RefiNet: 3D Human Pose Refinement with Depth Maps

Andrea D'Eusanio[1], Stefano Pini[1], Guido Borghi[2], Roberto Vezzani[1,2], Rita Cucchiara[1,2]
[1]DIEF - Dipartimento di Ingegneria "Enzo Ferrari"
[2]AIRI - Artificial Intelligence Research and Innovation Center
University of Modena and Reggio Emilia, Italy
Email: {andrea.deusanio, s.pini, guido.borghi, roberto.vezzani, rita.cucchiara}@unimore.it

***Abstract*—Human Pose Estimation is a fundamental task for many applications in the Computer Vision community and it has been widely investigated in the 2D domain, *i.e.* intensity images. Therefore, most of the available methods for this task are mainly based on 2D Convolutional Neural Networks and huge manually-annotated RGB datasets, achieving stunning results. In this paper, we propose *RefiNet*, a multi-stage framework that regresses an extremely-precise 3D human pose estimation from a given 2D pose and a depth map. The framework consists of three different modules, each one specialized in a particular refinement and data representation, *i.e.* depth patches, 3D skeleton and point clouds. Moreover, we present a new dataset, called *Baracca*, acquired with RGB, depth and thermal cameras and specifically created for the automotive context. Experimental results confirm the quality of the refinement procedure that largely improves the human pose estimations of off-the-shelf 2D methods.**

I. Introduction

The *Human Pose Estimation* (HPE), *i.e.* the localization of significant joints of the human body, on an image is a crucial and enabling task in many vision-based applications, like Action Recognition [1], [2] and People Tracking [3]. Recently, many methods based on deep learning architectures [4], [5], [6] have in turn improved the accuracy in joint detection and localization on intensity images, achieving stunning results. Encouraged by the seminal work of Shotton *et al.* [7] developed for depth images, the research on marker-less human pose estimation is now more focused on RGB images. The combination of effective deep learning approaches (*e.g.* Convolutional Neural Networks) and huge datasets of RGB images (*e.g.* COCO [8]) have led to impressive performance, in terms of accuracy, computational load, and generalization capabilities. Nowadays, it is possible to obtain a reliable localization of body joints even in presence of challenging situations such as occlusions, cluttered backgrounds, low-quality images, and so on. The pose is provided in 2D image coordinates, thus without the third coordinate – referred as the depth or z-value – and lacking any metric information.

In this work, we are focusing on applications that require a extremely-precise estimation of the 3D position of each joint. Taking into account, for instance, the automotive field [9], [10], the configuration of some car parameters could be set depending on anthropometric measures of the driver and the passenger. An automatic system based on computer vision could be an excellent solution in this regard.

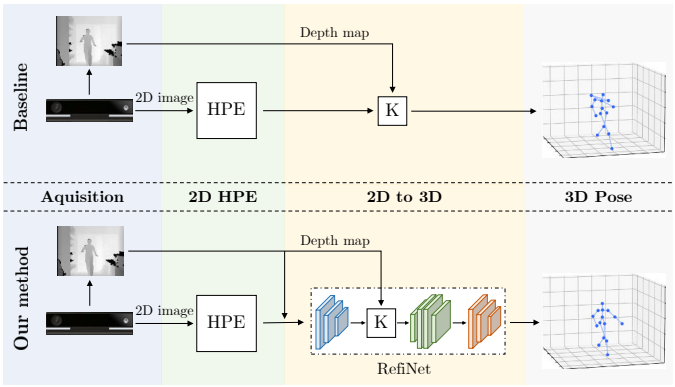


Fig. 1: A baseline method for the 3D human pose estimation from depth maps compared to the proposed one. K is the mapping operation between 2D and 3D coordinates, requiring camera calibration parameters and depth values. Further details are in Section III.

Some preliminary works [11], [12] have proposed methods to recover a complete 3D pose from RGB images with promising results. However, even if these methods predict a good estimation of the pose, they fail to recover the correct positioning in the camera space as well as the real scale of the body [12]. Thus, the errors will affect the computation of the corresponding measures of body parts and limbs (*e.g.* the exact height of person or the length of arms and legs). In these cases, depth sensors are a valid solution in place of traditional cameras. Indeed, depth cameras are more and more widespread, miniaturized, and cheap; they have been recently integrated in some embedded and mobile devices; and, in particular, they capture 3D information of the scene.

In this paper, we aim to combine the aforementioned successful deep learning architectures for 2D human pose estimation with the 3D measurement capabilities of depth sensors. Specifically, we focus on depth data collected through active devices, *i.e.* sensors coupled with an infrared light emitter. These sensors are safe for humans, invariant to environmental light conditions, and they can operate even without external light sources.

Aware that existing 2D pose estimation methods [4], [5], [6] achieve remarkable accuracy and real-time performance, and considering the lack of pose estimation systems relying

on depth data, we propose *RefiNet*, a framework designed to recover an accurate 3D human pose estimation from a given 2D one. In particular, *RefiNet* is a multi-stage system that regresses a precise 3D human pose through a sequential refinement of an approximate 2D estimation on a depth map. It consists in three different modules, each one specialized in a particular type of refinement and data representation. Thanks to its modular structure, each module can be activated or deactivated according to the needs. The initial 2D estimation can be obtained exploiting any 2D human pose estimation system available in the literature thanks to the adopted training procedure. The predicted 3D pose is expressed in the 3D camera-space coordinate system.

Summarizing, our contributions are the following:

- We propose *RefiNet*, a multi-stage framework for 3D human pose refinement. To the best of our knowledge, this paper is one of the first attempts to investigate the human pose refinement in combination with depth data;
- The framework, built as a set of three independent modules, exploits different data representations, ranging from 2D depth patches to 3D point clouds and is independent of the off-the-shelf methods used for the initial 2D human pose estimation.
- We present *Baracca*, a novel dataset acquired with a set of RGB, depth, and thermal cameras. It contains nearly 10k frames of 30 different subjects from 8 different points of view. The dataset is designed for the automotive context and contains sequences in which the subject is inside a car and others that simulate the outside car view. This dataset is used in conjunction with the proposed method for the anthropometric measure estimation task.

Dataset and source code are publicly available at https://aimagelab.ing.unimore.it/go/3d-human-pose-refinement.

II. Related Work

In this Section, we analyze the *Human Pose Estimation* on intensity (RGB or grey-level) and depth images and the *Human Pose Refinement* task.

Human Pose on intensity images. Intensity images represent the input of the large majority of methods available in the literature. Recently, most state-of-art 2D pose estimators exploit CNNs [13], [14], [15], [4], [16], [5]: we briefly analyze here works relevant for this work. The well-known work introduced by Cao *et al*. [4] proposed the use of *Part Affinity Fields* to learn the links between body parts. This work represents an evolution of the sequential architecture described in [13]. Recently, [5] introduced a model that preserve high-resolution representations through the whole pose estimation pipeline, repeating multi-scale fusions inside the deep model and achieving state-of-art results.

Since all these methods achieve a good accuracy in the 2D domain, we believe they can be successfully exploited also in other domains, *e.g*. the depth domain.

Human Pose on depth images. Only a limited number of works tackles the problem of human pose detection on depth maps, probably due to the limited number of datasets containing real or synthetic labelled depth data. Indeed, most of depth-based datasets are relatively small, *i.e*. not oriented to deep learning-based approaches, and automatically annotated, *i.e*. the annotations about the position of the body joints are extracted through [7], resulting in unreliable and imprecise annotations.

The work of Shotton *et al*. [7] represents a milestone in the human pose estimation task on depth maps: it is based on *Random Forest* trained on a not-released synthetic dataset. In addition to a reasonable accuracy and real-time performance, its widespread has been guaranteed by its implementation on the *Microsoft Kinect SDK*. Then, this method has been largely used both in gaming and research activities.

The method described in [17] proposes a method, based on *Hough forests*, that directly regresses body joint coordinates from depth maps, without the use of any intermediate representation. The system is able to localize visible as well as occluded body joints. In [18], random trees are employed to the body joint localization from a single depth image. Then, joints are classified using a nearest neighbor approach.

In [19], authors present the ITOP dataset, which contains about of 50k low-quality depth images from both top and side views with manually-annotated body joints. In the same work, they propose a model able to embed local regions into a view-invariant feature space for the human pose estimation. Recently, a new dataset, called *Watch-R(efined)-Patch* (WrP), has been proposed in [20], along with a fully convolutional [21] and multi-stage network architecture to perform pose estimation on depth maps. Starting from the automatic human pose annotations of *Watch-n-Patch* (WnP) [22], authors manually correct about 3k joint locations on depth maps.

Other works, related to scanner-based 3D models, estimate the human pose finding the correspondences between an acquired point cloud and a pre-defined 3D model [23], [24] or through a Gaussian mixture model [25]. These methods, based on 3D models instead of depth images acquired with active depth sensors, are out of the scope of this paper.

Human Pose Refinement. Existing methods of Human Pose Refinement are based on the 2D information of the intensity images. Generally, these methods [26], [14], [27] exploit a multi-stage architecture, trained end-to-end, in order to iteratively refine the pose estimation of previous stages or models. Others [28] exploit a shared weight model to estimate the error on the pose prediction. As reported in [29], all of these methods merge in a single model the pose estimation and the refinement task, obtaining a refinement module that is strictly dependent on the estimation approach. Moon *et al*. [29] proposed a solution called PoseFix, a model-agnostic human pose refinement network which is trained with synthetic poses generated exploiting the error statistics presented in [30]. A similar approach has been introduced in [31], where a simple post-processing network is trained through synthetic poses generated starting from ad-hoc rules.

The approach proposed in [32], based on RGB and segmentation images, focuses on body part to refine a 3D pose.

III. Proposed Method

RefiNet is a multi-stage framework that estimates an accurate 3D human pose in the real world using a depth image, starting from a set of 2D image-coordinates of the body joints, resulting from an approximate initial estimation. An overview of the proposed framework, compared to a baseline method, is shown in Figure 1. For the sake of clarity and ease of comprehension, the reported schema includes the generation of an initial 2D estimation ("HPE"), discussed in Section III-A.

Regardless of the method used for the initial estimation, *RefiNet* refines the predicted joints and outputs an accurate 3D human pose, expressed as a set of 3D joints in the camera space, *i.e.* in the absolute 3D camera coordinate system.

The *RefiNet* framework is developed in a modular way and it is divided into three different modules (Module A, B and C), detailed in the following subsections. Each module is individually trained and is independent from the others. During the testing phase, each module refines the noisy pose given as input. The overall pipeline of *RefiNet* is depicted in Figure 2.

A. Initial 2D pose estimation

RefiNet requires an initial 2D human pose estimation on a depth image. In this section, we present some approaches that can be used to obtain it and a baseline approach to recover the 3D body pose from the 2D estimation and the depth image.

The 2D coordinates of the body joints can be obtained using any off-the-shelf pose estimator applied on a 2D image, *i.e.* the RGB image, the IR amplitude channel, or the depth image (depending on the sensor/dataset used). Supposing the use of well-known human pose estimators trained on RGB datasets (such as COCO [8] and MPII Human Pose [33]), the RGB channel would ensure the best results, but not all the sensors and datasets provide it along with the depth channel. Moreover, coordinate translation and parallax issues between the RGB channel and the depth one should be taken into account. On the other end, the depth and the IR amplitude channel are aligned by definition, but the pose estimation methods may perform worse or even not work on these kinds of data. For instance, we had to retrain them from scratch to work on depth data. However, the lack of depth-based datasets with accurate manual joint annotations negatively affects the performance of these depth-based models.

Once the 2D estimation is computed and mapped in the depth-map space, each 2D joint coordinate can be translated into the camera-space 3D coordinates using camera calibration parameters and the value sampled from the depth image. The 3D pose estimation obtained with this approach is always an approximation. Even in case of correct 2D pose estimation, the resulting 3D joints would lie on the body surface and may be affected by errors due to occlusions and noise. An overview of this approach is depicted in Figure 1 (top). To overcome these limitations, we propose the use of the *RefiNet* framework, as in Figure 1 (bottom).

B. Module A: 2D patch-based refinement

The first module of the framework refines the 2D human pose exploiting visual cues computed on depth maps.
The input of the module are a set of body joints, expressed in (x, y) coordinates and the corresponding depth image. A depth-map patch is cropped around each body joint and used as input of the deep network described below, which outputs an offset w.r.t. the input coordinates. The offset is represented as a displacement vector $(\overrightarrow{x}, \overrightarrow{y})$ which denotes the displacement of each joint w.r.t. its initial position. Indeed, considering the input coordinates (x, y) and the predicted vector $(\overrightarrow{x}, \overrightarrow{y})$, the refined coordinates of each joint on the depth image can be further computed. In this way, Module A is able to correct small errors in the 2D joint predictions. It is worth note that a small error in terms of 2D coordinates on the depth image could highly influence the sampled z-value, resulting in an inaccurate 3D skeleton (see Figure 4a).
Model. The deep network of Module A is based on 3 different blocks. The first block takes the depth-image patches as input and extracts features through a single 7×7 convolutional layer with 64 feature maps. Then, the spatial dimension is reduced by applying a max pooling layer with stride $s = 2$. The second block id composed of 2 residual layers [34] with 64 and 128 feature maps and stride $s = 2$. An average pooling layer is then used to aggregate the feature maps. Finally, a series of fully connected layers with 256, 256, 2 hidden units regresses the $2D$ joint displacement from the averaged deep features.
Loss Function. The adopted loss function L_A is the mean squared error between the predicted and the ground truth offset for each body joint. A mask is applied in order to ignore non-visible joints in the loss computation:

$$L_A = \frac{1}{N} \sum_{i=1}^{n} W_i \cdot \left\| \overrightarrow{v_i} - \overrightarrow{t_i} \right\|_2^2 \tag{1}$$

where n is the number of joints of the skeleton and, for each joint i, $\overrightarrow{v_i} = (\overrightarrow{v_i^x}, \overrightarrow{v_i^y})$ is the predicted displacement, $\overrightarrow{t_i} = (\overrightarrow{t_i^x}, \overrightarrow{t_i^y})$ is the ground truth, and W_i is the binary mask. $W_i = 0$ iff the joint annotation is missing or the joint is not visible.
Training. Given a input joint in (x, y) coordinates, a 40×40 squared bounding box (patch) centered in (x, y) is extracted from the depth map. If needed, patches are padded accordingly to the joint location and the width and height of the depth image. Each patch is then normalized to obtain a zero-mean unit-variance tensor that is fed to the network. Network weights are updated using the Adam optimizer [35] and a learning rate of 0.001, in combination with batch normalization and dropout. The training phase is performed using ground truth data only, *i.e.* the network is not trained on the output of any specific 2D human pose estimator, aiming to obtain a generic refinement module. Specifically, the network input is artificially created applying Gaussian noise (with $\mu = 0, \sigma = 5$) on the ground-truth joint coordinates provided as annotations in the training dataset. This procedure aims to simulate the error distribution over the skeleton joints of a generic 2D Human Pose Estimation method.

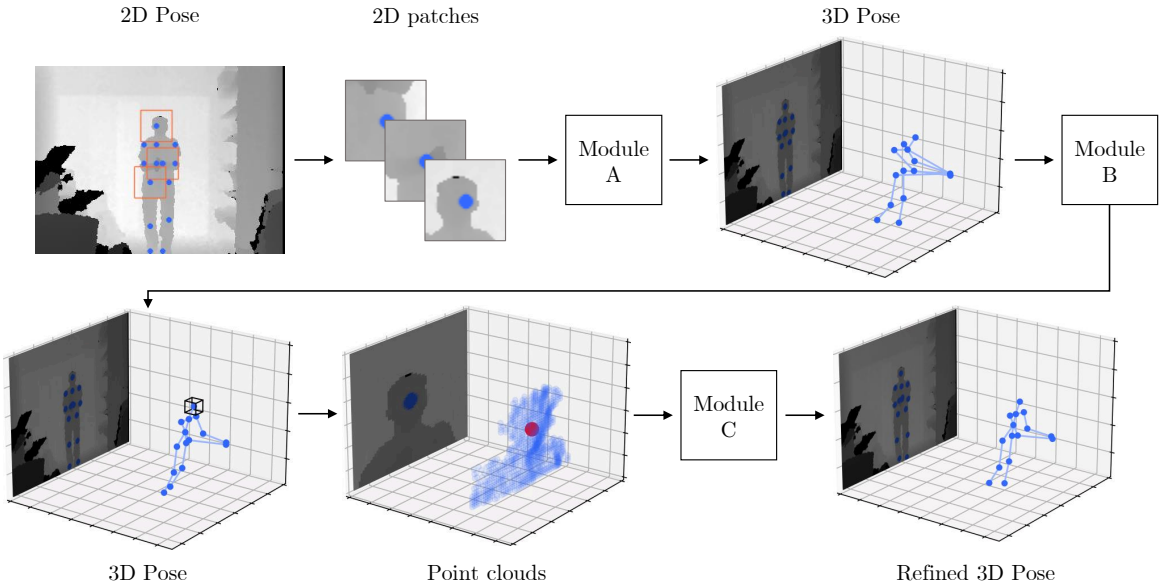


Fig. 2: Overview of the proposed framework referred as *RefiNet*. Module A analyzes 2D depth patches, extracted from depth maps. Module B works directly on the 3D skeleton while module C processes point clouds computed around individual joints.

C. Module B: skeleton-based refinement

The second module of the framework converts the 2D joint coordinates into the 3D camera space (*i.e.* the 3D real-world camera coordinates) and refines the 3D human pose using only the 3D skeleton. It takes the 2D (x, y) joint coordinates predicted in the depth map coordinate system as input and computes the 3D real-world coordinates x_C, y_C, z_C using the camera calibration parameters $K = \{f_x, f_y, c_x, c_y\}$:

$$\begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} = \begin{bmatrix} (x - c_x) \cdot \frac{z}{f_x} \\ (y - c_y) \cdot \frac{z}{f_y} \\ z \end{bmatrix} \tag{2}$$

where z is the value of the depth map sampled in (x, y), f_x and f_y are the focal lengths, c_x and c_y the coordinates of the optical center. To mitigate the effect of noise and missing depth data, the sampling of z is performed by calculating the median value within a 3×3 neighborhood of (x, y). Then, the 3D human skeleton – expressed as the set of body joints in camera space – is fed to the deep model described below, that predicts the refined 3D skeleton joints in real-world coordinates. Compared to the previous one, Module B converts the coordinates from the 2D depth-map space to the 3D camera space and directly regresses the refined position of every joint of the human skeleton.

Model. The network, inspired by the successful work of Martinez *et al*. [36], is composed of a sequence of 4 blocks: one input block, consisting in a fully-connected layer with 1024 units; two residual blocks, each containing 2 fully-connected layers with 1024 units; one output block, corresponding to a fully-connected layer with $n \cdot 3$ units where n is the number of joints of the skeleton. Each fully-connected layer consists of a linear layer, a batch normalization layer, and a ReLU activation.

Loss Function. The adopted loss function is the mean squared error between the predicted and the ground truth position of each skeleton joint in the camera coordinate system:

$$L_B = \frac{1}{N} \sum_{i=1}^{n} \|\mathbf{s}_i - \mathbf{t}_i\|_2^2 \tag{3}$$

where n is the number of joints and, for each skeleton joint i, $\mathbf{s}_i$ is the predicted joint position $(x_{C_i}, y_{C_i}, z_{C_i})$, and $\mathbf{t}_i$ is the ground truth joint position.

Training As in Module A, during the training procedure we apply Gaussian noise ($\mu = 0, \sigma = 5$) on the ground-truth annotations taken from the training dataset and use these noisy joints as input of the module. The noise is applied on the (x, y) coordinates, before retrieving the z-value and converting them into 3D camera coordinates, in order to simulate the error of a 2D human pose estimator. Adam [35] is adopted as optimizer and the learning rate is set to 0.001.

D. Module C: point cloud-based refinement

The third module of the framework firstly converts the depth map into a point cloud (using the camera calibration parameters K). Then, it refines the body joints exploiting the 3D information of the point cloud by sampling the points in the neighborhood of each joint location. To this aim, we exploit the *PointNet* architecture [37], specifically developed to handle point clouds. We extract and analyze small point clouds sampling a squared 3D space around each skeleton joint instead of considering the whole depth map – ranging from the head to the feet of the subject – to compute a single, huge point cloud.

Similarly to Module A, Module C is based on a deep model

that learns how to correct the location of each body joint by predicting independent offsets. Each regressed offset is expressed as the displacement vector $(\overrightarrow{x_C}, \overrightarrow{y_C}, \overrightarrow{z_C})$ between the input locations of the (x_C, y_C, z_C) joint coordinates in the camera space and the refined ones.

Model. Taking inspiration from [37], we propose an architecture based on 2 parts: one block for feature extraction and one for offset regression. The first block computes single-point features with a series of fully-connected layers. Then, single-point features are aggregated through a max-pooling layer. For further details, see [37]. The second block computes the joint offset from the point-cloud features through a fully-connected layer with 128 units and ReLU activation and an output layer with 3 units (corresponding to the 3D displacement vector).

Loss Function. The adopted loss function is the mean squared error between the predicted and the ground truth 3D offset for each skeleton joint:

$$L_C = \frac{1}{N} \sum_{i=1}^{n} \left\| \overrightarrow{v_i} - \overrightarrow{t_i} \right\|_2^2 \tag{4}$$

where n is the number of joints and, for each joint i, $\overrightarrow{v_i} = (\overrightarrow{x_C^{v_i}}, \overrightarrow{y_C^{v_i}}, \overrightarrow{z_C^{v_i}})$ is the predicted displacement while $\overrightarrow{t_i} = (\overrightarrow{x_C^{t_i}}, \overrightarrow{y_C^{t_i}}, \overrightarrow{z_C^{t_i}})$ is the ground truth value.

Training. As in the previous modules, Gaussian noise ($\mu = 0, \sigma = 42$) is added on the ground-truth annotations of the training set to create the input data. In this case, since the module works in the 3D camera space, the noise is added to the (x_C, y_C, z_C) coordinates of each joint before the crop of the point cloud. We adopt the Adam [35] optimizer, the learning rate is set to 0.001 and both batch normalization and dropout, with drop probability 0.2, are employed.

IV. Experimental Evaluation

In this section, we report the dataset used in the experiments and present the novel *Baracca* dataset. Then, we detail the conducted experiments and the obtained results. Finally, a real-world application is proposed on the *Baracca* dataset.

A. Datasets

The main drawback of using depth maps for the Human Pose Estimation task is the lack of manually-annotated datasets. Many datasets include annotations on the body surface (*e.g.* [20]) while datasets obtained using, for instance, the *Mocap* system are not always reliable because the depth value of a body joint usually correspond to the marker placed on the body surface and the visual appearance and 3D shape are altered by the markers.

ITOP. The dataset, introduced in [19], consists of about 40k training and 10k testing depth maps of 20 subjects performing 15 different actions. Depth images were recorded using two *Asus Xtion Pro*, a Structured Light depth sensor having a resolution of 320 × 240 pixels. One sensor is placed above ("top-view") and the other one in front of ("side-view") the acquired subject. Annotations consist in the 2D and 3D coordinates of 15 body joints. Exploiting the two points of view, the body joints are semi-automatically annotated and manually refined to lie inside the body of the subject, *i.e.* at the 3D center of the physical joint.

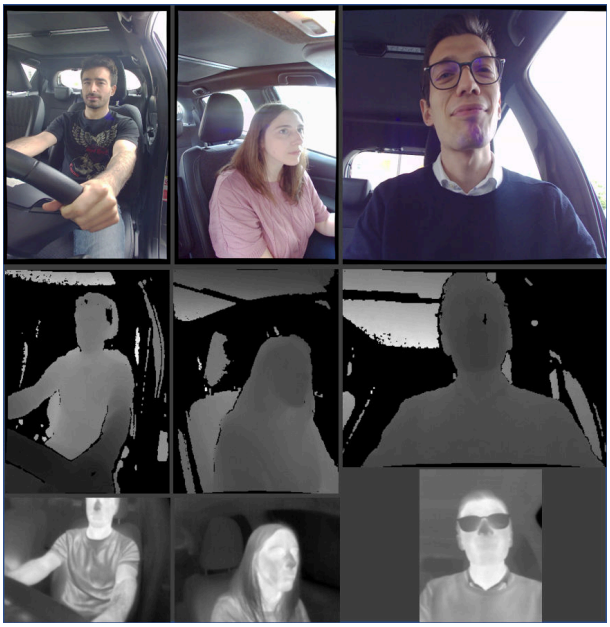


Fig. 3: Samples from the Baracca dataset, inside car sequences. The dataset contains RGB, IR, depth, and thermal images.

Baracca. To further evaluate the applicability of the proposed approach, we collected a new dataset, called *Baracca* and acquired with RGB, depth, and thermal cameras. It contains nearly 10k frames of 30 different subjects in 8 different positions. For each subject, 10 pictures are simultaneously acquired by each camera from 8 different points of view (5 outside the car, 3 inside the car). The same set of cameras is used for each point of view.

The dataset is designed for the automotive context and contains sequences in which the subject is inside a car and others that simulate the outside car view. Two cameras were exploited for the acquisition. The first one is the *Pico Zense DCAM710*[1], a depth sensor based on the Time-of-Flight technology. It is able to acquire infrared and depth images of 640 × 480 pixels at 30 fps, covering the range 0.2 - 5 meters. It is coupled with an RGB sensor (1920 × 1080 pixels). The second one is the *PureThermal 2* board[2] equipped with a *FLIR Lepton 3.5*[3], a low-resolution (160 × 120 pixels) radiometric thermal sensor. A set of anthropometric and biometric measurements is also provided for each subject, to allow the challenging task of their estimation from the acquired images. The available measurements are the following: *Age, Weight, Height, Shoulder width, Forearm length, Arm length, Torso width, Leg length, Eye height from the ground*. The dataset does not contain ground truth keypoint annotations.

[1] https://www.picozense.com/en/spec.html?spec=710

[2] https://groupgets.com/manufacturers/getlab/products/purethermal-2-flir-lepton-smart-i-o-module

[3] https://groupgets.com/manufacturers/flir/products/lepton-3-5

TABLE I: mAP (Eq. 5, percentage) and mDE (Eq. 6, centimeters), for both the side and the top view of the ITOP dataset. Mod. A, Mod. B, and Mod. C refer to the three modules of RefiNet. The ✓ symbol indicates that the module is used for the refinement. In the first row we report the results of the 2D predictors; improvements are calculated w.r.t. these values.

Side view

Mod. A	Mod. B	Mod. C	OpenPose [4]				HRNet [5]			
			mAP ↑	Improv.	mDE ↓	Improv.	mAP ↑	Improv.	mDE ↓	
			0.646	-	12.634	-	0.670	-	10.711	-
✓			0.687	6.35%	10.442	17.4%	0.699	4.32%	10.060	6.08%
	✓		0.775	20.0%	8.463	33.0%	0.787	17.5%	8.185	23.6%
		✓	0.719	11.3%	11.834	6.33%	0.734	9.55%	10.693	0.17%
✓	✓		0.796	23.2%	8.042	36.3%	0.804	20.0%	7.790	27.3%
✓	✓	✓	**0.818**	**26.6%**	**7.646**	**39.5%**	**0.824**	**23.0%**	**7.447**	**30.5%**

Top view

Mod. A	Mod. B	Mod. C	OpenPose [4]				HRNet [5]			
			mAP ↑	Improv.	mDE ↓	Improv.	mAP ↑	Improv.	mDE ↓	
			0.153	-	70.672	-	0.175	-	68.755	-
✓			0.164	7.19%	69.137	2.17%	0.173	-1.14%	68.580	0.25%
	✓		0.665	334.6%	10.464	85.2%	0.713	307.4%	9.836	85.7%
		✓	0.205	34.0%	68.218	3.47%	0.215	22.9%	66.444	3.36%
✓	✓		**0.675**	**341.2%**	**10.349**	**85.4%**	**0.718**	**310.3%**	**9.550**	**86.1%**
✓	✓	✓	0.619	304.6%	10.973	84.5%	0.663	278.9%	10.160	85.2%

B. Experiments

The proposed method, called *RefiNet*, performs a refinement of 2D body joints on a depth map in order to obtain accurate 3D pose coordinates in the real world. Thanks to the adopted training procedure, which requires only ground truth 3D keypoints, the architecture is independent from how the input 2D coordinates are calculated. As outlined in Section III, the independence from the method that predicts the 2D body joints allows the use of pre-trained 2D human pose estimators, such as OpenPose [4] and HRNet [5], on RGB or IR images. The predicted 2D coordinates needs to be mapped to the depth image then RefiNet can be applied to improve the 3D prediction (see Figure 1). However, authors of the ITOP dataset provide depth images only. Therefore, we train from scratch OpenPose and HRNet on the training set of ITOP using the Adam optimizer, a learning rate of 0.001 and weight decay 0.0001. Since our method is independent from the 2D model, we expect to obtain similar results with both the architectures.

In order to assess the quality of the predictions, we adopt two common evaluation metrics: the *mean Average Precision* (mAP), as proposed in [19], and the *mean Distance Error* (mDE). The mAP is the percentage of predicted joints whose 3D distance from the ground truth is lower than a threshold τ; the mDE is the average distance between the predicted joints and the ground truth. They are defined as

$$\text{mAP} = \frac{1}{N} \sum_{N} \left(\|\mathbf{v} - \mathbf{w}\|_2 < \tau \right) \quad [\%] \tag{5}$$

$$\text{mDE} = \frac{1}{N} \sum_{N} \|\mathbf{v} - \mathbf{w}\|_2 \quad [\text{cm}] \tag{6}$$

where N is the overall number of joints, $\mathbf{v}$ is the predicted joint while $\mathbf{w}$ is the ground truth joint. In our experiments, we set the threshold $\tau = 10$ cm, as in [19].

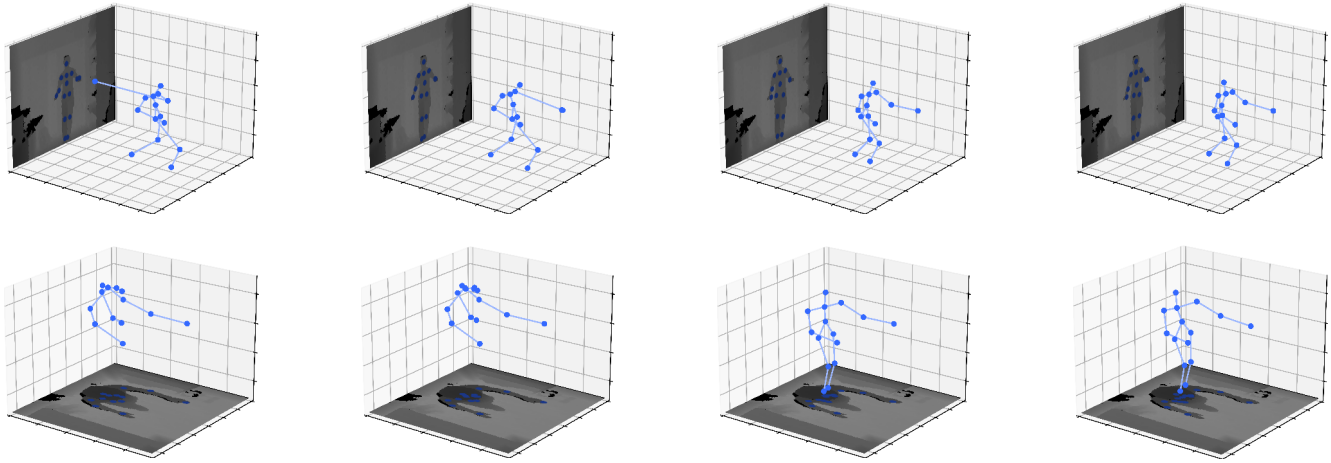
C. Results

The experimental results on the side and the top view of ITOP are reported in Table I. In the left part, a ✓ symbol specifies which modules of RefiNet are employed. The first row contains the results of the plain 2D to 3D pipeline (see Figure 1 top and Section III-A).

As it can be seen, the use of RefiNet with all the modules (top table, last row) achieves the best results on the ITOP Side view, with an overall improvement of about 25% over mAP and one of about 35% over mDE. As expected, refining the output of OpenPose and HRNet leads to similar results, confirming that RefiNet is invariant to different 2D predictors.

As shown in Figure 2, Module A refines the 2D position of the body joints, but the depth values are still inaccurate (due to their sampling from the depth map); Module B refines the 3D joints obtaining an accurate and plausible 3D skeleton; Module C refines the 3D skeleton by looking at the 3D points around the skeleton joints. Other qualitative results, which show the progressive improvement of the RefiNet components, are reported in Figure 4.

On the ITOP Top view, we achieve the best results using only the 2D patch-based module and the skeleton-based module (Mod. A and B) of RefiNet (bottom table, fifth row). In this case, the mAP improvement is around 325% and the mDE one is around 86%. As in the side view scenario, the results are similar regardless of the 2D predictor. On the contrary, in this setting the combined use of the point cloud-based module (Mod. C) with modules A and B does not lead to an improvement of the results due to the data available from this view. In fact, the first module (Mod. A) improves the 2D prediction on the depth map, but the lower part of the body is not visible from the top view then the z-axis coordinate sampled from the depth map is not correct. The



(a) 2D prediction (b) Module A (c) Module B (d) Module C

Fig. 4: Side and top view output samples. Starting from the left, initial 2D prediction [4] (input of RefiNet) then outputs of: 2D patch-based refinement (Mod. A), skeleton-based refinement (Mod. B), and point cloud-based refinement (Mod. C).

TABLE II: Comparison between 3D methods [19], [38], the baseline approach (based on [5]), and the proposed method.

Body part	ITOP side view				ITOP top view			
	[38]	[19]	Bas.	Ours	[38]	[19]	Bas.	Ours
Upper Body	**84.8**	84.0	71.2	77.9	84.8	**91.4**	32.8	72.1
Lower Body	72.5	67.3	62.3	**85.7**	46.1	54.7	0.1	**71.4**
Full Body	80.5	77.4	67.0	**81.8**	68.2	**75.5**	17.5	71.8

second module (Mod. B) learns to improve the 3D skeleton obtaining a plausible pose, in particular improving the depth axis of the lower-body joints. At this point, the third module (Mod. C) should refine the 3D prediction of each joint by looking at the point cloud. However, point clouds computed from the top view are partial or empty for most of the joints, leading to a decrease of the performance.

In Table II, we show a reference comparison between the baseline approach (Fig. 1, Sec. III-A), the proposed method, and the best results reported in [19]. Results show that our method reaches comparable results w.r.t. 3D methods specifically designed to work on depth maps. From the top view, results are lower due to the occlusion of the lower joints by the upper body. It should be noted that the proposed method can leverage on pre-trained deep models that can work "in the wild" and improve their predictions during the 2D to 3D conversion. The improvement w.r.t. the baseline approach is confirmed in all the tested cases.

D. Estimation of anthropometric measurements

In this section we present a real-world application of the proposed method using the Baracca dataset. As mentioned above, the height and other anthropometric measurements can be used to automatically adapt the car to the driver/passenger. In order to achieve a good user experience, the system should have a low average error and a low variance. In fact, not only should the system correctly estimate the height, but also avoid significant mistakes.

TABLE III: Height estimation. We averaged the mean error and std. deviation of each subject, expressed in cm. Baseline (based on [5]) predictions are compared to the refined ones.

Method	Baseline		Ours	
	Mean	Std	Mean	Std
LR	**5.586**	1.468	5.656	**1.330**
AdaBoost	4.347	1.018	**3.372**	**0.755**
RF	2.230	0.377	**1.983**	**0.321**
kNN	**0.783**	0.503	1.276	**0.348**

In order to obtain the subject's height, we firstly predict the 3D human pose with RefiNet. Then, we estimate the height of the subject starting from a set of body limb measures computed as the distance between 2 known 3D joints. Three well-known regression algorithms, such as Linear Regression, k-Nearest Neighbor, and Random Forest, are evaluated. We use HRNet [5] to predict the 2D pose from the depth map, then we refine the prediction with RefiNet, using the modules A and C. The 2D human pose estimator and RefiNet are trained on ITOP, while the tests on the Baracca dataset are conducted without any kind of fine-tuning or model adaptation. We use 3 relevant measures as input to the regressors; among the others we selected the head-neck, the neck-shoulder, and the shoulder-elbow distance (in cm) since the corresponding joints are usually visible from the in-car view. The output measure is the human height, one of the most important information required by car adaptation systems.

For each subject, we estimate his/her height for each frame (*i.e.* 10 images) recorded with 3 different camera positions (*i.e.* with the camera placed on the A pillar, on the rear-view mirror, and behind the steering wheel). We evaluate the quality of the algorithm using the average error and the standard deviation between the predictions and the real heights of the subjects. We performed *leave-one-out* cross validation at subject level, *i.e.* we leave out for test all the available data of a subject.

Average results are reported in Table III. As highlighted in

TABLE IV: Performance analysis of the proposed method. We report the number of parameters, the inference time and the amount of video RAM (VRAM) required by the system.

Model	Parameters (M)	Inference (ms)	VRAM (GB)
OpenPose	52.311	44.859	1.175
HRNet	28.536	43.385	1.107
Module A	0.828	1,872	0.669
Module B	4.302	0.824	0.665
Module C	2.935	13.806	1.681
RefiNet Pipeline	8.064	16.473	1.705

the table, the use of the refined joints slightly enhances the accuracy in terms of average error, but significantly improves the average standard deviation. This confirms that our framework is capable of systematically refine body joints thus correcting pose errors and reducing the skeleton variability.

An analysis of the required computing resources and memory is reported in Table IV. In addition to the 2D pose estimation method, our architecture requires only 16.5 ms to compute using 1.7 GB of RAM on a GPU NVidia 1080Ti.

V. Conclusion

We propose RefiNet, a multi-stage refinement framework which provides an accurate 3D human pose starting from a depth map and a coarse 2D pose. The first module improves the 2D position of joints on the depth map, the second one converts and improves the 3D representation, and the last one enhances the 3D absolute location using point clouds. Experiments on ITOP confirms that RefiNet steadily improves the baseline approach and results are comparable to the ones of 3D models. We also present Baracca, an automotive-oriented dataset containing RGB, IR, depth, and thermal images and anthropometric measurements. Results on Baracca show that the proposed method stabilizes the predicted human skeleton.

References

[1] L. L. Presti and M. La Cascia, "3d skeleton-based human action classification: A survey," *Pattern Recognition*, 2016. 1

[2] G. Borghi, R. Vezzani, and R. Cucchiara, "Fast gesture recognition with multiple stream discrete hmms on 3d skeletons," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 997–1002. 1

[3] M. Carraro, M. Munaro, and E. Menegatti, "Skeleton estimation and tracking by means of depth data fusion from depth camera networks," *Robotics and Autonomous Systems*, 2018. 1

[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017. 1, 2, 6, 7

[5] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *CVPR*, 2019. 1, 2, 6, 7

[6] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," in *arXiv preprint arXiv:1904.07850*, 2019. 1

[7] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*, 2011. 1, 2

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *ECCV*, 2014. 1, 3

[9] G. Borghi, S. Pini, R. Vezzani, and R. Cucchiara, "Mercury: a vision-based framework for driver monitoring," in *International Conference on Intelligent Human Systems Integration*. Springer, 2020, pp. 104–110. 1

[10] F. Manganaro, S. Pini, G. Borghi, R. Vezzani, and R. Cucchiara, "Hand gestures for the human-car interaction: the briareo dataset," in *International Conference on Image Analysis and Processing*. Springer, 2019, pp. 560–571. 1

[11] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-shot multi-person 3d pose estimation from monocular rgb," in *International Conference on 3D Vision (3DV)*, 2018. 1

[12] R. Dabral, N. B. Gundavarapu, R. Mitra, A. Sharma, G. Ramakrishnan, and A. Jain, "Multi-person 3d human pose estimation from monocular images," in *International Conference on 3D Vision (3DV)*, 2019. 1

[13] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016. 2

[14] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *ECCV*, 2016. 2

[15] G. Rogez, P. Weinzaepfel, and C. Schmid, "Lcr-net: Localization-classification-regression for human pose," in *CVPR*, 2017. 2

[16] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *ECCV*, 2018. 2

[17] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," in *ICCV*, 2011. 2

[18] H. Y. Jung, Y. Suh, G. Moon, and K. M. Lee, "A sequential approach to 3d human pose estimation: Separation of localization and identification of body joints," in *ECCV*, 2016. 2

[19] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei, "Towards viewpoint invariant 3d human pose estimation," in *ECCV*, 2016. 2, 5, 6, 7

[20] A. D'Eusanio, S. Pini, G. Borghi, R. Vezzani, and R. Cucchiara, "Manual annotations on depth maps for human pose estimation," in *International Conference on Image Analysis and Processing*, 2019. 2, 5

[21] D. Ballotta, G. Borghi, R. Vezzani, and R. Cucchiara, "Fully convolutional network for head detection with depth images," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 752–757. 2

[22] C. Wu, J. Zhang, S. Savarese, and A. Saxena, "Watch-n-patch: Unsupervised understanding of actions and relations," in *CVPR*, 2015. 2

[23] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real-time human pose tracking from range data," in *ECCV*, 2012. 2

[24] T. Helten, A. Baak, G. Bharaj, M. Müller, H.-P. Seidel, and C. Theobalt, "Personalization and evaluation of a real-time depth-based full body tracker," in *International Conference on 3D Vision (3DV)*, 2013. 2

[25] M. Ye and R. Yang, "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera," in *CVPR*, 2014. 2

[26] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *CVPR*, 2018. 2

[27] A. Bulat and G. Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," in *ECCV*, 2016. 2

[28] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," in *CVPR*, 2016. 2

[29] G. Moon, J. Y. Chang, and K. M. Lee, "Posefix: Model-agnostic general human pose refinement network," in *CVPR*, 2019. 2

[30] M. Ruggero Ronchi and P. Perona, "Benchmarking and error diagnosis in multi-instance pose estimation," in *CVPR*, 2017. 2

[31] M. Fieraru, A. Khoreva, L. Pishchulin, and B. Schiele, "Learning to refine human pose estimation," in *CVPR Workshops*, 2018. 2

[32] Q. Wan, W. Qiu, and A. L. Yuille, "Patch-based 3d human pose refinement," *arXiv preprint arXiv:1905.08231*, 2019. 2

[33] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *CVPR*, 2014. 3

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 3

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2014. 3, 4, 5

[36] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *ICCV*, 2017. 4

[37] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017. 4, 5

[38] H. Yub Jung, S. Lee, Y. Seok Heo, and I. Dong Yun, "Random tree walk toward instantaneous 3d human pose estimation," in *CVPR*, 2015. 7