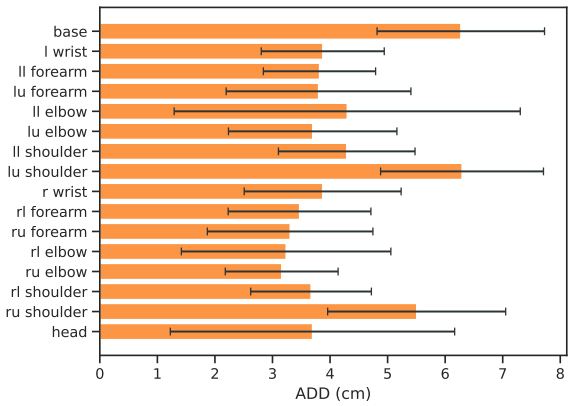


FIGURE 9: Results in terms of ADD metric (mean and std) for each robot joint on the real sequences with double-arm movements (l = left, r = right, ll = left-lower, lu = left-upper, rl = right-lower, ru = right-upper).

TABLE 6: Pose plausibility (see Sect. VI-A), *i.e.* the ability of the system to predict realistic joint locations, in terms of robot's limbs mean length error.

Method	Network	Limbs Error (cm)
3D regression	ResNet-18 [61]	1.22 ± 1.45
2D to 3D lifting	Martinez *et al.* [62]	0.67 ± 0.96
Volumetric heatmaps	Pavlakos *et al.* [63]	2.04 ± 1.94
SPDH	TransPose-R-A4	2.15 ± 5.44
SPDH	Uniformer-B	1.26 ± 1.90
SPDH	HRNet-32	1.00 ± 1.23
D-SPDH	HRNet-32	0.84 ± 0.73

4) SPDH vs D-SPDH representation

we take the top three baselines from the 2D pose estimation experiments, *i.e.* HRNet-32, TransPose, and Uniformer, and adapt them to predict the SPDH. Among the baselines, HRNet-32 is the best-performing one on the majority of mAP thresholds and on the ADD metric, so we use it as the backbone for D-SPDH. As stated by the results in Table 4, our approach outperforms SPDH by leveraging the double branch architecture and data augmentation. Moreover, as shown in Table 5, we report the results on the new test set with double-arm movements. This evaluation proves that our method obtains good results even though during training only single-arm movements are seen, outperforming the SPDH approach also in this scenario. Finally, as depicted in Figure 9, we analyze the performance of D-SPDH reporting the ADD metric for each robot joint. In this case, it is worth noting that the average error is similar for all the joints and the standard deviation (black line) is relatively low.

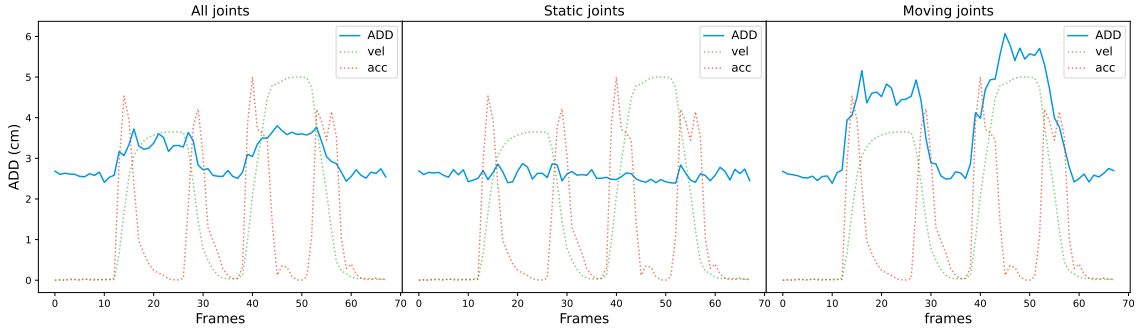


FIGURE 10: Temporal analysis of a real sequence with single-arm movement in terms of ADD and bone length (blue) with respect to velocity (green) and acceleration (red) of joints (see Sect. VI-A).

VI. DISCUSSION

A. MOVEMENT-ERROR CORRELATION AND POSE PLAUSIBILITY

To complete the experimental evaluation, we also explore the effect of the robot's movements on the accuracy of the final prediction. As depicted in Figure 10, we analyze the trend of the ADD metric with respect to the joints' movement in terms of acceleration and velocity. We split the graph into three sections considering the error for (i) all the joints, (ii) the static joints, and (iii) the moving joints. Indeed, the correlation between movement and error is present in most of the sequences suggesting that some actions generate a higher joint error. Moreover, the plots outline that the moving joints contribute the most to the error rate, so the static joints' location, *i.e.* the robot position, is preserved by the network over time. These elements suggest the possibility of including the temporal information in the pipeline to smooth the error caused by the movement of the robot arm.

As a second analysis of the results, we assess the problem of pose plausibility in terms of the robot's physical constraints. In particular, the goal is to prove that the length of the robot's limbs is preserved in the pose prediction, maintaining a realistic robot skeleton. We compute the limbs of the Rethink Baxter robot from its joints, obtaining a total of 15 limbs, where 4 are static. As shown in Table 6, D-SPDH obtains competitive results with a low average limb length error, demonstrating that the proportions of the robot are preserved while outperforming the competitors in the absolute 3D pose.

B. PERFORMANCE ANALYSIS

In the last part of our investigation, we analyze the impact of the proposed D-SPDH on the computational requirement. Specifically, we compare our system and the competitors in terms of execution time (expressed in milliseconds) against the ADD error, which well summarizes the performance of the system. For a fair comparison, all experiments are run on the same workstation with an Intel Core i7-7700K and an Nvidia GeForce GTX 1080 Ti, and performance are averaged over multiple input samples. Results of the performance analysis are graphically summarized in Figure 11. The two

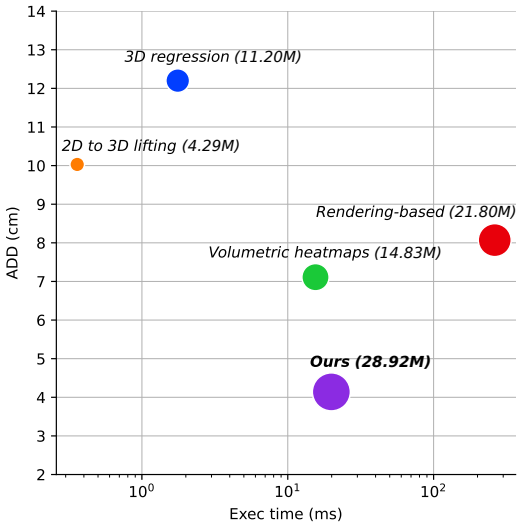


FIGURE 11: Performance comparison (see Sect. VI-B) of different approaches for 3D RPE in terms of execution time (expressed in milliseconds) and ADD metric (low is better). The circle size refers to the number of parameters which is specified next to each method.

main axes of the figure represent the ADD and the execution time, and the radius of the circles represents the number of parameters. Interestingly, our D-SPDH achieves the lowest error and a very competitive execution time, despite featuring the largest number of parameters. The execution time of D-SPDH enables real-time operation, *i.e.* the proposed system is able to achieve $\sim$ 50 frames per second. Unfortunately, solutions based on 2D to 3D lifting and 3D regression present faster execution time, but at the cost of reduced accuracy.

C. RESULTS DISCUSSION

Following our experimental evaluation, several observations emerge. Firstly, 2D Human Pose Estimation models prove effective not only in the conventional RGB input scenario but also when different modalities, such as depth maps, are employed. Then, we observe there is no necessity to devise specific backbones for the 2D RPE task, as demonstrated by

the successful adoption of HRNet-32, initially developed for human pose estimation.

Moving on to the 3D robot pose prediction, the 2D to 3D projection emerges as a straightforward technique, utilizing the Z value from depth maps. However, its limitation lies in predicting points only on the surface of objects, making it susceptible to challenges such as body occlusions.

Alternatively, employing a model directly regressing the 3D world coordinates of robot joints performs well on synthetic data but demonstrates a notable drop in performance due to domain shift when applied to real-world scenarios. This approach tends to overfit the training data, emphasizing the need for addressing domain shifts for improved performance. In this way, our proposed D-SPDH double-branch solution is a valuable solution marking a significant enhancement in the SPDH representation. Each branch specializes in extracting and predicting a specific heatmap, leading to improved accuracy and real-time performance. This advancement paves the way for potential applications in the development of collision-avoidance systems within industrial contexts.

Finally, the Sim2Real scenario simplifies acquiring new labeled data but poses challenges for the RPE task. Notably, there is a substantial performance gap between using synthetic and real-depth data as input. This aspect indicates an underexplored research field that needs further investigation: obtaining and annotating real-depth data, while not always practical, remains an effective strategy to enhance accuracy.

VII. CONCLUSION

In this paper, we have proposed the D-SPDH architecture to estimate the 3D pose of a robot, investigating the Sim2Real scenario, *i.e.* relying only on synthetic depth data as input during the training while testing the method on real sequences. We also have introduced SimBa^{++}, an extended version of the SimBa dataset including new challenging sequences with double-arm movements, on which the proposed system is tested and compared with literature baselines and competitors. The experimental evaluation confirms the suitability of the presented approach and the superior performance with respect to the literature, in terms of accuracy and real-time performance.

A variety of future work can be planned, ranging from the integration of temporal features in the framework to the use of domain adaptation techniques that reduce the semantic gap between synthetic and real scenes. Applying the proposed system to generic contexts (*e.g.* outdoor video surveillance or crowded scenarios) is non-trivial due to constraints from depth sensors. We observe a substantial lack of depth-based datasets containing extreme acquisition conditions (*e.g.* lighting, reflections) and multiple robots. In this context, the collection of a new dataset with these features could enable extensive evaluation procedures, aiming to measure the generalization capabilities of future models across different robot types and diverse environments. Finally, we highlight the need for a dataset representing realistic human-robot interaction in order to test the proposed method in a real scenario with both humans and robots.

ACKNOWLEDGMENT

Part of this work appears in the doctoral thesis of Alessandro Simoni, one of the authors of the article. The thesis titled "From Images to 3D Space: The Role of Semantic Keypoints for 3D Perception" is publicly available here: https://aimagelab.ing.unimore.it/imagelab/publications.asp

REFERENCES

[1] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield, "Camera-to-robot pose estimation from a single image," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 9426–9432.

[2] T. L. Munea, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang, and C. Yang, "The progress of human pose estimation: a survey and taxonomy of models applied in 2d human pose estimation," *IEEE Access*, vol. 8, pp. 133 330–133 348, 2020.

[3] X. Chen, C. Wei, Y. Yang, L. Luo, S. A. Biancardo, and X. Mei, "Personnel trajectory extraction from port-like videos under varied rainy interferences," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 7, pp. 6567–6579, 2024.

[4] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 348–353.

[5] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[6] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[7] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[8] A. Weiss, R. Buchner, M. Tscheligi, and H. Fischer, "Exploring human-robot cooperation possibilities for semiconductor manufacturing," in *2011 international conference on collaboration technologies and systems (CTS)*. IEEE, 2011, pp. 173–177.

[9] A. Weiss, A.-K. Wortmeier, and B. Kubicek, "Cobots in industry 4.0: A roadmap for future practice studies on human–robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 335–345, 2021.

[10] A. Kolbeinsson, E. Lagerstedt, and J. Lindblom, "Foundation for a classification of collaboration levels for human-robot cooperation in manufacturing," *Production & Manufacturing Research*, vol. 7, no. 1, pp. 448–471, 2019.

[11] E. Colgate, A. Bicchi, M. A. Peshkin, and J. E. Colgate, "Safety for physical human-robot interaction," in *Springer Handbook of Robotics*. Springer, 2008, pp. 1335–1348.

[12] K. Dautenhahn and J. Saunders, *New frontiers in human robot interaction*. John Benjamins Publishing, 2011, vol. 2.

[13] A. Pauliková, Z. Gyurák Babel'ová, and M. Ubárová, "Analysis of the impact of human–cobot collaborative manufacturing implementation on the occupational health and safety and the quality requirements," *Int. J. Environ. Res. Public Health*, vol. 18, no. 4, p. 1927, 2021.

[14] A. Simoni, S. Pini, G. Borghi, and R. Vezzani, "Semi-perspective decoupled heatmaps for 3d robot pose estimation from depth maps," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 569–11 576, 2022.

[15] Z. Zhang, L. Hu, X. Deng, and S. Xia, "Weakly supervised adversarial learning for 3d human pose estimation from point clouds," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 5, pp. 1851–1859, 2020.

[16] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, and J. Gall, "A survey on human motion analysis from depth data," in *Time-of-Flight and Depth Imaging. Sensors, Algorithms and Applications*. Springer, 2013, pp. 149–187.

[17] S. Höfer, K. Bekris, A. Handa, J. C. Gamboa, M. Mozifian, F. Golemo, C. Atkeson, D. Fox, K. Goldberg, J. Leonard *et al.*, "Sim2real in robotics and automation: Applications and challenges," *IEEE transactions on automation science and engineering*, vol. 18, no. 2, pp. 398–400, 2021.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[19] X. Chen, S. Dou, T. Song, H. Wu, Y. Sun, and J. Xian, "Spatial-temporal ship pollution distribution exploitation and harbor environmental impact analysis via large-scale ais data," *Journal of Marine Science and Engineering*, vol. 12, no. 6, 2024.

[20] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus time-of-flight kinect," *Computer Vision Image Understanding*, vol. 139, pp. 1–20, 2015.

[21] S. Pini, G. Borghi, R. Vezzani, D. Maltoni, and R. Cucchiara, "A systematic comparison of depth map representations for face recognition," *Sensors*, vol. 21, no. 3, p. 944, 2021.

[22] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 2018, pp. 969–977.

[23] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, pp. 2149–2154.

[24] R. Horaud and F. Dornaika, "Hand-eye calibration," *The international journal of robotics research*, vol. 14, no. 3, pp. 195–210, 1995.

[25] J. Heller, M. Havlena, A. Sugimoto, and T. Pajdla, "Structure-from-motion based hand-eye calibration using l∞ minimization," in *CVPR 2011*. IEEE, 2011, pp. 3497–3503.

[26] F. C. Park and B. J. Martin, "Robot sensor calibration: solving ax= xb on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[27] D. Yang and J. Illingworth, "Calibrating a robot camera." in *BMVC*, 1994, pp. 1–10.

[28] J. Ilonen and V. Kyrki, "Robust robot-camera calibration," in *2011 15th International Conference on Advanced Robotics (ICAR)*. IEEE, 2011, pp. 67–74.

[29] K. Pauwels and D. Kragic, "Integrated on-line robot-camera calibration and object pose estimation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2332–2339.

[30] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[31] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 590–596.

[32] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 3400–3407.

[33] Y. Zuo, W. Qiu, L. Xie, F. Zhong, Y. Wang, and A. L. Yuille, "Craves: Controlling robotic arm with a vision-based economic system," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4214–4223.

[34] A. Noguchi, U. Iqbal, J. Tremblay, T. Harada, and O. Gallo, "Watch it move: Unsupervised discovery of 3d joints for re-posing of articulated objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3677–3687.

[35] A. Sampieri, G. M. D. di Melendugno, A. Avogaro, F. Cunico, F. Setti, G. Skenderi, M. Cristani, and F. Galasso, "Pose forecasting in industrial human-robot collaboration," in *European Conference on Computer Vision*. Springer, 2022, pp. 51–69.

[36] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *arXiv preprint arXiv:2012.13392*, 2020.

[37] J. Bohg, J. Romero, A. Herzog, and S. Schaal, "Robot arm pose estimation through pixel-wise part classification," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3143–3150.

[38] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman *et al.*, "Efficient human pose estimation from single depth images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 35, no. 12, pp. 2821–2840, 2012.

[39] F. Widmaier, D. Kappler, S. Schaal, and J. Bohg, "Robot arm pose estimation by pixel-wise regression of joint angles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 616–623.

[40] J. Lambrecht, "Robust few-shot pose estimation of articulated robots using monocular cameras and deep-learning-based keypoint detection," in *2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA)*. IEEE, 2019, pp. 136–141.

[41] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: Efficient perspective-n-point camera pose estimation," *Int. J. Comput. Vis*, vol. 81, no. 2, pp. 155–166, 2009.

[42] S. Li, C. Xu, and M. Xie, "A robust o (n) solution to the perspective-n-point problem," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 34, no. 7, pp. 1444–1450, 2012.

[43] J. Tremblay, S. Tyree, T. Mosier, and S. Birchfield, "Indirect object-to-robot pose estimation from an external monocular rgb camera," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2020, pp. 4227–4234.

[44] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Single-view robot pose and joint angle estimation via render & compare," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 1654–1663.

[45] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[46] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 171–27 183, 2021.

[47] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 5693–5703.

[48] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2017, pp. 23–30.

[49] N. Morrical, J. Tremblay, Y. Lin, S. Tyree, S. Birchfield, V. Pascucci, and I. Wald, "Nvisii: A scriptable tool for photorealistic image generation," *arXiv preprint arXiv:2105.13962*, 2021.

[50] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2019.

[51] E. Frigieri, G. Borghi, R. Vezzani, and R. Cucchiara, "Fast and accurate facial landmark localization in depth images for in-car applications," in *International Conference on Image Analysis and Processing*. Springer, 2017, pp. 539–549.

[52] P. Zanuttigh, G. Marin, C. Dal Mutto, F. Dominio, L. Minto, and G. M. Cortelazzo, "Time-of-flight and structured light depth cameras," *Technology and Applications*, pp. 978–3, 2016.

[53] M. Ahad, A. Rahman, J. Tan, H. Kim, and S. Ishikawa, "Motion history image: its variants and applications," *Machine Vision and Applications*, vol. 23, no. 2, pp. 255–281, 2012.

[54] A. Martínez-González, M. Villamizar, O. Canévet, and J.-M. Odobez, "Efficient convolutional neural networks for depth-based multi-person pose estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4207–4221, 2019.

[55] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Europ. Conf. Comput. Vis.*, 2016, pp. 483–499.

[56] S. Yang, Z. Quan, M. Nie, and W. Yang, "Transpose: Keypoint localization via transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 802–11 812.

[57] K. Li, Y. Wang, G. Peng, G. Song, Y. Liu, H. Li, and Y. Qiao, "Uniformer: Unified transformer for efficient spatial-temporal representation learning," in *International Conference on Learning Representations*, 2022.

[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[59] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.

[60] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *Robotics: Science and Systems*, 2018.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[62] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[63] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 7025–7034.

ALESSANDRO SIMONI He currently holds the position of Deep Learning and Computer Vision Engineer at Covision Lab. In 2024, he obtained a PhD from the AImageLab at the University of Modena and Reggio Emilia. His research activities have focused on 3D reconstruction of objects and 3D pose estimation of humans and robots. Additionally, he has been involved in two corporate projects, one addressing 3D Human-Centric Scene Understanding in urban surveillance scenarios and the other focusing on 3D vehicle reconstruction in real-world environments.

GUIDO BORGHI is an Associate Professor within the Department of Education and Humanities, University of Modena and Reggio Emilia. He received the M.Sc. degree in Computer Engineering and the Ph.D. in Information and Communication Technologies from the University of Modena and Reggio Emilia, Italy, in 2015 and 2019, respectively. His research interests include Computer Vision and Deep Learning techniques applied to intensity and depth images for Face Analysis, Biometrics, Driver Monitoring and Human Computer Interaction.

LORENZO GARATTONI received the bachelor's and master's degrees from the University of Bologna, Italy. In 2012, he started working toward the PhD degree with the Universite libre de Bruxelles, focusing on cognition in multi-robot systems, under the supervision of Prof. M. Birattari. Since 2018, he has been working with Toyota Motor Europe, where he is a senior engineer developing perception technologies and control software for robots.

GIANPIERO FRANCESCA received the bachelor's and master's degrees from the "Sannio" University in Benevento, Italy. In 2011, he started working toward the PhD degree with the ULB University in Brussels, focusing on swarm robotics and automatic design of control software, under the supervision of Prof. M. Birattari. Since 2015, he has been working with Toyota Motor Europe. Today he is a senior engineer there, developing human activity recognition technology

ROBERTO VEZZANI graduated in Computer Engineering in 2002 and received his PhD course in Information Engineering in 2007 at the University of Modena and Reggio Emilia, Italy. Since 2016 is Associate Professor at the Dipartimento di Ingegneria "Enzo Ferrari" of the University of Modena and Reggio Emilia. His research interests mainly belong to video surveillance systems, with a particular focus on head, hand, and body pose estimation, vision-based HCI, motion detection, people tracking, and re-identification. Recently, he has also been involved in research projects on vision for automotive. He is the author of about 80 papers published in international journals and conference proceedings. He is member of ACM, IEEE and CVPL (Italian Chapter of IAPR).