Real-time Convolutional Networks for Depth-based Human Pose Estimation

Angel Martínez-González*†, Michael Villamizar*, Olivier Canévet* and Jean-Marc Odobez*†

Abstract— We propose to combine recent Convolutional Neural Networks (CNN) models with depth imaging to obtain a reliable and fast multi-person pose estimation algorithm applicable to Human Robot Interaction (HRI) scenarios. Our hypothesis is that depth images contain less structures and are easier to process than RGB images while keeping the required information for human detection and pose inference, thus allowing the use of simpler networks for the task. Our contributions are threefold. (i) we propose a fast and efficient network based on residual blocks (called RPM) for body landmark localization from depth images; (ii) we created a public dataset DIH comprising more than 170k synthetic images of human bodies with various shapes and viewpoints as well as real (annotated) data for evaluation; (iii) we show that our model trained on synthetic data from scratch can perform well on real data, obtaining similar results to larger models initialized with pre-trained networks. It thus provides a good trade-off between performance and computation. Experiments on real data demonstrate the validity of our approach.

I. INTRODUCTION

Person detection and pose estimation are core components for multi-party Human-Robot Interaction (HRI). In particular, social robotics aims to provide the robot with social intelligence to autonomously interact with people and respond appropriately. Detecting people in its surroundings and estimating their pose provide the robot the means for fine-level motion understanding, activity and behavior recognition, and in combination with other modalities, social scene understanding. Although pose estimation has been widely studied, deploying fast and reliable systems remains a challenging task. On one hand, scenario's dynamic nature, i.e. background clutter, multiple pose configurations, between people interaction, and sensing conditions may provoke partial observations hindering the detection process. On the other hand, complex and accurate systems bring high computation burden, disabling the possibility for real-time deployment under limited computational budget.

State-of-the-art. The classical method for body pose estimation is to model spatial relationships of body parts in a graphical model structure, provided part-specific detectors that perform over handcrafted features [24], [11], [10], [3]. Lately, Convolutional Neural Networks (CNN) have been proved to be an effective tool for pose estimation on RGB images. By means of a deep CNN, such a system detects human body parts in the image, which are subsequently parsed to produce body pose estimates.

* Idiap Research Institute, Switzerland. {angel.martinez, michael.villamizar, olivier.canevet, odobez}@idiap.ch
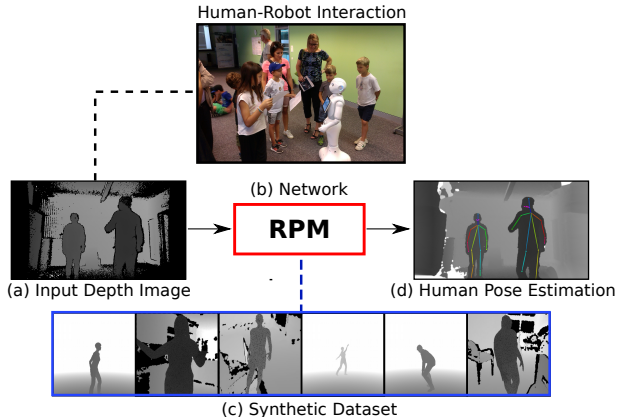† École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

Fig. 1: Overall scheme of the proposed method for efficient human pose estimation (d) from depth images (a). Our proposed RPM convolutional network (b) is trained with depth images consisting of synthetically generated people under multiple poses and positions combined with varying real background depth images (c).

A conventional way to address pose estimation with CNN is inspired by the cascade of detectors concept. That is, sequentially stacking detectors (blocks of convolutional layers) to improve and refine body part predictions using spatial image context. Image context is retrieved by various kernel resolutions [31], [17], [23], [30] or embedding coarse to fine prediction in the network architecture [22], [15], [5].

Spatial relationships between pairs of body parts are also considered in order to improve estimation and ease the inference stage. These relationships can be modeled by explicit regressors [16], [18], or embedded in a network architecture [29], [7]. Motivated by the cascade of detectors concept, [7] relies on recurrent detector blocks to refine predictions and encode body parts pairwise dependencies as a vector field between adjacent parts. Body landmarks[1] and pair-relationships are learned in an end-to-end fashion and jointly predicted in a multi-task approach. Although very deep network models like [7] have provided excellent results, the computational demands of these models grow with the network's depth and require large amounts of training data to prevent overfitting.

Depth data has also been used for pose estimation [25], [21], [28], [19], [9]. Indeed, depth discontinuities and variations preserve many essential features that also appear in natural images like corners, edges, silhouettes. It is also

---

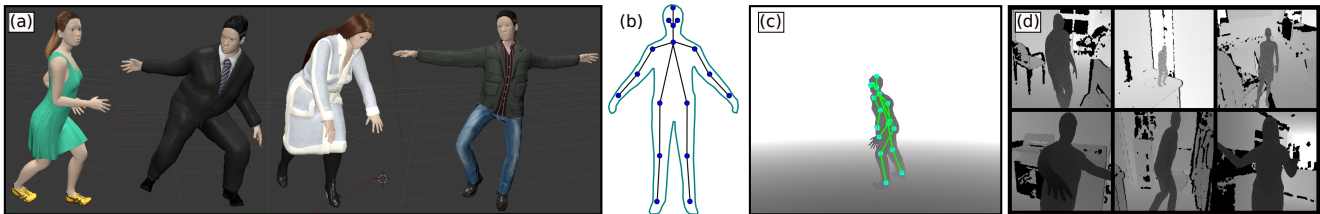[1]In this paper we use body parts and landmarks interchangeably.

Fig. 2: (a) Sample 3D characters with different poses and outfits; (b) skeleton model; (c) rendered synthetic depth image sample; (d) examples of training images, combining synthetic generated bodies with real background images.

texture and color invariant, which may help to remove ambiguities in scale and silhouette shapes. In [25], for example, a random forest based on simple depth features is computed to pixel-wise label the image as belonging to one of the different body parts. The need for training data was addressed by synthesizing depth images with a large variety of human shapes and poses using computer graphics. Despite the remarkable and real-time results, the method assumes background subtraction as a preprocessing step, and is limited to near-frontal pose and close-range observations.

CNN-based methods have also been proposed for articulated pose estimation from depth images [13], [30], [8]. However, these methods normally use an already pre-trained and large network (e.g. VGG [27]) as feature extractor to perform, subsequently, the prediction of human body landmarks. As a consequence, they are not appropriate for real-time pose estimation since such pre-trained networks involves many parameters and increases the computational cost during runtime.

Approach and contributions. Inspired by current advances in CNN, we investigate network architectures that perform on depth images for efficient and reliable pose estimation in social multi-party HRI applications, as illustrated in Figure 1. Depth data provides direct and very relevant information for body landmark detection like head, shoulders, or arms, although the lack of texture may limit its performance where only subtle depth variations are expected (eyes, arms on body). Also, thanks to the depth, moving from landmark localization to the actual 3D body pose will be more straight-forward than with RGB images only. The challenge addressed in this paper is thus to gain speed without sacrifing performance. In that direction, our contributions are:

- we propose a fast and efficient network based on residual blocks, called Residual Pose Machines (RPM), for body landmark localization from depth images;
- we built a dataset of Depth Images of Humans (DIH) comprising more than 170K synthetically generated depth images of humans, and which can be used for training purposes, along with 460 depth real images annotated with body landmarks. The dataset will be made publicly available;
- we demonstrate that models trained on synthetic data can perform well on real data.
- we show that our relatively shallow RPM model trained from scratch obtain similar results to larger models initialized with pre-trained networks, thus providing a

good trade-off between performance and computation.

In Section II, we present our pipeline to build synthetic depth images for training the network. Section III describes the proposed network for efficient pose estimation. Experiments and results are described in Section IV. Finally, Section V concludes the paper.

II. SYNTHETIC DEPTH IMAGE GENERATION

Training CNNs requires large amounts of data with annotations. Unfortunately, a precise manual annotation of depth images with body parts is not so easy, given that people roughly appear as blobs. Fortunately, as shown by Shotton et al. [25], synthesizing depth images of the human body is easier than synthesing real RGB images, since color, texture, and illuminations conditions are much more difficult to render in practice. In this paper, we follow this approach. Rougly speaking, we follow a randomized synthesis pipeline: we created a dataset of 3D human characters, took real motion capture (mocap) data to re-target their pose in the 3D space, selected several random viewpoints, added real backgrounds, and generated the ground-truth, as illustrated in Figure 2. The main issues are how to produce images with enough variations in human shapes, body pose and viewpoint configurations, how to (automatically) annotate these images, and how to simulate realistic backgrounds. This is detailed below.

Variability in body shapes. We built a dataset of 24 adult 3D characters using the modeling software Makehuman [2]. Characters are of both genders and with different heights and weights, and have been dressed with different clothing outfits to increase shape variation (skirts, coats, pullovers, etc.), see Figure 2(a) for a better illustration.

Variability in body poses. For each character, we performed motion retargeting from motion capture data. We relied on the publicly available motion capture database of CMU labs [1], selecting motion capture sequences grossly fitting our scenario in which the robot interacts with people appearing mainly in an upright configuration: people standing still, walking, turning, bending, picking up objects, etc.

Variability in view point. We placed the 3D character in a reference point with a reference orientation, and defined a recording zone as a circle of 8m radius centered at the character. Then, a camera was randomly placed (position and height) in this zone, and its orientation was defined by randomly selecting a point on the character torso and pointing the camera to it. Pose retargeting and depth buffer
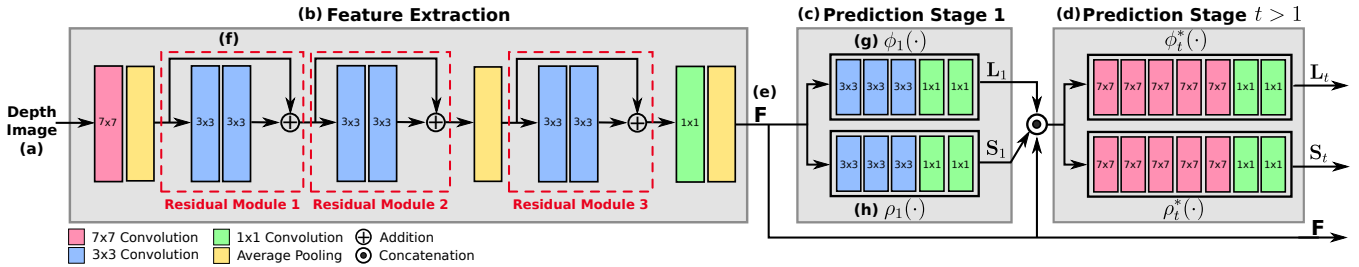
Fig. 3: Architectural design for the Residual Pose Machines (RPM). The input to the network is a single channel depth image (a). The feature extractor (b) is composed of three residual modules (f) producing N_w feature channels. The branches ϕ_t and ρ_t (c,d) predict confidence maps for the location of body parts and limbs.

rendering were performed using the computer graphics software Blender [4], observe Figure 2(c) for an example.

Dataset and annotations. The synthetic images of the DIH public dataset were generated with our synthesis pipeline [2]. They comprise 172148 images of a single person performing different types of motion under different viewpoints. As the human characters come along with 3D semantic joint locations, we were able to automatically record the location of 17 body landmarks (*head, neck, shoulders, elbows, wrists, hips, knees, ankles, eyes*, see Figure 2(b), in the world, camera, and image coordinate systems using the camera's calibration and projection matrices. Each of the 17 keypoints was then automatically labeled as visible or invisible by thresholding the distance between the keypoint and the body surface point closest to the camera located on the line between the keypoint and the camera. In addition, the silhouette's mask was also extracted to allow the incorporation of the body depth images in real depth images (cf below and Section IV-A).

Adding real background. Note that a realistic dataset needs as well realistic background content. Rather than generating a predefined set of images with random (real) background, such images were produced on the fly during training, as described in Section IV-A. Some example images are shown in Figure 2(d).

III. DEPTH-BASED POSE ESTIMATION APPROACH

Our model is inspired by the Convolutional Pose Machines (CPM) [7] approach, which builds a powerful CNN-based 2D body pose detector for color images trained to jointly localize the body parts and limbs of multiple people. In this section we present our network model, with the aim of reducing the number of parameters and speeding up the whole process for robotics applications.

A. Architecture

Overview. Figure 3 depicts the architecture of the proposed efficient network, dubbed Residual Pose Machines (RPM), to detect body parts and limbs, and which takes as input a single channel depth image.

More precisely, the input depth image, see Figure 3(a), is fed into a feature extraction module to get a compact and

discriminative feature representation denoted as **F** (refer to Figure 3(b,e)). Then, these features are passed to a series of prediction stages (Figure 3(c,d)) in order to localize in the image the body landmarks (nose, eyes, ankle, etc.) and limbs (segments between two landmarks according to the skeleton shown in Figure 2 such as forearms, forelegs, etc.).

Each prediction stage consists of two branches made of fully convolutional layers. The first branch, denoted as $\rho_t(\cdot)$, is trained to localize the body parts (Figure 3(h)), while the second one $\phi_t(\cdot)$ is trained to localized the body limbs (Figure 3(g)). The prediction stages are applied sequentially with the goal of refining the predictions of body parts and limbs using the result of the previous stage and incorporating spatial image context.

Finally, pose inference for RPM is performed in a greedy bottom-up step to gather parts and limbs belonging to the same person, in the same way as for CPM. Figure 4 shows some results of RPM on depth images.

Feature extraction network. Depth images exhibit less details than color images (i.e. color and texture), and posture information mainly lies in the person silhouette in combination with the body depth surface. This motivates for the use of a smaller network architecture compared to those used for color images. Therefore, rather than relying on the VGG-19 network used in [7] as feature extractor, we propose to use the smaller and lightweight network architecture shown in Figure 3(b). It consists of an initial convolutional layer followed by three residual modules (or blocks) [14] with small kernel sizes of 3×3. The network has three average pooling layers. Each residual module, see Figure 3(f), consists of two convolutional layers and a shortcut connection (hence the name 'residual' [14]: the inner part of the module is supposed to only model the incremental information since the shortcut represents the identity mapping). Batch normalization and ReLU are included after each convolutional layer and after the shortcut connection.

Our motivation to use residual blocks is that they are known to outperform VGG networks, and to be faster by having a lower computational cost [6].

Confidence maps and part affinity fields prediction. The feature extractor is followed by a succession of stages, each stage taking as input the features **F** and the output of the previous stage. As depicted on Figure 3 and mentioned

before, a stage consists of two branches of convolutional layers, the first branch predicting the location of the parts, and the second predicting the orientation of the limbs. We keep the same design of the branches $\phi_t(\cdot)$ and $\rho_t(\cdot)$ as in the original CPM [7] to maintain the effective receptive field as large as possible. That is, in the first prediction stage the network has three convolutional layers with filters of 3×3 and two layers with filters of 1×1, whereas in the remaining stages there are five and two convolutional layers with filters of 7×7 and 1×1, respectively.

Table I shows a comparison of the number of parameters for different designs of RPM and CPM. Note that with only one stage there is no refinement since the first stage only takes as input the features. Specifically, RPM-1S denotes RPM with one stage while RPM-2S corresponds to the network with two prediction stages (refinement).

B. Training and confidence map ground truthing

We regress confidence maps for the location of the different body parts and predict vector fields for the location and orientation of the body limbs. In this section, for simplicity, we follow the original notation of [7]. The ideal representation of the body part confidence maps $\mathbf{S}^*$ encodes the locations on the depth image as Gaussian peaks. Let $\mathbf{x}_j$ to be the ground truth position of body part j on the image. The value for pixel $\mathbf{p}$ in the confidence map is computed as follows

$$\mathbf{S}_j^*(\mathbf{p}) = exp\left(\frac{||\mathbf{p}_j - \mathbf{x}_j||_2^2}{\sigma}\right), \tag{1}$$

where σ is empirically chosen.

The ideal representation of the limbs $\mathbf{L}^*$ encodes the confidence that two body parts are associated, in addition to information about the orientation of the limbs by means of a vector field. Consider a limb of type c that connects two body parts j_1 and j_2, e.g. elbow and wrist, with positions on the depth image $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$. The ideal limb affinity field at point $\mathbf{p}$ is defined as

$$\mathbf{L}_c^*(\mathbf{p}) = \begin{cases} \mathbf{v}, & \text{if } \mathbf{p} \text{ on limb } c, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where $\mathbf{v}$ is the unit vector that goes from $\mathbf{x}_{j_1}$ to $\mathbf{x}_{j_2}$. The set of pixels that lie on the limb are those within a distance to the line segment that joins the two body parts.

Intermediate supervision is applied at the end of each prediction stage to prevent the network from vanishing gradients. This supervision is implemented by two L_2 loss functions, one for each of the two branches, between the predictions $\mathbf{S}_t$ and $\mathbf{L}_t$ and the ideal representations $\mathbf{S}^*$ and $\mathbf{L}^*$. The loss functions at stage t are

$$f_t^1 = \sum_{\mathbf{p} \in \mathbf{I}} ||\mathbf{S}_t(\mathbf{p}) - \mathbf{S}^*(\mathbf{p})||_2^2, \tag{3}$$

$$f_t^2 = \sum_{\mathbf{p} \in \mathbf{I}} ||\mathbf{L}_t(\mathbf{p}) - \mathbf{L}^*(\mathbf{p})||_2^2. \tag{4}$$

The final multi-task loss is computed as $f = \sum_{t=1}^{T} \left(f_t^1 + f_t^2\right)$ where T is the total number of network stages.

C. Implementation details

Image preprocessing. The depth images are normalized by scaling linearly the depth values in the $[0, 8meter]$ range into the $[-0.5, 0.5]$ range. Furthermore, note that the real data contains noise and missing values, especially around body silhouette due to the sensing process (see Figure 3). Although more advanced domain adaptation techniques could be used to reduce the mismatch between the clean synthetic data and the noisy real data distributions [12], [26], in this paper we considered using a simple inpainting preprocessing to fill out the noise and shadows around the body silhouette and thus prevent sharp discontinuities to potentially affect the network output. This is shown in the experimental section.

Network training. Pytorch is used in all our experiments. We train different network architectures with stochastic gradient descent with momentum for 100K iterations each. We set the momentum to 0.9, the weight decay constant to 5×10^{-4}, and the batch size to 10. We uniformly sample values in the range $[4 \times 10^{-10}, 4 \times 10^{-5}]$ as starting learning rate and decrease it by a factor of 10 when the training loss has settled. All networks are trained from scratch and progressively, i.e. to train network architectures with t stages, we initialize the network with the parameters of the trained network with $t-1$ block detectors.

Part association. We use the algorithm presented in [7] that uses the part affinity fields as confidence to associate the different body parts and perform the pose inference.

IV. EXPERIMENTS AND RESULTS

We conducted experiments using the synthetic images of the DIH dataset as well as on the real depth images of the dataset (described below). We present the experimental protocol in Section IV-B, focusing on both accuracy and computational aspects. The analysis of the results is presented in Section IV-D, where we study the impact on performance of different modeling elements like network architecture or preprocessing.

A. Data preparation

Synthetic data. We split the synthetic images of our DIH dataset into three folds with the following percentage and amount of images: training (85%, 146327), validation (5%, 8606), and testing (10%, 17215).

Training data: data augmentation with real background images. Relying only on clean depth images of the human body may hinder the generalization capacity of a trained network due to the data mismatch with real images. Thus, to avoid our pose detector to overfit clean synthetic image details, we propose to add perturbation to the synthetic images, and in particular, to add real background content which will provide the network with real sensor noise.

Adding real background content. Obtaining real background depth images (which do not require ground-truth) is easier than generating synthetic body images. As backgrounds, we consider the dataset in [20] containing 1367 real depth
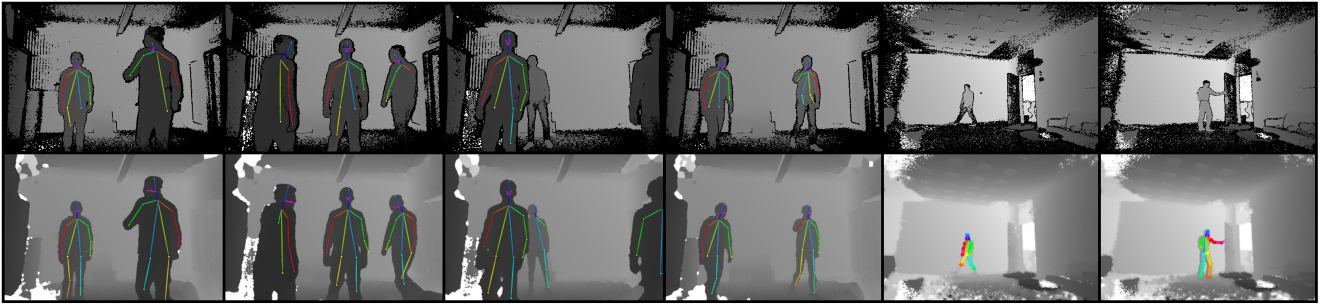
Fig. 4: Output of the proposed RPM-2S for some sequence instances of the testing set. Top: results on raw depth images. Bottom: results on depth images with inpainting.

images recorded with a Kinect 1 and exhibiting depth indoor clutter. We divided it into three folds (training, validation and test) which were associated with the corresponding synthetic body data folds. Then, during learning, training images were produced on the fly by randomly selecting one depth image background and body synthetic images, and compositing a depth image using the character silhouette mask. Care was taken to avoid conflicting depth data (i.e. the character depth value should be in front of the background), and with the character's feet lying on a flat surface. Sample results are shown in Fig. 2(d).

Pixel noise. During training we randomly select 20% of the body silhouette's pixels and set their value to zero.

Image rotation. Training images are rotated with a probability 0.1 by a randomly selected angle in the range $[-30, 30]$ degrees.

Test data. To evaluate the performance of our algorithm, we rely on typical HRI real data captured with a Kinect 2 and exhibiting one or multiple people people passing in front of or interacting with our robot. We manually annotated the body landmarks in 460 periodically sampled images, resulting in a dataset of 546 person instances.

B. Evaluation protocol

Accuracy metric. We use standard precision and recall measures derived from the Percentage of Correct Keypoints (PCKh) evaluation protocol as performance metrics [32]. More precisely, after the forward pass of the network, we extract all the landmark predictions p whose confidence are above a threshold τ, and run the part association algorithm to generate pose estimates from these predictions[3]. Then, for each landmark type, and for each ground truth points q, we associate the closest prediction p (if any) whose distance to q is within a distance threshold $d = \kappa \times h$, where h stands for the height of the bounding box of the person (in the ground truth) to which q belongs to. Such associated p are then counted as true positives, whereas the rest of the landmark predictions are counted as false positives. Ground truth points q with no associated prediction are counted as false negatives. The average recall and precision values can

[3]Note that in this algorithm, landmark keypoints not associated with any estimates are automatically discarded

then be computed by averaging over the landmark types and then over the dataset. Finally, the average recall and precision values used to report performance are computed by averaging the above recall and precision over several distance thresholds d by varying κ in the range $[0.05, 0.15]$.

Computational performance. Model complexity is measured via the number of parameters it comprises, and the number of frames per second (FPS) it can process when considering only the forward pass of the network. This was measured using the median time to process 2K images at resolution 444×368 with an Nvidia card GeForce GTX 1050.

C. Tested models

Proposed model. Our pose detection model is built as in Section III. We configure the network parameters $T = 2$, to profit from spatial context and $N_w = 64$ to balance the speed-accuracy trade-off. We refer to this configuration as RPM-2S. In experiments, we will evaluate the impact of different parameters like: the number of stages T in the cascade of detectors part of the network; the number of N_w feature channels in the residual blocks, and the impact of inpainting preprocessing step.

CPM Baseline. We consider the original architecture presented in [7], trained as our model with the DIH data. As in the original work, the architecture parameters are initialized using the first 10 layers of the VGG-19 network. To accomodate the need for the 3 channel (RGB) image input expected by VGG-19, the single depth channel is repeated three times.

D. Results

Table I compares the performances of the different methods. We report both the average recall and precision for all landmark types in the complete skeleton model (see Fig. 2b)), and for the upper body, i.e. *head, neck, shoulders, elbows and wrists*, since upper-body detection might be sufficient for most typical HRI application. The table also compares the FPS and the number of trainable parameters of the different networks. As an extra experiment, for comparison, we show the results obtained by running the code of [7] over the registered RGB images. For this experiment, the performances were computed over the body parts that the skeleton

Architecture	Stages	w	N. Parameters	FPS	Performance			
					Complete body		Upper body	
					AP	AR	AP	AR
Cao et al [7]	6	–	51.86 M	3.6	69.89*	67.43*	78.75*	78.10*
CPM-1S	1	–	8.38 M	18.6	65.52	51.67	68.74	66.85
CPM-2S	2	–	17.07 M	11.2	70.36	**57.03**	76.00	71.77
RPM-1S	1	64	0.51 M	56.7	64.63	44.34	73.62	57.65
RPM-2S	2	64	2.84 M	35.2	65.46	56.77	74.86	71.96
RPM-3S	3	64	5.17 M	20.8	63.81	56.34	71.05	69.95
RPM-1S	1	128	1.83 M	22.5	45.30	41.59	51.84	55.65
RPM-2S	2	128	10.5 M	12.5	**72.19**	56.11	**84.10**	**72.91**

TABLE I: Comparison of the performance and architecture components for the different tested network architectures. Note that results from [7] (marked with *) are provided for indication, as they are computed over RGB images on a disjoint set of body landmark types than the one we use for depth. See Section IV for details.
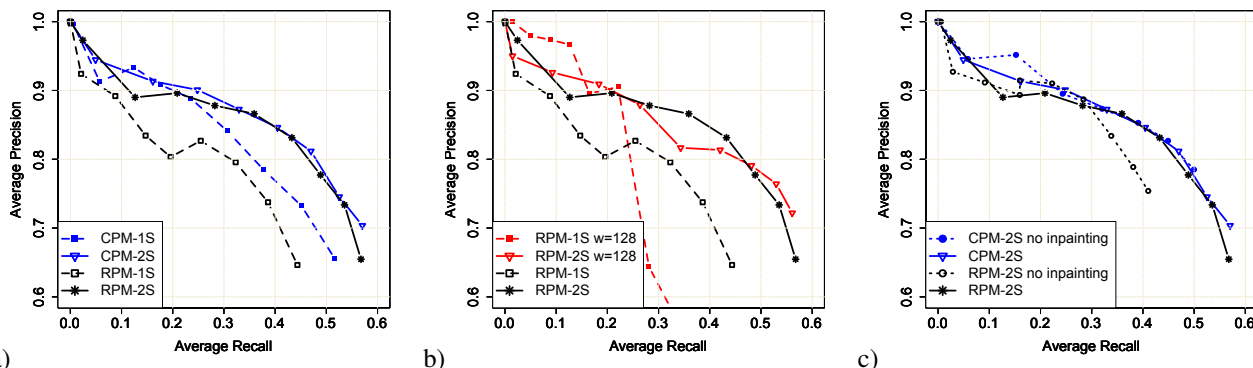

Fig. 5: Average recall-precision curves. a) comparison between the baseline CPM and the proposed RPM networks, up to 2 prediction stages. b) impact of the number of feature channels N_w in the feature extractor CNN. c) impact of the inpainting preprocessing.

model in [7] has in common with our skeleton model. In addition, Figure 5 reports precision-recall curves obtained by varying the τ threshold which impacts the number of detected keypoints before the body part association step.

Analysis. From this table, we can see that our proposed network RPM-2S ($N_w = 64$) performs as well as the baseline CPM-2S (e.g. for upper-body, it has a recall-precision of 72 and 74.8 vs 71.7 and 76 for CPM-2S) but with 6 times less parameters and being 3.14 times faster. Interestingly, we can notice from Fig. 5a that without using recursion (RPM-1S), the smaller complexity of our feature extraction CNN indeed leads to degraded performance compared to the original CPM-1S, but that this gap is filled once the recursion is introduced (compare the RPM-2S and CPM-2S curves).

Number of feature channels N_w. From Table I, increasing it (to 128) for $T = 2$ improves much the performance. Our RPM now outperforms the baseline configuration CPM-2S, especially when considering only the upper body, and is still smaller and slightly faster. However, from Fig. 5b, we can notice that the precision-recall curves are not that different between $N_w = 64$ and $N_w = 128$, somehow mitigating the above conclusion.

Number of recursive stages T. Table I shows that the results saturate when increasing it beyond $T = 2$ (i.e. for $T = 3$).

Inpainting preprocessing. We found this step to be important to improve the performance of the different tested models. This is particularly true for our RPM model, as shown by
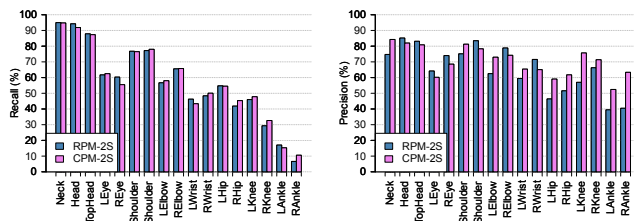

Fig. 6: Recall (left) and precision (right) per body landmark for the proposed RPM-2S and the baseline CPM-2S.

the curves in Figure 5(c). We believe that one explanation is that, since the training and test images (synthetic and real) come from different distributions, this preprocessing removes some of these differences by eliminating noisy details and discontinuities that typically appear in real depth images. Figure 4 shows a qualitative comparison of applying inpainting as preprocessing step. The figure shows typical multi-person HRI scenarios where person occlusion and partial observations are commonly observed.

Performance per body landmark. They are reported in Figure 6 for our RPM-2S model and CPM-2S. Both models show similar recall for the different parts of the skeleton. As for precision, the difference is more notorious in body parts of the lower body. Note that in our testing data, these body parts are the most affected by noise, appearing less well defined and even mixed with the background after the

preprocessing.

V. CONCLUSIONS

This paper has investigated the use of depth images and CNNs to perform fast and reliable human pose estimation. Specifically, we investigated and proposed a real-time neural network architecture with fast forward pass that can be easily deployed in Human-Robot Interaction applications. We also created the DIH dataset comprising a large amount of synthetic images of human-bodies of various shapes and poses, along with real images. This dataset will made publicly available. Our experiments and speed-accuracy trade-off analysis show that on depth images, the smaller CNN architectures we propose achieve similar performance results as larger versions with a much less expensive computational cost.

Our study opens the way to further research. One limitation remains the differences that synthetic depth images exhibit with real ones. While the inpainting preprocessing mitigates this issue, domain adaptation techniques might be more appropriate at bridging the existing gap between the data distributions and transfer realistic details to synthetic images.

ACKNOWLEDGMENTS

REFERENCES

[1] Cmu motion capture data. http://mocap.cs.cmu.edu/.

[2] Makehuman - open source tool for making 3d characters. http://www.makehuman.org/.

[3] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009. Best Paper Award Honorable Mention by IGD.

[4] Blender Online Community. Blender - a 3d modelling and rendering package, 2017. http://www.blender.org.

[5] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, 2016.

[6] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

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

[8] Ben Crabbe, Adeline Paiement, Sion Hannuna, and Majid Mirmehdi. Skeleton-free body pose estimation from depth images for movement analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 312–320, 12 2015.

[9] Meng Ding and Guoliang Fan. Articulated gaussian kernel correlation for human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[10] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *Int. J. Comput. Vision*, 61(1):55–79, January 2005.

[11] Martin Fergie and Aphrodite Galata. Dynamical pose filtering for mixtures of gaussian processes. In *BMVC*, 2012.

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016.

[13] Albert Haque, Boya Peng, Zelun Luo, Alexandre Alahi, Serena Yeung, and Li Fei-Fei. Towards viewpoint invariant 3d human pose estimation. In *European Conference on Computer Vision (ECCV)*, October 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[15] Peiyun Hu and Deva Ramanan. Bottom-up and top-down reasoning with hierarchical rectified gaussians. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5600–5609, 06 2016.

[16] Eldar Insafutdinov, Mykhaylo Andriluka, Leonid Pishchulin, Siyu Tang, Evgeny Levinkov, Bjoern Andres, and Bernt Schiele. Articulated multi-person tracking in the wild. In *CVPR*, 2017. Oral.

[17] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schieke. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision (ECCV)*, 2016.

[18] Umar Iqbal, Anton Milan, and Juergen Gall. Posetrack: Joint multi-person pose estimation and tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] A. Kadkhodamohammadi, A. Gangi, M. de Mathelin, and N. Padoy. A multi-view rgb-d approach for human pose estimation in operating rooms. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, page 10. IEE, Mar 2017.

[20] Kourosh Khoshelham and Er Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. In *Sensors 2012, 12, 14371454. 2013*, page 8238, 2013.

[21] Hanguen Kim, Sangwon Lee, Dongsung Lee, Soonmin Choi, Jinsun Ju, and Hyun Myung. Real-time human pose estimation and gesture recognition from depth images using superpixels and svm classifier. *Sensors*, 15(6):12410–12427, 2015.

[22] Alejandro Newell, Kaiyu Yang, and Jia Deng. *Stacked Hourglass Networks for Human Pose Estimation*, pages 483–499. Springer International Publishing, Cham, 2016.

[23] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[24] Varun Ramakrishna, Daniel Munoz, Martial Hebert, Andrew J. Bagnell, and Yaser Sheikh. Pose machines: Articulated pose estimation via inference machines. In *ECCV*, 2014.

[25] 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 *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1297–1304, Washington, DC, USA, 2011. IEEE Computer Society.

[26] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE Conference on Vision and Pattern Recognition, CVPR*, 2017.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[28] Luciano Spinello and Kai Arras. People detection in rgb-d data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3838–3843, 09 2011.

[29] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1799–1807. Curran Associates, Inc., 2014.

[30] Keze Wang, Shengfu Zhai, Hui Cheng, Xiaodan Liang, and Liang Lin. Human pose estimation from depth images via inference embedded multi-task learning. In *Proceedings of the 2016 ACM on Multimedia Conference*, MM '16, pages 1227–1236, New York, NY, USA, 2016. ACM.

[31] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[32] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2878–2890, December 2013.