Unsupervised Human Pose Estimation on Depth Images

Thibault Blanc-Beyne[1,2*], Axel Carlier[2], Sandrine Mouysset[2], and Vincent Charvillat[2]

[1] Ebhys, ZA La Cigalière III, 84250 Le Thor
[2] Université de Toulouse - IRIT, 2 rue Charles Camichel, 31079 Toulouse
{firstname.name}@irit.fr

Abstract. Human pose estimation is a widely studied problem in the field of computer vision that consists in regressing body joints coordinates from an image. Most state-of-the-art techniques rely on RGB or RGB-D data, but driven by an industrial use-case to prevent musculoskeletal disorders, we focus on estimating human pose based on depth images only. In this paper, we propose an approach for predicting 3D human pose in challenging depth images using an image-to-image translation mechanism. As our dataset only consists in unlabelled data, we generate an annotated set of synthetic depth images using a human3D model that provides geometric features of the pose. To fit the challenging nature of our real depth images as closely as possible, we first refine the synthetic depth images with an image-to-image translation approach using a modified CycleGAN. This architecture is trained to render realistic depth images using synthetic depth images while preserving the human pose. We then use labels from our synthetic data paired to the realistic outputs of the CycleGAN to train a convolutional neural network for pose estimation. Our experiments show that the proposed unsupervised framework achieves good results on both usual and challenging datasets.

Keywords: Depth images · Unsupervised learning · Human pose estimation · Image-to-image translation.

1 Introduction

Musculoskeletal disorders (MSD) are a leading cause of health issues in the workplace [26]. This condition covers various types of injuries that typically affect tendons, muscles, ligaments, etc. Industrial work is often an important source of MSD, due to the repetitive nature of the tasks that the humans operate as well as the physical constraints that are exerted on their limbs. More generally, monitoring working conditions is becoming an international concern [1]; in this context devising systems that can help preventing MSD is an important challenge.

* This work was supported by CIFRE ANRT 2017/0311.

In our work, we study the particular case of the waste industry, in which human operators are especially prone to developing MSD. Both industrial and domestic waste are processed in facilities called waste sorting centers, whose role is to separate different types of waste (for example papers from cardboard and plastics) for later recycling or incinerating. While the sorting process is largely automated, human operators are required to perform negative sorting, i.e. remove unwanted objects to guarantee the purity of the output waste streams. This work involves repetitive movements, with sometimes undesirable limbs angulations, which may eventually lead to MSD.

We aim at designing at system that would help identify risk situations, by monitoring the humans operators posture and revealing harmful angulations. Driven by our industrial context, we should perform these operations under several constraints:

(i) The monitoring should be non-invasive, to avoid affecting the operators work;
(ii) The lightning conditions can not be controlled: the sensor may face a window or be in a dark corner of the facility;
(iii) Anonymity of the workers should be preserved to respect their privacy and avoid abusive uses of the collected data;
(iv) The operators may wear reflecting working clothes.

For all these reasons, we choose to frame this problem as human pose estimation on depth images.

Human pose estimation, which is the task of localizing anatomical keypoints (joints) or parts, has traditionally been a very challenging task in computer vision, both in 2D and 3D. Most work focus on single-person pose estimation in the 2D setting [2, 25] using various architectures of convolutional neural networks (CNN) applied on single RGB images. Another line of work approaches the problem with the goal of detecting multiple persons, still on RGB frames. These more recent methods often use a top-down model, as proposed in [10], where (i) human detection is applied and (ii) single-pose estimation is realized on each detection. However, the most efficient methods rely on bottom-up approaches [7]. These methods make use of ResNet blocks [14] or of a multi-stage CNN to first detect joints, and later match them to perform a complete pose estimation. 3D pose estimation from single images remains a challenge to this day, due to the lack of available labelled data and to the ambiguity of getting 3D information from single images. Using deep learning methods, the 3D pose is often inferred from the 2D estimation [6] as deep neural networks trained on 3D datasets captured in a motion capture context [15] do not generalize well [36]. To tackle this issue, an unsupervised approach based on domain adaptation was recently introduced by [35]. They first train a 2D/3D pose estimator on depth images and a segmentation module on the 2D pose. Using domain adaptation, the authors finally use these two components to train a 2D/3D pose estimator on RGB images.

Even though RGB-D sensors have been used by the general public for many years now (Microsoft released the first version of the Kinect sensor in 2010), only a few research work have focused on how to infer information using depth images

only. In particular, most previous work make use of the human segmentation and pose estimation embedded in the Kinect [29], which implicitly assumes that the user is standing in front of the sensor. As previously adopted by [4] in the context of hospitals for action recognition and fall detection, our approach focuses on using only the sensor depth information (disregarding the color information), in order to respect the human operators privacy.

This implies to estimate human pose from unlabelled, noisy depth images. To do so, we propose in this paper to present a framework of unsupervised pose estimation process, described in Figure 1.

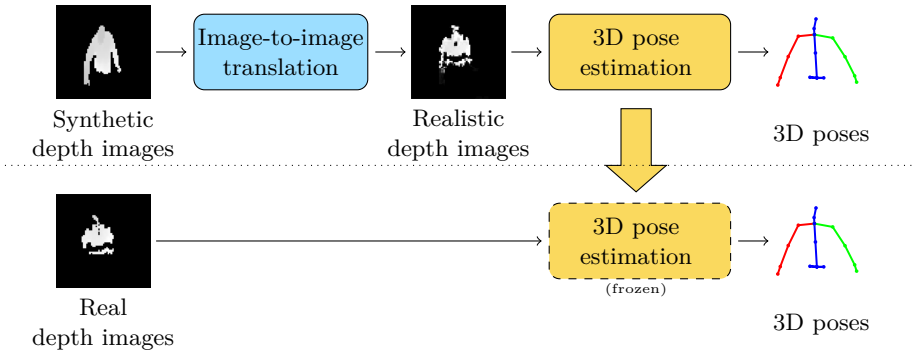

Fig. 1. The framework of our unsupervised pose estimation process.

Our framework relies on the CycleGAN [38] approach, a particular GAN architecture, that is able to preserve the geometric properties of the image while changing its texture. We propose to first generate an annotated set of synthetic depth images using a human 3D model that provides geometric features of the pose (section 3). We then use a modified CycleGAN architecture that alters the generated synthetic depth images to mimic real ones, while preserving geometric content, i.e. the human pose (section 4). In section 5, we present our pose estimation on real images based on a single convolutional neural network that infers the 3D position of body joints. Finally we present our experiments and discuss our results in section 6.

2 Related Work

Human Pose Estimation on depth images. Only a few work on human pose estimation discard the RGB images and only analyze the depth values. Among the recent approaches that build on the advances brought by deep learning techniques, Jiu et al. [17] do not output an explicit body pose, but segment body parts using features built by spatially constrained multiscale, CNNs [11]. Wang et al. [32] introduce an inference-embedded multi-task learning framework. They

use a CNN to generate a heatmap of body parts, which is then fed to an inference network seeking the optimal configuration of body parts, using both appearance and geometric compatibility. Similarly, Haque et al. [13] first detect local body parts. A global body pose is then iteratively produced using a leveraged convolutional and recurrent network containing a long short term memory (LSTM) module. Moon et al. [24] propose a voxel-to-voxel network for hand and body pose estimation. They split the 3D space into a grid of voxels, and estimate a per-voxel likelihood for each joint. Marín-Jiménez et al. [22] present a model which uses a CNN to compute weights used to estimate the 3D pose as a linear combination of prototype poses.

Generative adversarial networks. Generative adversarial network (GAN) is a new machine learning paradigm introduced by Goodfellow et al. in 2014 [12]. In this framework, two models are simultaneously trained: a generative model G which captures the data distribution, and a discriminative model D that separates the data generated by G from the real training data. The training stability of such system is notoriously difficult to maintain, and improvements were proposed to improve it [28]. GANs were also extended by [23] as conditional GAN to make the generator able to sample data conditioned on class labels. The key principle is the use of an adversarial loss, that ideally makes the generated images indistinguishable from real images at the end of the training. GANs have achieved impressive results in image editing [37], generation [27] and conditional generation [34].

Image-to-Image translation. GANs are also used for the task of image-to-image translation. Recent approaches rely on a dataset of paired input/output images, such as "pix2pix" [16], which uses conditional GANs to learn a translation function between input and output images, improved to high resolution images by [33] and by the Multimodal Unsupervised Image-to-image Translation (MUNIT) [39] to control the style of translated output. StarGAN [9] performs image-to-image translations for multiple domains using a single network trained on several datasets, while [8] performs image-to-image translation using pose as latent space. A lot of other methods tackle the unsupervised setting and aim to map two different data domains. An approach is to use a weightsharing strategy as in CoGANs [20] or cross-modal scene networks [3]. Another idea is to encourage the input and the output to share specific content while differing in style [30]. Popular methods use transitivity to regularize the data thanks to cycle-consistency [38]. These approaches rely on the fact that, if an image is translated from one domain to the other and back, the resulting image should ideally be the one we started with. This allows to perform image-to-image translation on unpaired datasets, for instance in the task of semantic segmentation [19].

3 Training data generation

In our industrial context in which human activity is monitored to prevent musculoskeletal disorders, we are bounded by several constraints such as sensors positions, uncontrollable lightning conditions, reflective surfaces and clothes, and

the overwhelming number of moving objects. To protect workers' privacy and favor our system's acceptability, we choose to acquire data using depth cameras. The challenging environment introduces complexity and noise to the depth images and usual human pose estimation approaches fail to produce exploitable results. We build what we will denote as our real dataset by gathering around 60,000 images from 8 live-recorded sessions in a waste sorting center.

Examples of real depth images are depicted in Figure 2. The operator is segmented from the frame using the approach proposed in [5].

Depth data is directly acquired and stored on 640×480, 16 bits images rather than simple 8 bits PNG images, in order to correctly capture the depth value of each pixel. Indeed, precision in depth values is needed to be able to distinguish between the hand of the operator and the object he/she may hold. This kind of storage enables us to perform a better normalization of our data before feeding it to our networks, while PNG images would have provided us with a poorer depth sampling of the operator.

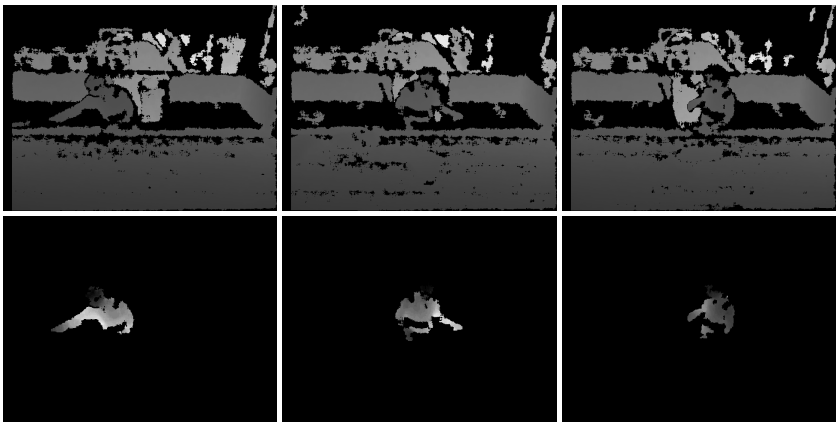

Fig. 2. Examples of images from our dataset (top row) and their associated segmentation (bottom row).

Unfortunately, we can not directly use this data to train our human pose estimation network. Indeed, machine learning often implies a critical need of large amounts of labelled data during the training phase. In our use-case, annotating a 3D human pose just from a depth image is a very difficult and time-consuming task, and in fact almost impossible to realize with a good enough precision as it involves providing a 3D information on a 2D space. In addition, our industrial constraints make it likely that the shooting conditions differ from one sorting center to another. The depth camera may have to be positioned differently or the human operators may wear different reflective clothes, which would result in a new type of data, and would thus require a new annotation campaign to retrain the neural network. We believe this process is not sustainable in our use-

case. That is why we decided to rely on a synthetic labelled dataset rather than handcrafted annotations.

The synthetic data is generated using the free and open source 3D creation suite Blender[3]. The rigged 3D human model was created using an free opensource software called MakeHuman[4]. We made use of the Blender scripting tool to generate a highly variable set of poses stored as key frames, while its compositing tool allowed us to render synthetic depth frames. The range of angulations for each joint of our animated model was built thanks to an ergonomic study to closely match those of the sorting activity and thus, those of our real data. These images are encoded the same way we encoded our real depth images, and at the same resolution. Figure 3 shows some examples of depth images from our synthetic dataset, consisting in around 200,000 images.

Fig. 3. Examples of synthetic images created using Blender. To match our use case, these images represents only the upper body of the user, in a very variable set of poses.

However, we will see later (see Section 6) that training our pose estimator directly with this synthetic data only does not give satisfactory results.

Thus, it is needed to transform this data for it to better match our real data. For this task, we chose to use CycleGAN [38].

4 Image-to-image translation

CycleGAN [38] is an unsupervised approach for image-to-image translation which does not require paired data. The CycleGAN allows to find a mapping between the distributions of the two sets of data. In the following, we present the architecture described by figure 4 with training details and provide first results on the synthetic dataset.

[3] https://www.blender.org
[4] http://www.makehumancommunity.org

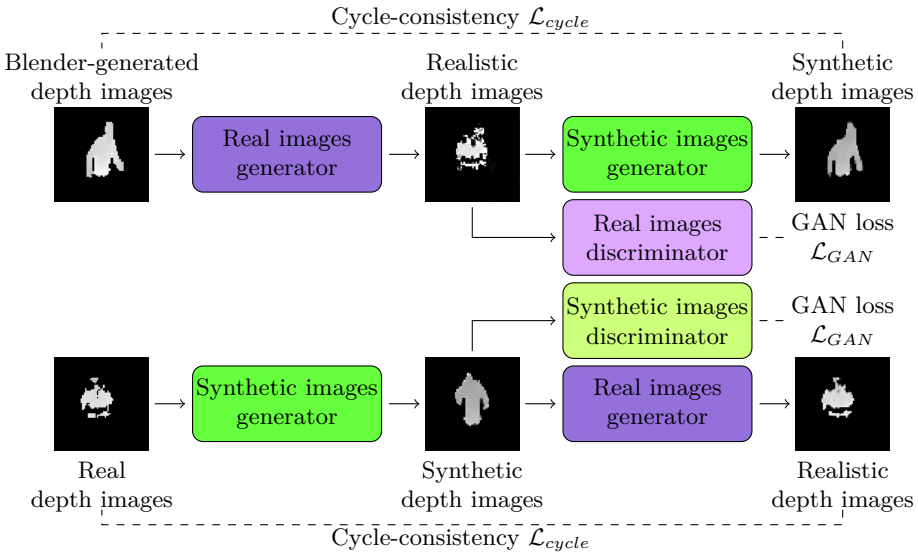

Fig. 4. Overview of the image-to-image translation training process.

4.1 Model architecture

The CycleGAN is composed of two generative networks and two discriminators. As we need lightweights architectures to achieve real-time performance in our industrial use-case, and as the operator is located at the center of the image and only occupies a small part of the depth image, we crop the depth images to only keep the operator and resize it into 64×64×1 images.

After several tests about the architecture of the generators, the final architecture of both generators is the same and consists of three parts:

- a first descending part composed of three convolutional layers with stride to reduce the spatial dimension of the images, with the number of filter doubling at each convolutional layer from 32 to 128;
- a second resnet part containing three residual blocks [14] of a single 2D convolutional layer of 128 filters;
- a third ascending part consisting of three deconvolutional layers with stride to retrieve the original image size, with the number of filters decreasing from 128 to 32.

All the convolutional and deconvolutional layers have a kernel size of 4×4, with zero-padding, and are followed by a ReLU activation function and an Instance Normalization layer [31]. The final layer is a 2D convolutional layer with a single filter, a kernel size of 4×4 and hyperbolic tangent activation function to obtain our final normalized translated image.

The discriminators are two 64×64 PatchGANs [16, 30], which are fully convolutional neural networks whose goal is to classify whether 64×64 overlapping

image patches are real or fake by outputting a probability map. More precisely, both discriminators contain a four convolutional layers with stride, an increasing number of filters from 32 to 256, a kernel size of 4×4, with zero-padding, and are followed by a LeakyReLU activation with $\alpha = 0.2$ function. They are all followed by an Instance Normalization layer, apart from the first one which is not normalized. A final convolutional layer with one filter outputs the patchs discrimination. These architectures are inspired from the architectures proposed by [38] for both generators and discriminators, and are given in Figure 5.

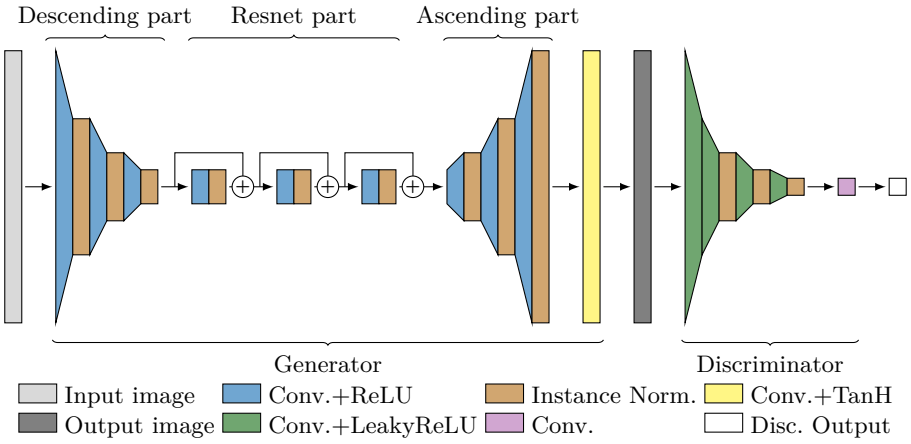

Fig. 5. The architectures of the generators and discriminators of our proposed CycleGAN.

4.2 Training details

To train this model, we use, as described in [38], three losses: two adversarial losses $\mathcal{L}_{GAN}$ and a cycle-consistency loss $\mathcal{L}_{cycle}$.

For two generators $G_R : S \to R$, $G_S : R \to S$ where R and S are the respective Real and Synthetic image datasets and their associated discriminators D_R and D_S, we express the adversarial losses as:

$$\mathcal{L}_{GAN}(G_R, D_R, S, R) = \mathbb{E}_{r \in R}[||D_R(r)||_2] + \mathbb{E}_{s \in S}[||1 - D_R(G_R(s))||_2] \quad (1)$$

where G_R aims to generate images that are similar to synthetic images from S and D_R tries to distinguish real samples r from generated samples, and:

$$\mathcal{L}_{GAN}(G_S, D_S, R, S) = \mathbb{E}_{s \in S}[||D_S(s)||_2] + \mathbb{E}_{r \in R}[||1 - D_S(G_S(r))||_2] \quad (2)$$

where G_S aims to generated images that are similar to real images from R and D_S tries to distinguish synthetic samples s from generated samples. We use an

L^2 loss function rather than a negative log-likelihood because this loss is more stable during training and generates higher quality results [21].

In the meantime, the goal of the cycle-consistency loss is to assure that $G_S(G_R(s)) \approx s$, i.e that a translation cycle should bring s back to the original synthetic image. The same goes with real images, where we want to assure that $G_R(G_S(r)) \approx r$. This cycle-consistency loss is then expressed as:

$$\mathcal{L}_{cycle}(G_S, G_R) = \mathbb{E}_{s \in S}[||G_S(G_R(s)) - s||_1] + \mathbb{E}_{r \in R}[||G_R(G_S(r)) - r||_1] \quad (3)$$

Finally, the global loss to train our model is:

$$\begin{aligned} \mathcal{L}(G_S, G_R, D_S, D_R) = & \mathcal{L}_{GAN}(G_R, D_R, S, R) \\ & + \mathcal{L}_{GAN}(G_S, D_S, R, S) \\ & + \lambda \mathcal{L}_{cycle}(G_S, G_R) \end{aligned} \quad (4)$$

where λ controls the relative importance of the two kind of loss.

We tried several differents values of λ before sticking to 10 as advised in the original CycleGAN paper [38]. We train the networks using the Adam optimizer [18] with a learning rate of 10^{-6}.

4.3 Preliminary results

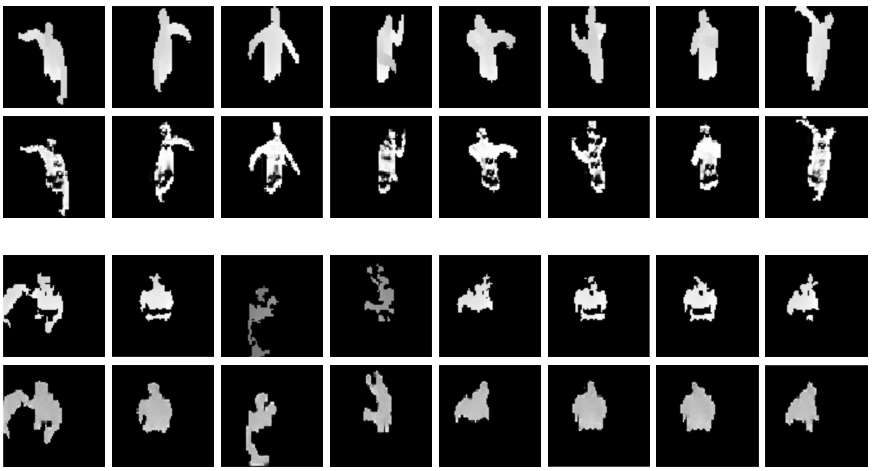

Fig. 6. Examples of image-to-image translation performed by our CycleGAN. We can see that the pose of the user is preserved during both translations. First row: synthese images from our synthetic dataset. Second row: translated realistic images associated to the images from the first row. Third row: real images from our real dataset. Last row: translated synthetic images associated to the images from the third row.

Some results are shown in Figure 6. We see that the real images generator manages to realistically degrade the synthetic images by adding noise on the

human operator's body and removing parts, similarly to a real image. On the other hand, the synthetic images generator seems to fill holes and put together parts from real images to make them look more "synthetic". However, and as expected, both generators neither add nor remove objects around the user.

5 Pose estimation on real images

We perform 3D pose estimation on single depth images using a deep convolutional neural network. Since our real datasets do not contain 3D human pose annotations, we can not directly train our network on this data. However, thanks to our synthetic dataset and the previously trained CycleGAN, we can generate annotated realistic depth images. This process may ideally yield an unlimited set of realistic depth images paired to the 3D human pose annotation corresponding to the synthetic images given as input. We use these couples between realistic depth images and synthetic 3D human pose annotations to train our pose estimation network.

The 3D pose is defined by the following fifteen upper body joints: *BaseSpine*, *MiddleSpine*, *TopSpine*, *Neck*, *Head*, *LeftShoulder*, *LeftElbow*, *LeftWrist*, *LeftHand*, *RightShoulder*, *RightElbow*, *RightWrist*, *RightHand*, *LeftHip* and *RightHip*. These joints are illustrated in Figure 7.

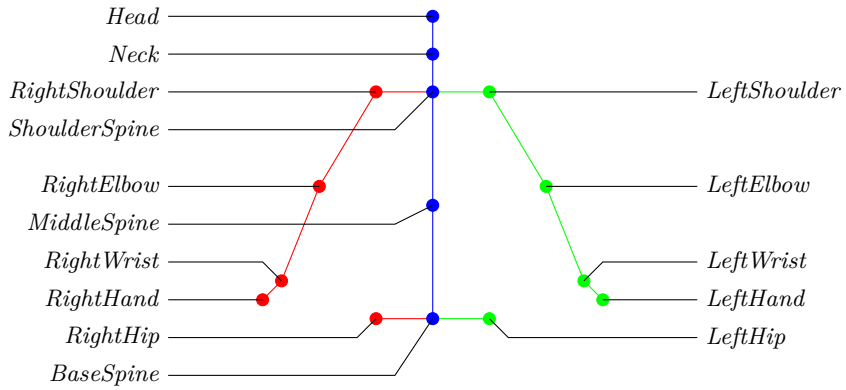

Fig. 7. The fifteen joints we used to define our 3D human pose.

The 3D pose estimation network is a convolutional neural network. Its inputs are 64×64 depth images, and it outputs a 45-dimensional vector representing the 3D position of the fifteen body joints. It contains three blocks of three 2D convolutional layers of 64 4×4 filters, with batch normalization and ReLU activation function. The blocks are separated by max-pooling layers. The final layer is a dense layer of 15×3 neurons, to estimate the 3D pose of the fifteen joints. An illustration of this architecture is given in Figure 8.

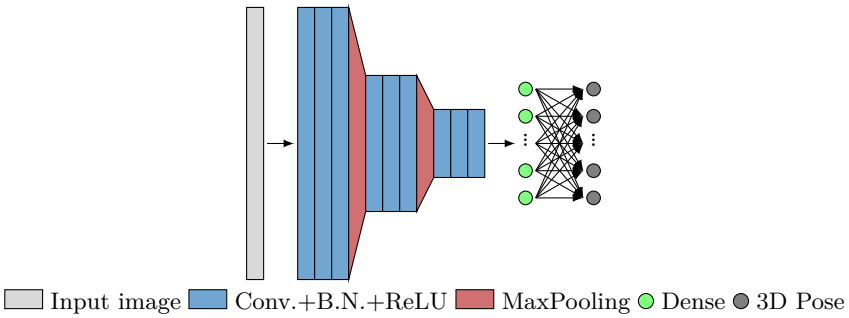

Fig. 8. The architecture of our pose estimation neural network.

To train this model, we use a mean squared error loss function. We use the Adam optimizer [18], with a learning rate of 10^{-3} and mini-batchs of 32 images.

Examples of our results on pose estimation are depicted in Figure 9. We can see that our results look consistent with the depth images, however it is highly dependent on the variability of the synthetic dataset and of the quality of the user segmentation in the depth images. Indeed, our synthetic dataset lacks of synthetic images with the arm up in the air and, as shown in the third row of Figure 9, the wrongly estimated pose often display this kind of error on flawed segmented frames.

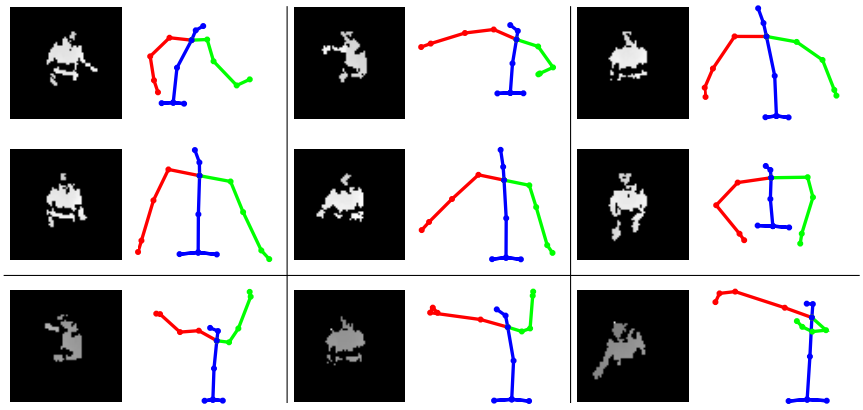

Fig. 9. Examples of real depth images and their associated 3D pose estimation. First two rows: the poses seem consistent with the depth image of the user. Third row: example of failure cases of our approach. After analysis, these errors often occurs when there are flaws in the segmentation, for instance pixels of the background breaking the depth normalization of the image (that we can see here with a different shade of gray in the image of the third row).

6 Experiments

We compare the different results thanks to various commonly used measures [35] such as Percentage of Correct Key-Points (PCK) at 150mm (PCK@150mm) and 80mm (PCK@80mm), that is the percentage of joints whose distance between predicted and true position lies within a given threshold, the Mean Per Joint Position Error (MPJPE), which is the mean euclidean distance between ground truths and predictions, and the Procrustes Analysis Mean Per Joint Position Error (PAMPJPE), which is a MPJPE where a similarity transformation is applied before computing the euclidean distance. For the PCK measures, the higher is the better, while for MPJPE and PAMPJPE, given in millimeters, the lower is the better.

In order to be able to measure the quality of our results and since we do not have annotations on real data, we created a new degraded dataset by applying a combination of white noise and image parts removal to build the *noisy and degraded* dataset, consisting of 200,000 640×480 synthetic images. We perform an ablation study on this second dataset with three kinds of training : a supervised training where the model is trained on the same kind of data as testing data, a "CycleGAN" training where the model is trained the outputs of the CycleGAN and an unsupervised training where the model is directly trained on the undegraded synthetic data, and tested on the *noisy and degraded* dataset. This ablation study proves that the image-to-image translation block is useful in order to improve the quality of the pose estimation.

As shown in Table 1, using the outputs of our image-to-image translation block clearly improves the quality of the pose estimations compared to directly training the same network on the synthese depth images, even though its performances are still worse than the supervised setting: the PCK@150mm and PCK@80mm improves respectively from 55.0% to 93.3% and 0.2% to 46.0%, while the MPJPE and PAMPJPE lower from 148.0 to 84.3 (43% reduction) and 87.6 to 65.9 (25% decrease).

Table 1. Results of our pose estimation network under various training conditions. We observe that using a CycleGAN improves the results compared to the unsupervised setting, despite being worse than the supervised learning.

Kind of data	Training type	PCK@150mm	PCK@80mm	MPJPE 3D	PAMPJPE 3D
Synthetic	Supervised	100.0	100.0	10.9	9.9
Noisy	Supervised	100.0	100.0	21.4	18.8
&	CycleGAN	93.3	46.0	84.3	65.9
Degraded	Unsupervised	55.0	0.2	148.0	87.6

It shows that our CycleGAN is able to correctly capture some features of degraded depth images and translate them on the synthetic images. It is also important to notice that the real difficulty of our dataset lies in the degradation

induced by the industrial context (i.e the holes in the depth image) and not in the imprecision of the sensor (simulated by the white noise).

To be able to perform more measures about the quality of our results, we acquired another real dataset in a controlled setting, simulating a simplified industrial context. These images were acquired with a Kinect V2 sensor in a bird's eye view of the user, allowing us to compare our pose estimation to the pose estimation provided by the sensors. This dataset contains around 13,000 depth frames of several subjects in a wide variety of human poses. However, comparing to this data is difficult for several reasons:

- our model joints are not exactly located at the same place on the human body than those of the Kinect;
- we estimate a 3D human pose, while Kinect V2 only estimates a 2D+Z human pose: the two first coordinates (X,Y) are estimated as pixel coordinates in the depth frame and then translated to the real world space thanks to the camera intrinsic parameters, but the third one is computed by adding a given offset to the depth value of the pixel, which can be far away from the real position of the joint when the sensor is not in front of the user;
- the Kinect V2 pose estimation is calibrated for an entertainment setting where the sensor is in front of the user. In our setting, the sensor has a bird's-eye view on the user, which leads to further deformations of the estimated pose.

An illustration of these issues is given in Figure 10. In particular, we can see that the Kinect pose estimation is strongly tilted forwards, while ours stays straight, which is closer to the real position of the user.

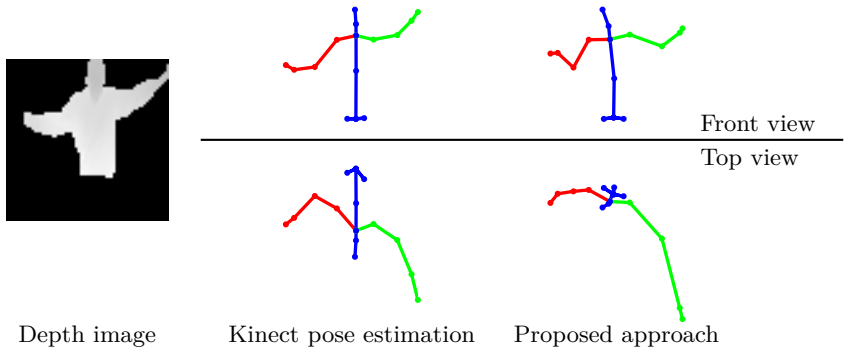

Fig. 10. Comparison of the kinect pose estimation against the proposed pose estimation. Left: the segmented depth frame. Center: the 2D+Z pose estimation provided by the Kinect sensor. Right: the proposed pose estimation.

However, despite not being able to compare our results to a real ground truth, we confirm the results that we showed on synthetic data (see Table 2).

Table 2. Results of our pose estimation compared to the pose estimation provided by the Kinect V2 sensor.

Training type	MPJPE 3D	PAMPJPE 3D
CycleGAN	**268.7**	**131.6**
Unsupervised	442.1	199.6

Due to the facts given previously, there is a large gap between the Kinect pose estimation and ours, but the use of the CycleGAN greatly reduces this gap and thus improves the quality of our network's pose estimation. The CycleGAN improves the quality of the pose estimations, by reducing both the MPJPE and PAMPJE, respectively by 39% and 34%.

7 Conclusion

In this paper, we propose an unsupervised framework to provide 3D pose estimation on single depth images to prevent musculoskeletal disorders in an industrial use-case. The approach relies on a synthetic dataset composed of synthetic depth images. Using an unsupervised image-to-image translation CycleGAN preserving geometric features, we refine these images to render realistic depth images. These images, coupled to the body joints annotations from the input synthetic images, allow us to efficiently train a simple 3D pose estimation network. Several experiments performed both on degraded synthetic data and on real data acquired in a simulated industrial context proves the efficiency of this image-to-image translation process to perform unsupervised pose estimation.

In future work, we could exploit temporal information to improve the results of both our CycleGAN and pose estimator. We could also introduce more complicated models.

References

1. Aleksynska, M., Berg, J., Foden, D., Johnston, H.E.S., Parent-Thirion, A., Vanderleyden, J.: Working Conditions in a Global Perspective. Publications Office of the European Union (2019)
2. Alp Güler, R., Neverova, N., Kokkinos, I.: Densepose: Dense human pose estimation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7297–7306 (2018)
3. Aytar, Y., Castrejon, L., Vondrick, C., Pirsiavash, H., Torralba, A.: Cross-modal scene networks. IEEE transactions on pattern analysis and machine intelligence **40**(10), 2303–2314 (2017)
4. Banerjee, T., Rantz, M., Li, M., Popescu, M., Stone, E., Skubic, M., Scott, S.: Monitoring hospital rooms for safety using depth images. AI for Gerontechnology (2012)
5. Blanc-Beyne, T., Carlier, A., Charvillat, V.: Iterative dataset filtering for weakly supervised segmentation of depth images. In: 2019 IEEE International Conference on Image Processing. pp. 1515–1519. IEEE (2019)

6. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In: European Conference on Computer Vision. pp. 561–578. Springer (2016)
7. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7291–7299 (2017)
8. Chan, C., Ginosar, S., Zhou, T., Efros, A.A.: Everybody dance now. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5933–5942 (2019)
9. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: The IEEE Conference on Computer Vision and Pattern Recognition (June 2018)
10. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: Rmpe: Regional multi-person pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2334–2343 (2017)
11. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Scene parsing with multiscale feature learning, purity trees, and optimal covers. arXiv preprint arXiv:1202.2160 (2012)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
13. Haque, A., Peng, B., Luo, Z., Alahi, A., Yeung, S., Fei-Fei, L.: Towards viewpoint invariant 3d human pose estimation. In: European Conference on Computer Vision. pp. 160–177. Springer (2016)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence **36**(7), 1325–1339 (2013)
16. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
17. Jiu, M., Wolf, C., Taylor, G., Baskurt, A.: Human body part estimation from depth images via spatially-constrained deep learning. Pattern Recognition Letters **50**, 122–129 (2014)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6936–6945 (2019)
20. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Advances in neural information processing systems. pp. 700–708 (2017)
21. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z.: Multi-class generative adversarial networks with the l2 loss function. arXiv preprint arXiv:1611.04076 **5** (2016)
22. Marín-Jiménez, M.J., Romero-Ramirez, F.J., Muñoz-Salinas, R., Medina-Carnicer, R.: 3d human pose estimation from depth maps using a deep combination of poses. Journal of Visual Communication and Image Representation **55**, 627–639 (2018)
23. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)

24. Moon, G., Yong Chang, J., Mu Lee, K.: V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5079–5088 (2018)
25. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European conference on computer vision. pp. 483–499. Springer (2016)
26. Punnett, L., Wegman, D.H.: Work-related musculoskeletal disorders: the epidemiologic evidence and the debate. Journal of electromyography and kinesiology **14**(1), 13–23 (2004)
27. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: International Conference on Learning Representations (05 2016)
28. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in neural information processing systems. pp. 2234–2242 (2016)
29. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1297–1304. IEEE (2011)
30. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2107–2116 (2017)
31. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
32. Wang, K., Zhai, S., Cheng, H., Liang, X., Lin, L.: Human pose estimation from depth images via inference embedded multi-task learning. In: Proceedings of the 24th ACM international conference on Multimedia. pp. 1227–1236. ACM (2016)
33. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)
34. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5907–5915 (2017)
35. Zhang, X., Wong, Y., Kankanhalli, M.S., Geng, W.: Unsupervised domain adaptation for 3d human pose estimation. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 926–934. ACM (2019)
36. Zhou, X., Sun, X., Zhang, W., Liang, S., Wei, Y.: Deep kinematic pose regression. In: European Conference on Computer Vision. pp. 186–201. Springer (2016)
37. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: European Conference on Computer Vision. pp. 597–613. Springer (2016)
38. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: The IEEE International Conference on Computer Vision (October 2017)
39. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: Advances in Neural Information Processing Systems. pp. 465–476 (2017)