Pattern Recognition Letters 171 (2023) 185–191

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Depth-based 3D human pose refinement: Evaluating the refinet framework

Andrea D'Eusanio[a], Alessandro Simoni[a], Stefano Pini[a], Guido Borghi[c,*], Roberto Vezzani[a,b], Rita Cucchiara[a,b]

[a] Department of Engineering "Enzo Ferrari" (DIEF), University of Modena and Reggio Emilia, 41125 Modena, Italy
[b] Artificial Intelligence Research and Innovation Center (AIRI), University of Modena and Reggio Emilia, 41125 Modena, Italy
[c] Dipartimento di Informatica Scienza e Ingegneria (DISI), University of Bologna, 47521 Cesena, Italy

ARTICLE INFO

Article history:
Received 22 April 2021
Revised 24 November 2022
Accepted 4 March 2023
Available online 7 March 2023

Edited by: Maria De Marsico

MSC:
68T07
68T45

Keywords:
3D Human pose estimation
Human pose refinement
Depth maps
Point cloud

ABSTRACT

In recent years, Human Pose Estimation has achieved impressive results on RGB images. The advent of deep learning architectures and large annotated datasets have contributed to these achievements. However, little has been done towards estimating the human pose using depth maps, and especially towards obtaining a precise 3D body joint localization. To fill this gap, this paper presents RefiNet, a depth-based 3D human pose refinement framework. Given a depth map and an initial coarse 2D human pose, RefiNet regresses a fine 3D pose. The framework is composed of three modules, based on different data representations, i.e. 2D depth patches, 3D human skeletons, and point clouds. An extensive experimental evaluation is carried out to investigate the impact of the model hyper-parameters and to compare RefiNet with off-the-shelf 2D methods and literature approaches. Results confirm the effectiveness of the proposed framework and its limited computational requirements.

1. Introduction

The representation of human poses through body joints is shown to be informative enough to enable subsequent tasks such as Action Recognition [1,2] and People Tracking [3]. Therefore, Human Pose Estimation (HPE) techniques are a key element for scene and people understanding.

Recently, several methods based on RGB images, deep learning algorithms [4–6], and large RGB datasets (*e.g.*COCO [7]) have achieved stunning results and impressive performance in terms of accuracy, generalization capabilities, and computational load. These literature methods usually provide the human pose as 2D locations of the body joints in image coordinates, ignoring the depth dimension and any metric information.

Recently, promising results have been obtained by 3D estimator based only on intensity images [8,9]. However, these approaches do not regress the 3D pose in the camera coordinate system and the scale of the human body: indeed, the use of 3D information is strictly required to overcome the aforementioned limitations and depth maps provided by active depth sensors can be an effective solution in place or in addition to RGB cameras.

With this in mind, we propose to combine off-the-shelf 2D Human Pose Estimation (2D HPE) methods with the 3D information provided by depth cameras in the form of depth maps. We present a modular framework, namely *RefiNet*, that recovers a precise 3D human pose in camera space through a series of refinements of an initial coarse 2D estimation and a depth map. The system is composed of three independent modules: the first one refines the initial 2D locations of the joints using the depth map, the second module refines the 3D human skeleton as a whole, the third module refines the 3D joint locations using the point cloud computed from the depth map. Each module is separately trained and can be independently enabled or disabled. For the initial 2D HPE, we rely on existing 2D methods (*e.g.*[4–6]). However, thanks to the adopted training procedure, the system does not rely on any specific 2D HPE model.

In this paper, extending our previous work [10], we conduct an in-depth analysis of the hyper-parameters of each module and their influence on the final performance of the whole system. We

* Corresponding author.
E-mail addresses: a.deusanio@unimore.it (A. D'Eusanio), alessandro.simoni@unimore.it (A. Simoni), s.pini@unimore.it (S. Pini), guido.borghi@unibo.it (G. Borghi), roberto.vezzani@unimore.it (R. Vezzani), rita.cucchiara@unimore.it (R. Cucchiara).

https://doi.org/10.1016/j.patrec.2023.03.005

also compare our proposal with literature competitors and discuss the advantages and disadvantages of the analyzed solutions. Moreover, we propose an updated version of the second module of RefiNet, obtaining a significant performance improvement. Finally, we investigate the portability of the whole system evaluating its performance on both GPUs and CPUs. We publicly release the code at https://aimagelab.ing.unimore.it/go/3d-human-pose-refinement.

2. Related work

Human Pose on RGB images. The large majority of HPE methods available in the literature are based on RGB images. Indeed, state-of-the-art human pose estimators exploit CNNs [4,5,11–14] and large annotated datasets to provide 2D poses. One of the pioneering methods based on deep learning is the one described by Wei et al. [11], who proposed a sequential architecture that iteratively estimates the 2D location of body joints. The evolution of this paradigm is represented by the method proposed by Cao et al. [4], in which the Part Affinity Fields are used to learn the links between the body parts of each person. More recently, Sun et al. [5] addressed the problem of preserving high-resolution representations along the whole pose estimation pipeline, through the use of multi-scale fusions at model architecture level. Aware of the remarkable accuracy obtained by these deep learning approaches, we propose to exploit them in conjunction with other input types, *e.g.*the depth data.

Human Pose on Depth Maps. Depth maps are an uncommon data type for the human pose estimation models, probably due to the lack of publicly released datasets with synthetic or real depth data and the limited size, in terms of annotated data, of the existing ones. In addition, the method proposed by Shotton et al. [15] is often used to automatically annotate depth data with body joint locations, resulting in unreliable and imprecise annotations. This method is widely used due to its implementation in the *Microsoft Kinect SDK*: based on Random Forest and trained using a synthetic proprietary dataset, it obtains real time speed and a reasonable accuracy. Hough forests are used by Girshick et al. [16] to regress the coordinates of visible and occluded body joints directly from depth maps. In the work of Jung et al. [17], a nearest neighbor approach, combined with random trees, is used to localize body joints from single depth maps. The remaining works [18,19] are based on accurate but expensive 3D scanners: they usually propose techniques to match fixed pre-defined body models to the acquired point cloud.

As mentioned above, there is a limited number of depth-based datasets for HPE in the literature. The ITOP (*Invariant Top View*) dataset [20] contains about 50k low-quality depth images acquired placing two depth sensors (*Asus Xtion Pro*) in the top and the side point of view; body joints are manually annotated. A new dataset has been recently proposed by DEusanio et al. [21], in which the pose annotations of the Watch-n-Patch dataset [22] have been manually refined, for more than 3k frames. In addition, a multi-stage and fully convolutional neural network is proposed, obtaining real time speed and a good accuracy [23].

Human Pose Refinement. Similar to the Human Pose Estimation task, most of the Human Pose Refinement methods are based on 2D intensity images. In general, several methods [12,24,25] are based on multi-stage architectures, trained through an end-to-end procedure, in which each stage of the model refines the previous predictions. Others [26] exploit a shared weight model to estimate the error on the pose prediction. As shown by Moon et al. [27], all of these methods merge the pose estimation and the refinement task in a single model, obtaining a refinement framework in which the pose estimation and its refinement are strictly dependent on each other. In this paper, we propose an approach that overcomes this issue. Zhang et al. [28] recently proposed a method that predicts an initial 3D pose which is then refined by a point cloud-based network. Wan et al. [29] proposed an approach, based on RGB and segmentation images, that focuses on body parts to refine a 3D pose. The lack of training data is addressed specifically by Moon et al. [27], proposing a model-agnostic human pose refinement network that is trained with synthetic data expressly generated with the error statistics presented by Ruggero Ronchi and Perona [30,31] introduced a similar approach: a deep neural model is trained using a synthetic dataset created with ad-hoc rules.

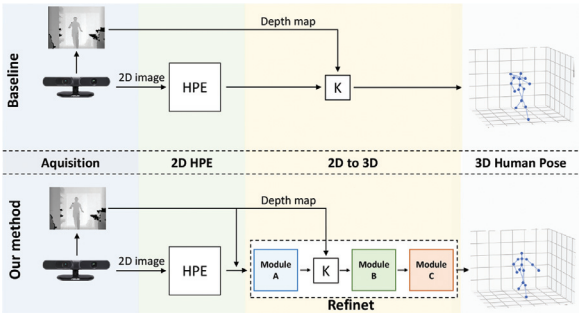


Fig. 1. RefiNet is a modular framework that, given an initial 2D HPE and a depth map, computes a refined 3D pose. The block denoted with the letter K is the mapping between 2D and 3D coordinates, requiring camera calibration parameters and depth values.

3. Proposed method

RefiNet is a modular framework composed of three different modules that, given as input a depth image and a set of 2D image coordinates of the body joints, outputs a refined and accurate 3D human pose in camera space, *i.e.*in the absolute 3D camera coordinate system. Fig. 1 shows a visual summary of RefiNet, highlighting the differences with a conventional baseline model that directly outputs a 3D pose sampling the z coordinate from the depth map, without any refinement procedure. In the proposed system, the initial 2D HPE is provided by an existing off-the-shelf method. The three different modules, here referred to as (Module A, B, and C) are further detailed in the following sections and an overall pipeline is reported in Fig. 2. During the training phase, each module is independently trained, *i.e.*the output of the previous module is not requested to train the following one. On the contrary, Gaussian noise is added to the ground-truth annotations and these noisy joints are exploited as training data. In this way, each module is capable of refining the input noisy pose during the testing phase.

3.1. Initial 2D human pose estimation

As mentioned above, RefiNet starts its computation from an initial 2D human pose obtained from a depth or RGB image. This initial body pose can be computed using any off-the-shelf pose estimator available in the literature, applied on a 2D image, represented, for instance, by an RGB image, or a depth map (encoded as grey-level image), or an IR amplitude map. Certainly, the best results would be obtained by training and testing human pose estimation on huge RGB datasets, such as *COCO* [7] and *MPII Human Pose* [32]. However, not all active depth cameras are coupled with RGB sensors thus able to provide both intensity and depth data at the same time. Moreover, coordinate translation and parallax issues between the RGB channel and the depth one should be taken into account. The use of IR amplitude images available on ToF cameras can represent a partial solution since IR and depth are aligned by definition (*i.e.*they are acquired by the same sensor). On the other hand, the pose estimation methods may perform worse or even not work on this kind of data. Therefore, we decide to train

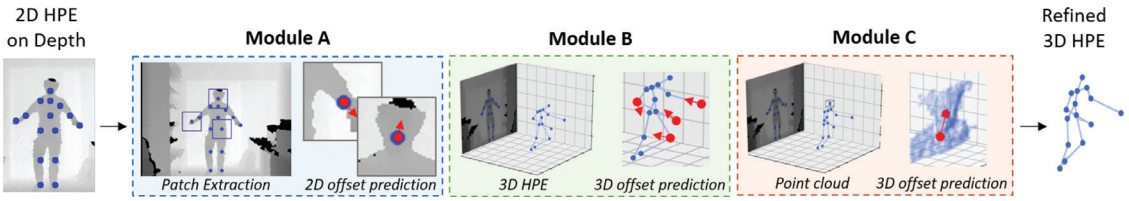
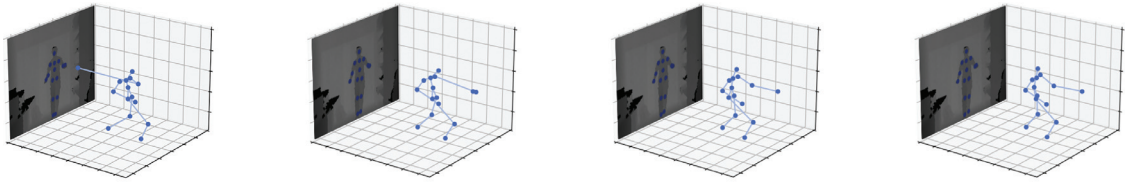


Fig. 2. Overview of the modules that compose the *RefiNet* framework. Module A receives 2D depth patches, extracted from depth maps, and provides 2D offsets. Module B takes the whole 3D skeleton as input while Module C analyzes point clouds sampled around joints. Both Module B and Module C output 3D offsets..



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

Fig. 3. Visual examples of the output of each module of the RefiNet framework. Starting from the left, the initial 2D HPE, with the related depth map, is the input of the framework. Then Module A refines the skeleton through 2D patches, while Module B and Module C work on the 3D skeleton and point cloud, respectively..

2D state-of-the-art methods on depth images from scratch to provide the initial pose to the proposed framework. Once the 2D HPE is computed on the input depth image, each joint is associated to a specific depth value. Thus, including also the camera calibration parameters, it is possible to compute the 3D coordinates of each joint in camera space. Unfortunately, this translating procedure always introduces an approximation since, 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. RefiNet directly addresses the aforementioned problems.

3.2. General training procedure

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

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

$$L_o = \frac{1}{n}\sum_{i=1}^{n} W^i \cdot \left\| \delta^i - t^i \right\|_2^2 \tag{1}$$

where n is the number of joints of the skeleton and $W^i \in \{0, 1\}$ is the visibility mask for the i-th joint (which is 1 if it is visible, 0 otherwise). For each joint, δ^i and t^i are the predicted and the ground truth displacements.

3.3. Module a: 2D patch-based refinement

Module A aims to refine the initial 2D pose estimation through the prediction of 2D offsets. The input is represented by a depth map and the body joint coordinates computed on it, expressed in the (x, y) form. This module is composed of a model that receives as input 2D patches cropped around each predicted joint in the depth map. The output is a displacement vector $\delta = (\delta_x, \delta_y)$ which denotes the displacement of each joint with respect to its initial position: indeed, each final joint position is computed as $x + \delta_x, y + \delta_y$, *i.e.*the sum of the input coordinates and the predicted offset. Module A is specifically designed to correct the small errors in the 2D human pose that negatively influence the sampled z-value, resulting in inaccurate 3D coordinates. A visual example of this procedure is shown in Fig. 3: the z-value of the left elbow after Module A (Fig. 3a) is more accurate than the initial one (Fig.).

Model. The model is a deep neural network consisting of 3 different blocks. The first one receives in input a patch and applies a single 7×7 convolutional layer with 64 feature maps and reduces the spatial dimension with a max-pooling layer with stride $s = 2$. Taking inspiration from He et al. [33], the feature maps are then fed to 2 residual layers with 64 and 128 channels and stride $s = 2$. Finally, the feature maps are grouped using an average pooling layer and used as input for a sequence of 3 fully connected layers with 256, 256, and 2 hidden units. From a general point of view, Module A learns to predict 2D coordinate displacements for each patch independently.

Training. We apply a normalization procedure on each patch, in order to obtain zero-mean and unit-variance tensors. We directly apply random Gaussian noise to the input 2D joints. We investigate the influence of the standard deviation σ of the error distribution in the experimental section. The optimizer is *Adam* [34] with base learning rate of 10^{-3}, in combination with batch normalization and random dropout.

3.4. Module b: Skeleton-based refinement

The goal of the second module is to refine the 3D human pose estimation relying only on information provided by the 3D skeleton. The input of Module B is represented by the 2D body joint coordinates and the depth map, while the output is the same coordinates but in the 3D camera space. As first step, the bidimensional (x, y) input is converted in real-world coordinates (x_C, y_C, z_C) using the camera calibration parameters $K = \{f_x, f_y, c_x, c_y\}$. The coordinate conversion process is computed as follows:

$$(x_C,\ y_C,\ z_C) = \left((x - c_x) \cdot \frac{z}{f_x},\ (y - c_y) \cdot \frac{z}{f_y},\ z \right) \tag{2}$$

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

sampled z is calculated as the median value within a 3×3 neighborhood centered in (x, y). Then, the 3D human skeleton – expressed as the set of 3D body joints – is fed to the deep model described below. Similarly to Module A and differently from the previous version [10], an offset is regressed for each body joint, in order to move the joints from the incorrect location to the most plausible one. Each predicted offset is a three-dimensional displacement vector $\delta = (\delta_x, \delta_y, \delta_z)$ between the location (x_C, y_C, z_C) of each input joint, expressed in camera-space coordinates, and the refined position.

Model. The deep architecture of Module B is inspired by the work of Martinez et al. [35]. Specifically, the model is based on a sequence of 4 blocks. The first one is a fully-connected layer with 1024 units. This block is followed by two residual blocks, each containing 2 fully-connected layers with 1024 units. Finally, the output block is a fully-connected layer with $n \cdot 3$ units where n is the number of skeleton joints. We adopt the same setting for each fully-connected layer, using batch normalization and ReLU as activation function.

Training. For training, a random Gaussian noise is applied on the (x, y) ground-truth coordinates, before sampling the z-value from the depth map. We observe that this procedure can effectively simulate the initial error of a 2D human pose estimator, and thus be used to train a model able to refine 3D coordinates retrieved from the augmented 2D ones. This procedure can lead to large variations on the z axis when coordinates are lifted from 2D to 3D, since little variations on the (x, y) plane can correspond to drastically different depth values (*e.g.*see Fig.).

In this case, the 3D predicted displacement and the corresponding ground truth of the i-th joint, defined in the loss function L_o (see Eq. 1), are defined as $\delta^i = (\delta^i_x, \delta^i_y, \delta^i_z)$ and $t^i = (x^i_C, y^i_C, z^i_C)$. As in Module A, the learning rate is set to 10^{-3} and Adam [34] is adopted as optimizer.

3.5. Module c: Point cloud-based refinement

The third module aims to refine the joint locations relying on the 3D information of the point cloud sampled around each joint of the skeleton. Thus, the input of Module C is a point cloud, which is computed starting from the depth map and the camera calibration parameters, here referred as K. Then, the point cloud is sampled around the center of each joint. Starting from an initial 3D volume size, we expand it progressively until it contains a minimum amount of points or reaches a predefined maximum size, set to 150 mm higher than the initial 3D size. As minimum and maximum number of points, in contrast to the values ($[128, 2000]$) used in the previous version of RefiNet, described in [10], we used $[32, 512]$, which we empirically select as the best trade-off between accuracy and inference speed. If the number of points in the volume is higher than the maximum, we randomly drop the exceeding points. We decide to sample and analyze small point clouds instead of considering the whole point cloud, ranging from the head to the feet of the subject. This approach reduces the computational load and the GPU memory requirements. The recent *PointNet* architecture [36] is exploited to regress a 3D displacement for each body joint. Each regressed offset is expressed as the displacement vector $\delta = (\delta_x, \delta_y, \delta_z)$ between the input locations of the (x_C, y_C, z_C) joint coordinates in camera space and the refined ones.

Model. The model architecture is inspired by the work of Qi et al. [36] and consists of two different blocks with different goals: the first is responsible for the feature extraction while the second one for the offset regression. The single-point features computed by the first block are aggregated through a max-pooling layer and then used as input for the second block. The second block consists of a fully-connected layer with 128 units, a ReLU activation, and an output layer with 3 units, corresponding to the 3D displacement vector.

Training. The training procedure of this module follows the paradigm used for the previous modules, *i.e.*using random Gaussian noise applied on the 3D ground-truth annotations available in the train dataset. Preserving the 3D camera space in which the module works, the noise is added to the (x_C, y_C, z_C) coordinates of each joint before the crop of the point cloud.

As in the previous module, the 3D predicted displacement and the corresponding ground truth of the i-th joint, used in the loss L_o (see Eq. 1), are defined as $\delta^i = (\delta^i_x, \delta^i_y, \delta^i_z)$ and $t^i = (x^i_C, y^i_C, z^i_C)$. We set the initial learning rate to 10^{-3} and use the Adam [34] optimizer, along with dropout (with drop probability $p = .2$) and batch normalization.

4. Experimental evaluation

4.1. Dataset

One of the main limitations of using depth data is the lack of datasets containing depth maps, specifically collected for the Human Pose Estimation. Moreover, several existing datasets include only the joint annotations placed on the body surface, *e.g.*[21], while datasets obtained using *Mocap* systems are not always reliable because the depth value of body joints usually correspond to the markers placed near the body surface. It is also important to note that using markers alters the visual appearance and the 3D shape of the person.

In our experimental validation, we use the ITOP dataset [20], which has been acquired using two *Asus Xtion Pro*, a Structured Light depth sensor having a resolution of 320×240 pixels. The dataset consists of about 40k training and 10k testing depth maps of 20 subjects performing 15 different actions. One sensor is placed above ("top-view") and the other one in front of ("side-view") the acquired subject. Annotations consist of 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.

In this paper, we focused on the "side-view" part of the dataset, which contains recordings from the usual frontal view.

4.2. Experiments

For our experimental evaluation, we adopt two state-of-the-art 2D human pose estimators, *i.e.*OpenPose [4] and HRNet [5]. We train them from scratch on the ITOP dataset using the Adam optimizer, a learning rate of 10^{-3} and weight decay 10^{-4}. We expect to obtain similar results with both the architectures, since RefiNet is independent of the method that predicts the initial 2D body joints.

We adopt two common evaluation metrics in the HPE field in order to assess the overall quality of RefiNet framework: the *mean Average Precision* (mAP), as proposed by Haque et al. [20], 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:

$$\text{mAP} = \frac{1}{n} \sum_n \left(\|\mathbf{v} - \mathbf{w}\|_2 < \tau \right) \quad [\%] \tag{3}$$

$$\text{mDE} = \frac{1}{n} \sum_n \|\mathbf{v} - \mathbf{w}\|_2 \quad [\text{cm}] \tag{4}$$

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 [20].

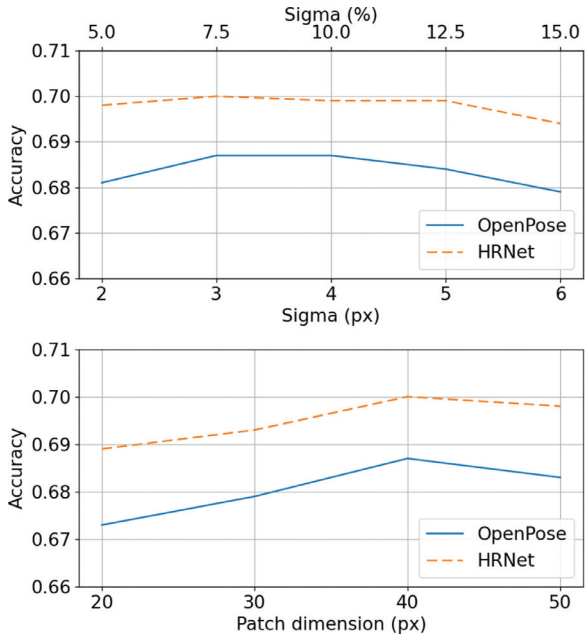


Fig. 4. Effects of Gaussian noise σ (top) and 2D patch size (bottom) on mAP accuracy of Module A. OpenPose and HRNet refer to the initial set of joints..

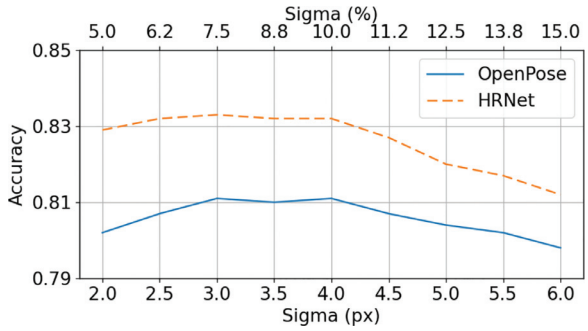


Fig. 5. Effects of Gaussian noise σ on mAP accuracy of Module B.

Module A. Two key hyper-parameters of Module A are the standard deviation of the Gaussian noise and the 2D patch size. Both the parameters are expressed in pixels.

As shown in Fig. 4 (top), the standard deviation of the random Gaussian noise added to the ground-truth joints has a limited impact: this element confirms the ability of Module A to correctly refine the original pose. We observe an accuracy peak at $\sigma = 3$ (corresponding to the 7.5% of the patch size), which is the value that is used in the experiments reported in Section 4.3. On the other hand, the 2D patch size has a higher impact on performance, as shown in Fig. 4 (bottom). In this case, we use a patch size of 40×40 pixels in the rest of the experiments.

Module B. For the Module B, we evaluate the main hyper-parameter of the added Gaussian noise, *i.e.*its standard deviation. Also in this case, the parameter is expressed in pixels since the noise is added to the 2D human pose (to simulate the error of an inaccurate 2D human pose estimator).

As shown in Fig. 5, the hyper-parameter has a substantial impact on the performance of this module. The higher accuracy is obtained using $\sigma \in [3.0, 4.0]$. Thus, we use $\sigma = 3.0$ in the following experiments.

Module C. For the last module, we consider two hyper-parameters: the standard deviation of the added Gaussian noise and the size of the considered 3D volume. Both the parameters are expressed in millimeters.

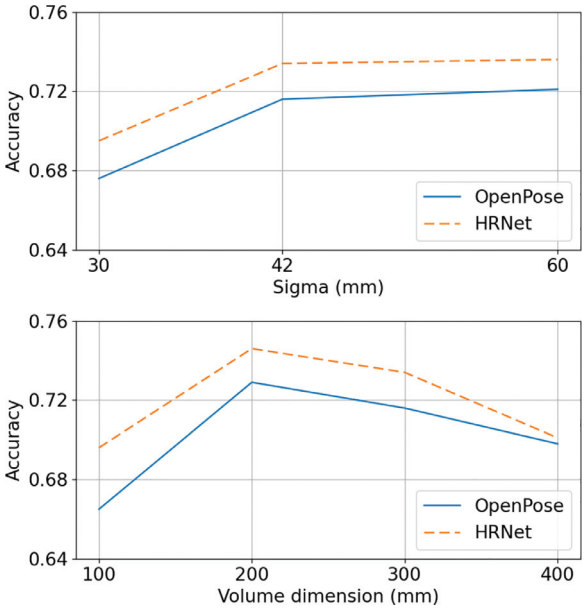


Fig. 6. Effects of Gaussian noise σ (top) and 3D patch size (bottom) on mAP accuracy of Module C. OpenPose and HRNet refer to the initial set of joints..

As shown in Fig. 6 (top), the standard deviation of the random Gaussian noise added to the 3D ground-truth joints has a limited impact, with an accuracy peak in the range $\sigma \in [42, 60]$. In the following experiments, we set $\sigma = 42$. As in Module A, the initial size of the considered 3D volume has a higher impact on performance, as shown in Fig. 6 (bottom). In this case, we set a volume size of $200 \times 200 \times 200$ mm in the experiments of the following Section.

4.3. Results

Experimental results obtained on the ITOP dataset are reported in Table 1 and compared with our previous work [10]. In addition, as an ablation study, we report the results obtained exploiting only one module at a time, indicated with a $\surd$ symbol, during the testing phase. It is worth noting that RefiNet framework leads to better results with an overall improvement of about 27% over mAP and about 37% over mDE. As expected, refining the output of OpenPose and HRNet leads to similar results, confirming that RefiNet is invariant to different off-the-shelf 2D predictors.

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

We compare the proposed framework with literature methods in Table 2. Following the literature convention [20], we present the mAP metric divided into the upper and lower body parts in addition to the full body. Specifically, we report the results from the work of Haque et al. [20], our first version of the proposed framework [10], the recent method proposed by Zhang et al. [28], and a baseline approach. For the baseline, we first obtain the 2D joint locations through HRNet [5], then the 3D joints are computed using the 2D locations, the corresponding z-values taken from the depth map, and the camera calibration parameters. Experimental results show that RefiNet achieves comparable accuracy with the respect to methods designed to directly work on depth images. Specifically, the proposed framework effectively improves the predictions ob-

Table 1
Results in terms of mAP and mDE obtained on ITOP dataset. Mod. A, Mod. B and Mod. C refer to the three modules of RefiNet. Improvements are computed with the respect to poses obtained with the initial 2D pose estimators. The ✓ symbol indicates that the module is used for the refinement, since in RefiNet each module can be independently enabled or disabled.

Refinement Method	Mod. A	Mod. B	Mod. C	OpenPose [4]				HRNet [5]			
				mAP ↑	Improv.	mDE ↓	Improv.	mAP ↑	Improv.	mDE ↓	Improv.
None				0.646	-	12.634	-	0.670	-	10.711	-
[10]	✓			0.687	6.35%	10.442	17.4%	0.699	4.32%	10.060	6.08%
		✓		0.775	20.0%	8.463	33.0%	0.787	17.5%	8.185	23.6%
			✓	0.719	11.3%	11.834	6.33%	0.734	9.55%	10.693	0.17%
	✓	✓	✓	0.818	26.6%	7.646	39.5%	0.824	23.0%	7.447	30.5%
Ours	✓			0.687	6.35%	10.415	17.6%	0.700	4.48%	9.994	6.69%
		✓		0.811	25.5%	8.258	34.6%	0.833	24.3%	8.335	22.2%
			✓	0.735	13.8%	11.630	7.95%	0.752	12.2%	10.436	2.57%
	✓	✓	✓	0.833	28.9%	7.347	41.8%	0.842	25.7%	7.217	32.6%

Table 2
Comparison between 3D HPE methods [20,28,37], the baseline approach (based on HRNet [5]), and the proposed method.

Method	ITOP side view		
	Upper Body	Lower Body	Full Body
Baseline	*71.2*	*62.3*	*67.0*
[37]	84.8	72.5	80.5
[20]	84.0	67.3	77.4
[28]	88.8	94.1	89.6
[10]	77.9	85.7	81.8
Ours	80.8	88.1	84.2

Table 3
Performance analysis in terms of the number of learnable parameters, the amount of RAM and the inference time required by the system.

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

tained from off-the-shelf Human Pose Estimation methods originally developed to work on the 2D domain. The method of Zhang et al. [28] confirms that point clouds are an effective information source for this task. It shows that the adoption of a specific adversarial loss function in combination with a-priori dataset knowledge (the bone length ratio) can improve the mAP, at the expense of training complexity.

We also analyze the computational requirements of RefiNet in terms of number of learnable parameters, required video memory, and inference time (on both CPU and GPU). We evaluate these measures running the framework on a computer equipped with an *Intel i7-7700K* and a *GPU NVidia 1080Ti* and report the results in Table 3. As it can be seen, RefiNet is able to run in real time and the three modules introduce a limited overhead in terms of parameters, memory usage, and inference time w.r.t.the off-the-shelf 2D HPE methods. Compared to Zhang et al. [28], RefiNet is lighter in terms of computational load, running at about 130 fps instead of 24.4 fps.

5. Conclusion

In this paper, we improve and evaluate RefiNet, a modular framework that aims to refine a 3D pose, starting from a depth map and a coarse 2D pose only. Thanks to the adopted training procedure, the system does not depend on any specific off-the-shelf human pose estimator. The modules are independent of each other and introduce a limited overhead in terms of computing resources with the respect to the baseline deep models. Experimental results on ITOP confirm that RefiNet steadily improves the baseline approach and results are comparable to the ones of 3D models. Several future works can be planned. For instance, an adversarial loss can be introduced, having shown to improve the overall accuracy. In addition, the use of the single modules of RefiNet framework can be further investigated. Finally, the proposed pipeline can be also applied and analyzed on a multi-person scenario, in a top-down fashion.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] L.L. Presti, M. La Cascia, 3D skeleton-based human action classification: asurvey, Pattern Recognit. (2016).
[2] G. Borghi, R. Vezzani, R. Cucchiara, Fast gesture recognition with multiple stream discrete hmms on 3d skeletons, ICPR, IEEE, 2016.
[3] M. Carraro, M. Munaro, E. Menegatti, Skeleton estimation and tracking by means of depth data fusion from depth camera networks, Rob. Auton. Syst. (2018).
[4] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, CVPR, 2017.
[5] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, CVPR, 2019.
[6] X. Zhou, D. Wang, P. Krähenbühl, Objects as points, arXiv preprint arXiv:1904.07850, 2019.
[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: common objects in context, ECCV, 2014.
[8] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, C. Theobalt, Single-shot multi-person 3d pose estimation from monocular rgb, in: International Conference on 3D Vision (3DV), 2018.
[9] R. Dabral, N.B. Gundavarapu, R. Mitra, A. Sharma, G. Ramakrishnan, A. Jain, Multi-person 3d human pose estimation from monocular images, in: International Conference on 3D Vision (3DV), 2019.
[10] A. D'Eusanio, S. Pini, G. Borghi, R. Vezzani, R. Cucchiara, Refinet: 3d human pose refinement with depth maps, ICPR, 2020.

[11] S.-E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, Convolutional pose machines, CVPR, 2016.
[12] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, ECCV, 2016.
[13] G. Rogez, P. Weinzaepfel, C. Schmid, Lcr-net: localization-classification-regression for human pose, CVPR, 2017.
[14] B. Xiao, H. Wu, Y. Wei, Simple baselines for human pose estimation and tracking, ECCV, 2018.
[15] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, CVPR, 2011.
[16] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, A. Fitzgibbon, Efficient regression of general-activity human poses from depth images, ICCV, 2011.
[17] H.Y. Jung, Y. Suh, G. Moon, K.M. Lee, A sequential approach to 3d human pose estimation: separation of localization and identification of body joints, ECCV, 2016.
[18] V. Ganapathi, C. Plagemann, D. Koller, S. Thrun, Real-time human pose tracking from range data, ECCV, 2012.
[19] T. Helten, A. Baak, G. Bharaj, M. Müller, H.-P. Seidel, C. Theobalt, Personalization and evaluation of a real-time depth-based full body tracker, in: International Conference on 3D Vision (3DV), 2013.
[20] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, L. Fei-Fei, Towards viewpoint invariant 3d human pose estimation, ECCV, 2016.
[21] A. DEusanio, S. Pini, G. Borghi, R. Vezzani, R. Cucchiara, Manual annotations on depth maps for human pose estimation, ICIAP, 2019.
[22] C. Wu, J. Zhang, S. Savarese, A. Saxena, Watch-n-patch: unsupervised understanding of actions and relations, CVPR, 2015.
[23] D. Ballotta, G. Borghi, R. Vezzani, R. Cucchiara, Fully convolutional network for head detection with depth images, in: ICPR, IEEE, 2018, pp. 752–757.
[24] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, J. Sun, Cascaded pyramid network for multi-person pose estimation, CVPR, 2018.
[25] A. Bulat, G. Tzimiropoulos, Human pose estimation via convolutional part heatmap regression, ECCV, 2016.
[26] J. Carreira, P. Agrawal, K. Fragkiadaki, J. Malik, Human pose estimation with iterative error feedback, CVPR, 2016.
[27] G. Moon, J.Y. Chang, K.M. Lee, Posefix: model-agnostic general human pose refinement network, CVPR, 2019.
[28] Z. Zhang, L. Hu, X. Deng, S. Xia, Weakly supervised adversarial learning for 3d human pose estimation from point clouds, IEEE TVCG (2020).
[29] Q. Wan, W. Qiu, A.L. Yuille, Patch-based 3d human pose refinement, arXiv preprint arXiv:1905.08231 (2019).
[30] M. Ruggero Ronchi, P. Perona, Benchmarking and error diagnosis in multi-instance pose estimation, CVPR, 2017.
[31] M. Fieraru, A. Khoreva, L. Pishchulin, B. Schiele, Learning to refine human pose estimation, CVPR Workshops, 2018.
[32] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2d human pose estimation: new benchmark and state of the art analysis, CVPR, 2014.
[33] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR, 2016.
[34] D.P. Kingma, J. B, Adam:a method for stochastic optimization, ICLR (2014).
[35] J. Martinez, R. Hossain, J. Romero, J.J. Little, A simple yet effective baseline for 3d human pose estimation, ICCV, 2017.
[36] C.R. Qi, H. Su, K. Mo, L.J. Guibas, Pointnet: deep learning on point sets for 3d classification and segmentation, CVPR, 2017.
[37] H. Yub Jung, S. Lee, Y. Seok Heo, I. Dong Yun, Random tree walk toward instantaneous 3d human pose estimation, CVPR, 2015.