LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images

Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid, *Fellow, IEEE*

Abstract—We propose an end-to-end architecture for joint 2D and 3D human pose estimation in natural images. Key to our approach is the generation and scoring of a number of pose proposals per image, which allows us to predict 2D and 3D poses of multiple people simultaneously. Hence, our approach does not require an approximate localization of the humans for initialization. Our Localization-Classification-Regression architecture, named LCR-Net, contains 3 main components: 1) the pose proposal generator that suggests candidate poses at different locations in the image; 2) a classifier that scores the different pose proposals; and 3) a regressor that refines pose proposals both in 2D and 3D. All three stages share the convolutional feature layers and are trained jointly. The final pose estimation is obtained by integrating over neighboring pose hypotheses, which is shown to improve over a standard non maximum suppression algorithm. Our method recovers full-body 2D and 3D poses, hallucinating plausible body parts when the persons are partially occluded or truncated by the image boundary. Our approach significantly outperforms the state of the art in 3D pose estimation on Human3.6M, a controlled environment. Moreover, it shows promising results on real images for both single and multi-person subsets of the MPII 2D pose benchmark and demonstrates satisfying 3D pose results even for multi-person images.

Index Terms—Human 3D pose estimation, 2D pose estimation, detection, localization, classification, regression, CNN

1 Introduction

State-of-the-art methods for 2D human pose estimation in real images obtain excellent performance using Convolutional Neural Network (CNN) architectures [1], [2], [3]. However, occlusion still remains a significant challenge as analyzed in [3]. One way to recover body part locations in cases of occlusions is to reason about the full-body 3D pose. Methods for 3D human pose understanding require training data that is only available through Motion Capture (MoCap) systems [4], [5], [6]. Even if they show accurate pose estimation results (including occluded joints) in controlled environments, these approaches do not generalize well to real images, with the exception of recent work based on data synthesis that shows promising results in the wild [7], [8]. In this paper, we propose a method that results in multiple full-body 2D and 3D pose hypotheses in different regions of the image. These *pose proposals* are efficiently sampled, scored and refined using an end-to-end CNN architecture inspired by the latest work on object detection [9]. Finally, the pose proposals are combined to estimate both the location and the 2D-3D pose of the individuals present in the observed scene. Our method recovers full-body poses, even when the persons are partially occluded or truncated by the image boundary as illustrated in the examples presented in Figure 1.

CNNs have been used for full-body pose estimation both in regression [7], [10], [11], [12], [13] and classification [8] approaches. Regression networks are trained to directly estimate the 2D or 3D location of the body joints, whereas a classification approach defines pose classes and returns the average pose of the top scoring class. Increasing the number of clusters improves precision of the estimation in classification approaches

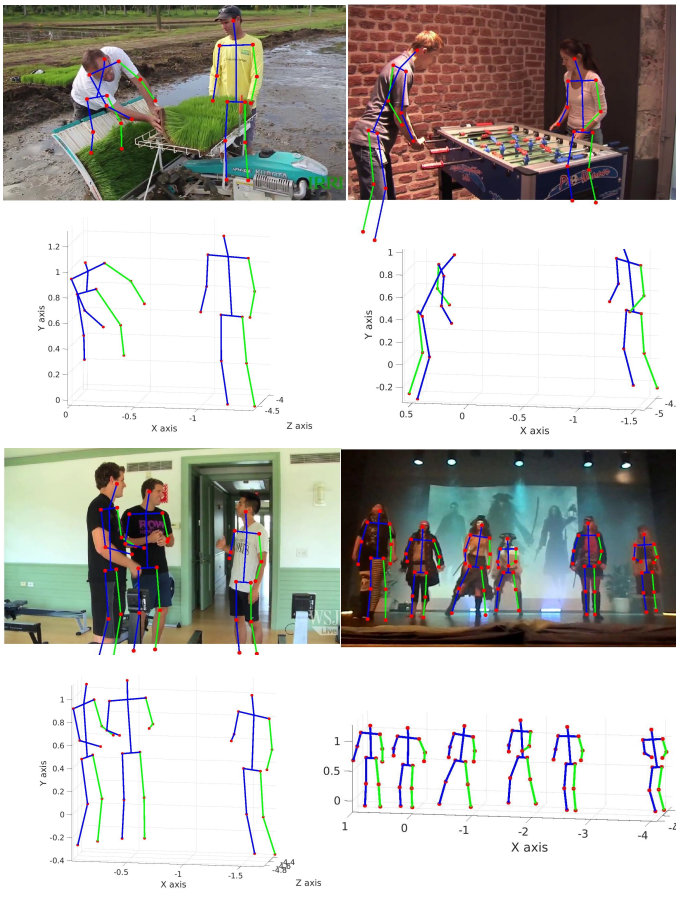


Fig. 1. Examples of multi-person 2D-3D pose detections in natural images. For each image, we show the 2D and 3D poses that are estimated jointly, even in cases of occlusions or truncations, by reasoning in terms of full-body 2D-3D pose.

- *G. Rogez and C. Schmid are with the THOTH team, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, France.*
  E-mail: firstname.lastname@inria.fr
- *P. Weinzaepfel is with NAVER LABS Europe, Meylan, France.*
  E-mail: philippe.weinzaepfel@naverlabs.com

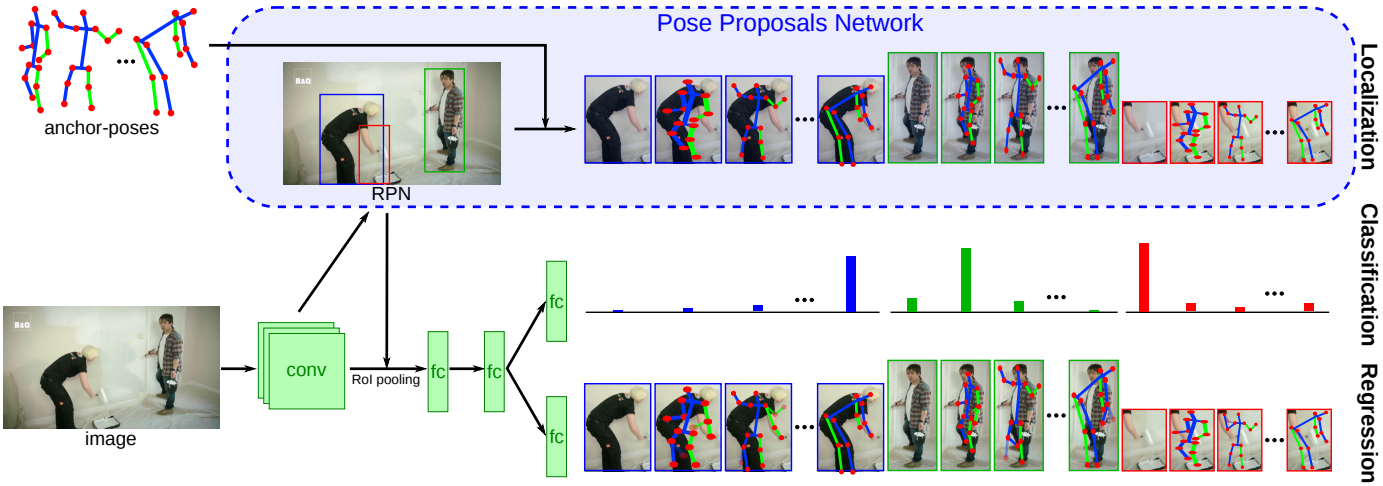


Fig. 2. Overview of our LCR-Net architecture (poses only shown in 2D for better readability). We first extract candidate regions using a Region Proposal Network (RPN) and obtain pose proposals by placing a fixed set of anchor-poses into these boxes (top). These pose proposals are then scored by a classification branch and refined using class-specific regressors, learned independently for each anchor-pose.

but makes discrimination harder. Regression methods can only predict one pose for a given image and fail to model multi-modal outputs, e.g., for ambiguous cases. In this paper, we argue that for full-body human pose estimation, the discriminative power of classification networks can be combined with the smoothness of regression methods by a simple yet elegant modification within the learning procedure. The architecture is similar in spirit to Faster R-CNN [9] which jointly localizes and classifies objects while regressing a refined bounding box. This is achieved using a Region Proposal Network (RPN) that generates high-quality region proposals where object bounds and objectness scores are predicted. Instead of classifying objects, we propose to classify human poses. The key idea of our approach is to quantify the space of valid full-body poses and jointly train a K-way classifier on this partitioned space as well as local pose regression models, e.g. one per pose cluster. To this end, we formulate a joint classification-regression loss function that combines coarse pose classification and class-specific pose regression. Given a set of K hypothetical pose classes, we output for each proposed image region a list of K refined 2D-3D poses and the associated classification scores.

In summary, we propose an end-to-end Localization-Classification-Regression architecture, named LCR-Net, that detects 2D and 3D poses in natural images, see Figure 2. The network proceeds by extracting candidate regions for the person localization. We obtain *pose proposals* by locating the set of K hypothetical pose classes, denoted as anchor-poses, in these candidate boxes. Each pose proposal is then scored using a classification branch and regressed independently for each anchor-pose. The localization, *i.e.*, extraction of the pose proposals, classification and per anchor-pose regression, share layers and can be trained end-to-end. Our final output consists in a number of 2D-3D poses per images that are obtained by aggregating similar pose proposals, in terms of 2D location and 3D pose. To the best of our knowledge, our work is the first to tackle multi-person 3D pose estimation from a single image. The work presented in this paper is an extension of [14]. We analyze three ways to considerably boost the 2D-3D pose estimation performance of our LCR-Net architecture: (1) the use of additional synthetic data to augment the size of the training data sets, (2) a variant of the architecture with an iterative process as in [1], [3] that further refines regression and classification results and (3) an improved alignment of the candidate regions of interest similar to the recent [15] that better conserves the spatial details by avoiding rounding operations. Altogether, our complete method, called LCR-Net++, significantly improves over the initial version, achieving a boost in performance of more than 20mm in 3D and 5% in 2D. Our approach outperforms the state of the art for 3D pose estimation in a controlled environment, even when compared to methods that leverage temporal smoothing or rely on initial localization of the human and show promising results in real images, estimating the poses both in 2D and 3D.

This paper is organized as follows. After reviewing the related work in Section 2, Section 3 introduces LCR-Net. Extensive experimental results, both in 2D and 3D, are presented in Section 4.

2 Related Work

In this section, we review related work for 2D (Section 2.1) and 3D (Section 2.2) human pose estimation from single images.

2.1 Human localization and 2D pose estimation

Most state-of-the-art approaches for 2D human pose estimation employ CNN architectures [1], [2], [3], [13], [16], [17], [18], [19]. They can be divided into two groups: (a) methods which first search the image for local body parts and model their dependencies using graphical models [1], [16], [19] and (b) holistic approaches that directly estimate the full body [13], [17].

Methods based on local body parts require a tight bounding box around each human to estimate his pose [3], [20], [21], others can detect multiple people in natural images at once [1]. Most methods extract joint heatmaps, *i.e.*, probabilistic maps that estimate the probability of each pixel to contain a particular joint. An iterative procedure is often used [1], [3], [20]: a refined estimate of the heatmaps is obtained from the previous estimate and the convolutional features. Joint positions can be estimated by taking the local maxima of the heatmaps. In Convolutional Pose Machines, Wei *et al.* [20] refine the predictions over successive stages with intermediate supervision at each stage. In the Stacked Hourglass network [3], repeated bottom-up, top-down processing

used in conjunction with intermediate supervision improve the performance of the network.

Papandreou *et al*. [21] also compute a per-joint regressor at each pixel to refine the position of the joints, that may lack precision due to the stride of CNNs. Given the joint positions extracted from the heatmaps, additional post-processing is required to build human poses, such as graph partitioning [22]. Cao *et al*. [1] proposed an alternative approach by also regressing affinities between joints, *i.e*., the direction of the bones, together with the heatmaps. In contrast to these methods that build human poses from local body parts, our method extract full-body 2D and 3D poses, even in case of occlusions.

Holistic approaches often assume that the individuals have been localized, and that a bounding box around each person is available. Toshev and Szegedy [13] directly regress the positions for each joint using an iterative procedure. Fan *et al*. [17] combines the local appearance with an holistic view of the body to estimate the position of the joints. Instead of relying on a multi-stage approach, our network is trained in an end-to-end fashion and outputs both 2D and 3D poses jointly.

2.2 3D human pose from a single image

Methods for 3D human pose estimation from a single image can be decomposed into two groups: (a) the ones that first compute 2D poses and use them to estimate 3D poses and (b) approaches that directly learn mappings from image features to 3D poses.

Motivated by the recent advances in 2D pose detection, a large body of work tackles 3D pose estimation from 2D poses assuming that the 2D joints are available [23], [24], provided by an off-the-shelf 2D pose detector [25], [26], [27], [28], [29], [30], [31], [32] or obtained through a 2D pose estimation module within the proposed architecture [33], [34]. Most of these methods reason about geometry. Chen and Ramanan [29] estimate 3D pose from 2D through a simple nearest neighbor search on a given 3D pose library with a large number of 2D projections. Moreno-Noguer [30] formulates the problem as a 2D-to-3D distance matrix regression. Nie *et al*. [31] predict the depth of human joints based on their 2D locations using LSTM, whereas Martinez *et al*. [32] lift 2D joints to 3D space using a simple, fast and lightweight deep neural network. These methods remain limited by the performance of the 2D pose estimator.

Some other approaches directly estimate the 3D pose from image features [35], [36], [37], [38], [39]. Recently, this has been naturally extended to end-to-end mappings using CNN architectures, either in monocular images [7], [8], [12], [40], [41], [42] or in videos [11], [43]. Pavlakos *et al*. [42] propose a volumetric representation for 3D human pose and employ a ConvNet to predict per-voxel likelihoods for each joint. In [44], a structure-aware regression approach is followed with a reparameterized pose representation using bones instead of joints.

Finally, some recent approaches treat 2D and 3D pose estimation jointly or iteratively [44], [45], [46], [47], [48]. In [47], the authors learn how to fuse 2D and 3D image cues while in [48] a multi-stage CNN architecture leverages the knowledge of plausible 3D landmark locations to refine the search for better 2D locations. Most similar to our approach is the classifier of [8] that outputs a distribution of scores over a quantized set of 2D-3D poses. We also use a classifier where each class corresponds to a particular 2D-3D orientated pose but we combine classification and regression in an effective architecture that refines the pose using a class-specific regression stage. Importantly, the method of [8] requires a well-aligned bounding box around the subject while we jointly localize and estimate 2D and 3D pose of multiple people in real-world images.

Large-scale training data is necessary to train accurate state-of-the-art CNN architectures for pose estimation. While 2D pose data are obtained by manually annotating images captured in-the-wild, reliable 3D poses are acquired using motion capture (MoCap) systems in constrained environment. As a consequence, many methods for 3D pose estimation are trained and evaluated in these controlled and unrealistic scenarios [4], [5] and do not generalize well to real-world images. Some architectures have been proposed to take advantage of the different sources of training data, *i.e*., indoor images with MoCap 3D poses and real-world images with 2D annotations [49], [50]. To generalize to in-the-wild images, Mehta *et al*. [49] proposed a 2D-to-3D knowledge transfer, *i.e*., using pre-trained 2D pose networks to initialize the 3D pose regression networks while in [50] the common representations between the 2D and the 3D tasks are shared. To compensate for the lack of large scale in-the-wild datasets, recent work has also proposed to generate training images for particular 3D pose datasets such as the CMU MoCap dataset [6] by stitching image regions [8], animating human 3D models [7], [51], using a game engine [52] or by rendering textured 3D body scans [53], [54]. These synthetic datasets have proved to be useful for training CNN architectures, yet often requiring a domain adaptation stage. However, none is realistic enough in terms of clothing, hair or interactions with objects to be considered as a fully-convincing alternative to real images. Recently, Lassner *et al*. [55] proposed a self-improving, scalable method that obtains high-quality 3D body model fits for 2D images. We also generate "pseudo" ground-truth 3D pose annotations for real-world images following a simple yet effective method that leverages 2D pose annotations to 3D using large-scale motion capture data.

3 LCR-NET

We propose to detect human poses using a Localization-Classification-Regression Network (LCR-Net). In this paper, a human pose (p, P) is defined as the 2D pose p, *i.e*., the pixel coordinates of each joint in the image; and the 3D pose P, *i.e*., 3D location of each joint relative to the body center (in meters). We consider poses with 13 joints. We assume that a fixed set of K 2D-3D anchor-poses is given, denoted by $\{(a_k, A_k)\}_{k=1..K}$. In this paper, they are obtained by clustering a large set of poses and using the center of each cluster as anchor pose, see Section 4 for details.

Figure 2 shows an overview of our LCR-Net architecture. Given an image, we first compute convolutional features. The *Localization* component, also called Pose Proposals Network in the context of pose detection, outputs a list of pose proposals. Pose proposals consist of a set of candidate locations where the anchor-poses are hypothesized. Next, a Region-of-Interest (RoI) pooling layer aggregates the features inside each candidate region. After two fully-connected layers, the network is split into two components. The *Classification* branch estimates the probability of anchor-poses to be correct at each location. It thus jointly learns to localize humans, as well as to estimate which anchor-pose is more probable. The *Regression* branch computes an anchor-pose-specific regression that estimates the difference between the true

human pose and the pose proposal (Figure 3). Our loss is the sum of three losses that we describe in more details in the following:

$$\mathcal{L} = \mathcal{L}_{Loc} + \mathcal{L}_{Classif} + \mathcal{L}_{Reg} \quad . \tag{1}$$

Note that the convolutional features are shared between the three components and that the classification and regression branches also share features from two fully-connected layers. The architecture allows end-to-end training for localizing humans and estimating their 2D-3D poses, in contrast to most previous works which run a human detector before estimating the pose.

3.1 Localization: pose proposals network

The Pose Proposal Network outputs a set of N×K pose proposals, *i.e.*, 2D-3D pose hypotheses obtained by placing the K anchor-poses in the N bounding boxes generated by the RPN [9]. These pose proposals will be scored and refined by the classification and regression branches respectively, see Figure 2. The loss of the localization component is the loss of the RPN network:

$$\mathcal{L}_{Loc} = \mathcal{L}_{RPN} \quad . \tag{2}$$

During training, each bounding box B is labeled with a ground-truth class $c_B \in \{0 \dots K\}$ and a pose regression target t_{c_B}. The ground-truth class c_B is set to 0 (corresponding to background) if the bounding box has an Intersection over Union (IoU) below 0.5 with all ground-truth poses. The IoU between a box and a pose is computed using the bounding box around all joints of the pose, with a fixed additional margin of 10%. If B has a high overlap with several poses, let (p, P) be the ground-truth pose with the highest IoU with the box. The label c_B is set to $c_B = \text{argmin}_k \, D_{3D}(A_k, P)$ where $D_{3D}(.,.)$ is the distance between oriented 3D poses centered at the torso. This label will be used by the classification branch (Section 3.2). If the label c_B is non-zero, we also define a pose regression target, used in the regression branch (Section 3.3), t_{c_B} for the box B as $t_{c_B} = (\tilde{p} - \tilde{a}_{c_B}, P - A_{c_B})$ where $\tilde{p}$ and $\tilde{a}_{c_B}$ denote the 2D pose and anchor-pose normalized in the range $[0..1]$ according to the box coordinates (see arrows on Figure 3). This normalization makes the regression independent of scale and position of the person and the box in the image.

3.2 Classification

The classification component aims at predicting the closest anchor-pose, *i.e.*, the correct label, for each bounding box B. In other words, each bounding box is assigned a probability for each anchor-pose (and the background class). Let u be the probability distribution estimated by the network, obtained by three fully-connected layers after RoI pooling, see Figure 2, followed by a softmax. The classification loss is defined using the standard log loss of the true class:

$$\mathcal{L}_{Classif}(u, c_B) = -log \; u(c_B) \quad . \tag{3}$$

3.3 Regression

The regression component aims at refining the coarse anchor-poses located in the region proposals as depicted in Figure 3. The specificity of our approach is that the regression is anchor-pose-specific and a regressor is learned independently for each anchor-pose. The regression outputs v are obtained by using a fully-connected layer after the two fully-connected layers shared with the classification branch (see Figure 2). The dimension of v is equal to $(K+1) \times 5 \times \#joints$, where the factor of 5 reflects the components of the 2D and 3D coordinates. We denote by v_{c_B} the subvector of v corresponding to the regression for anchor-pose c_B. The regression loss is defined as:

$$\mathcal{L}_{Reg}(v, t_{c_B}) = [c_B \geqslant 1] \;\; \|t_{c_B} - v_{c_B}\|_S \quad , \tag{4}$$

with $\|.\|_S$ the smooth-L1 loss, a robust version of the L2 loss which is less sensitive to outliers:

$$\|x\|_S = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{otherwise.} \end{cases} \tag{5}$$

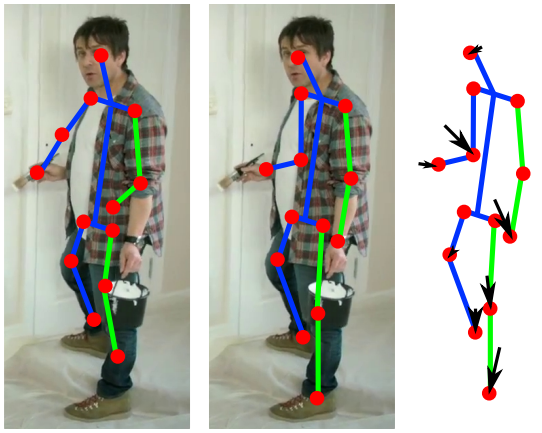


Fig. 3. The regression aims at refining the anchor-pose to match the ground-truth 2D-3D pose (only shown in 2D for better readability).

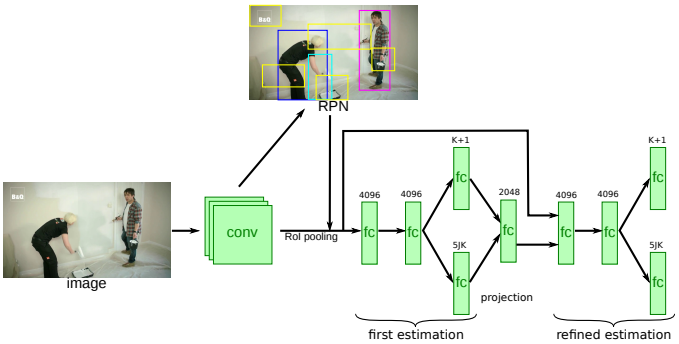


Fig. 4. Illustration of the iterative estimation procedure. The losses are applied on each estimate.

3.4 Iterative Estimation

We propose a variant of the architecture in which the regression and classification are iteratively estimated and refined. Such an iterative estimation is common in pose estimation [1], [3]. More precisely, we add several layers at the end of the LCR-Net networks, see Figure 4. A first estimate of the classification and regression is obtained using two fully-connected layers, that are shared between the two tasks, followed by a fully-connected layer for each task. The result of this first estimate is combined with the features pooled over the RoI to refine the estimation. In more details, the output of these first classification and regression are concatenated and fed to a fully-connected layer to obtain a fixed representation of 2048 dimensions, independently of K. We then concatenate this feature vector with the convolutional features

pooled over the RoI and feed it to a similar network architecture as done for the initial estimate: two fully-connected layers followed by an additional layer for classification and another one for regression. Losses are applied at every estimate during training, while the last estimation is taken at test time.

3.5 Implementation details

Similar to Faster R-CNN, we use an approximate joint training version, in which boxes are considered as fixed by the RoI pooling layer. We replace the RoI pooling layer by a RoI align layer similar to the recent Mask R-CNN [15]. In the traditional RoI pooling layer, the region of interest coordinates are first rounded according to the stride of the convolutional features, then split into a fixed number of cells for which the coordinates are also rounded, and a max-pooling operator is applied in each cell. In contrast, the RoI align layer is designed to conserve the spatial details as it avoids these rounding operations. The features for 4 regularly sampled points per cell are obtained by bilinear interpolations and a max-pooling operator is used in each cell. We use the same parameters as [9] for RPN. For the classification and regression loss, we use 256 boxes per batch, with 32 boxes coming from 8 different images, *i.e.*, from more images than in the standard version. We have more labels and, consequently, we need more diversity inside each batch. One quarter of the boxes are on humans, the remaining ones on background. The network is based on VGG-16 architecture [56] and the weights are initialized with ImageNet [57] pretraining.

Fig. 5. Illustration of the pose proposal integration (PPI). The pose proposals (a) are grouped based on 2D overlap and 3D pose to identify the persons and the modes (b). Final pose estimates (c) are obtained by averaging the 2D poses in the selected modes and thresholding.

3.6 Pose proposals integration

Our LCR-Net outputs a set of refined pose proposals with associated classification scores $s(p, P) = u(c_B)$ from Equation 3. To penalize pose proposals with one or several joints outside the bounding box B with respect to poses that are entirely inside the box, and consequently more likely to be accurately estimated, we propose to rescore the proposals using:

$$s' = s \, \frac{\sum_i f(p_i, B)}{\#joints} \ , \tag{6}$$

where function $f(p_i, B) = 1$ inside the box B and gradually decreases outside $f(p_i, B) = exp(-\mathfrak{D}^2(p_i, B)/\sigma_b^2)$, $\mathfrak{D}(p_i, B)$ being the Euclidean distance of joint j_i to the box B. In practice, σ_b is set to 25 pixels. If all the joints are inside B, then $s' = s$.

Multiple proposals cover each person present in the image. One possibility is to use a non-maximum suppression algorithm (NMS) and return the top scoring proposal for a given region as estimated pose. Instead, we propose to aggregate proposals which are close in terms of image location and 3D pose. We refer to this post processing stage as the pose proposal integration (PPI), see Figure 5.

We start with grouping pose proposals with a sufficient spatial overlap in the 2D image, *i.e.*, an IoU above a certain threshold for the bounding boxes around the 2D joints. We take the top scoring proposal in the image and determine all the pose proposals that overlap sufficiently with this top scoring proposal. We repeat this step with the remaining pose proposals and their top scoring elements until no pose proposals are left. The resulting groups are coherent in terms of spatial overlap but can consist of very different 3D poses and hence the modes in 3D pose space need to be identified. Let $\mathcal{P} = \{(p, P)\}$ be the set of pose proposals in a group, each one with a classification score $s(p, P)$. We first pick the proposal with the highest score, *i.e.*, $(p^*, P^*) = \text{argmax}_{(p,P)\in\mathcal{P}} \, s(p, P)$. We then select the set $\mathcal{Q}$ of pose proposals in the group $\mathcal{P}$, for which the 3D distance D_{3D} from P^* is below a threshold T_{3D}:

$$\mathcal{Q} = \left\{(p, P) \in \mathcal{P} \mid D_{3D}(P^*, P) < T_{3D}\right\} \ . \tag{7}$$

This selection ensures that we do not average poses that belong to different modes. The PPI is thus parameterized by 2D and 3D thresholds, *i.e.*, IoU and T_{3D} respectively.

We then obtain our final 2D pose p (and similarly the 3D pose) by averaging the 2D poses in mode $\mathcal{Q}$ weighted by their scores:

$$p = \frac{1}{S} \sum_{(q,Q)\in\mathcal{Q}} s(q, Q) \times q \ , \tag{8}$$

with S the sum of the individual scores, *i.e.*, $S = \sum_{(q,Q)\in\mathcal{Q}} s(q, Q)$. The score for this pose p is set to S, which results in a higher score for poses with multiple pose proposals. We iterate this process, starting from the highest scored pose among the ones that have not yet been covered by a mode.

3.7 Pseudo ground-truth 3D pose

To train our network, we need full-body 2D and 3D ground-truth poses associated with each training image. Existing datasets with images captured in-the-wild only provide 2D joint locations of the visible joints. Inspired by Iqbal *et al.* [33] who use 2D poses to retrieve the normalized nearest 3D poses from a motion capture dataset, we propose to infer ground-truth 3D poses from 2D annotations using a nearest neighbor (NN) search performed on the annotated joints. A similar method was recently followed in [29] to estimate 3D pose from 2D joints locations.

A large corpus of MoCap 3D poses is first projected orthographically on multiple random virtual views to generate a very large set of 2D poses and associated orientated 3D poses $\mathcal{M} = \{(p_m, P_m)\}_m$. Next, given an annotated 2D pose p, a search is performed with the normalized pose $\bar{p} = p/||p||$ to estimate the closest match, *i.e.*, the 3D pose and camera view within $\mathcal{M}$ for which the 2D distance D_{2D} is smallest:

$$(p_m^*, P_m^*) = \underset{(p_m,P_m)\in\mathcal{M}}{\text{argmin}} \ D_{2D}(\bar{p}, \bar{p}_m) \ . \tag{9}$$

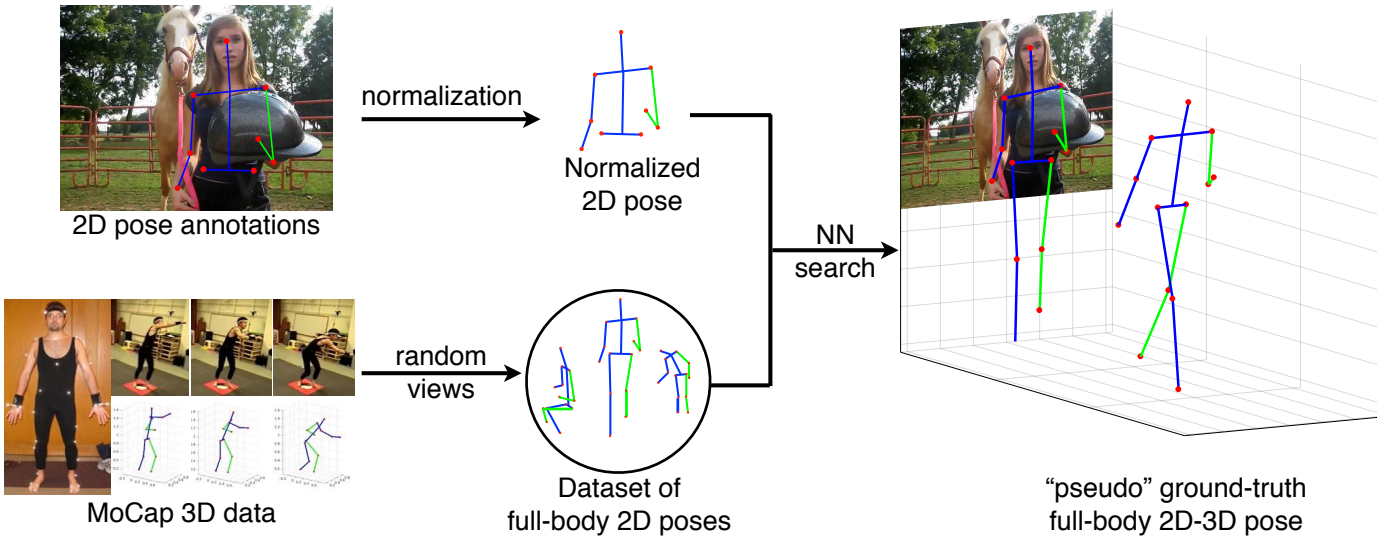


Fig. 6. Pseudo ground-truth full-body 2D-3D pose annotation. From left to right: given an image with a manual 2D annotations, the pose is first normalized, then it is compared against a dataset of full-body 2D poses. These 2D poses are obtained by projecting a large corpus of MoCap 3D poses on multiple random views and normalizing them with respect to the annotated joints only. The closest pose is recovered and used (a) to define a “pseudo” ground-truth full-body 3D pose and (b) to complete missing annotations of the 2D pose.

The 3D pose of the closest match P^*_m is then considered as “pseudo” ground-truth of the query 2D pose p. In practice, when humans are truncated or partially occluded, some joints of the 2D pose can be missing. In such cases, the normalized poses $\bar{p}$ and $\bar{p}_m$ are computed using the annotated joints only. The recovered 2D pose p^*_m is then employed to complete missing 2D annotations so that each training instance is associated with full-body 2D and 3D annotations. See example in Figure 6.

4 Experimental Results

In this paper, we address joint 2D and 3D human pose detection in natural images. To the best of our knowledge, there exists no dataset with 3D annotations for real-world images. To evaluate our method, we thus perform separate experiments on (a) 3D pose estimation in a controlled environment, *i.e.*, on the Human3.6M dataset [5] (Section 4.1), and (b) 2D and 3D pose estimation in natural images on the MPII human pose dataset [58] (Section 4.2).

4.1 3D pose detection on Human3.6M

Dataset and evaluation protocols. The Human3.6M dataset [5] contains 3.6M human poses from 11 actors performing 17 different scripted actions. The videos are captured in a controlled environment from 4 different camera viewpoints while accurate 3D poses are measured using a MoCap system. Accurate 2D poses are also available for each camera view. To exhaustively compare our results with the state of the art, we use the three different protocols used in the literature. The first one, denoted as P1, is introduced in [59] and employed in [8], [33]: six subjects (S1, S5, S6, S7, S8 and S9) are used for training and every 64^{th} frame of subject S11/camera 2, *i.e.*, a total of 928 frames, are used for testing. We report the 3D pose error (mm), averaged over the 13 joints. Since most methods report results using more than 13 joints, we also present results for a model trained to estimate 17 joints instead of 13, adding pelvis, back, torso and neck keypoints. As in [33], we report a 3D pose error that measures accuracy of

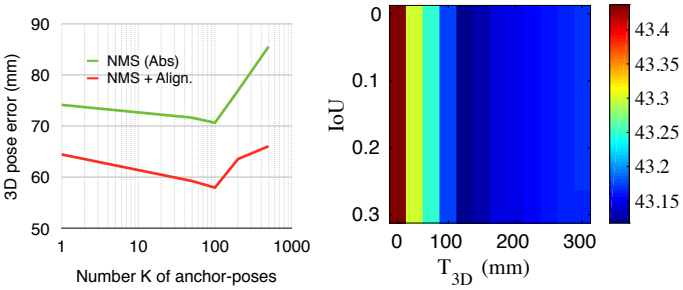
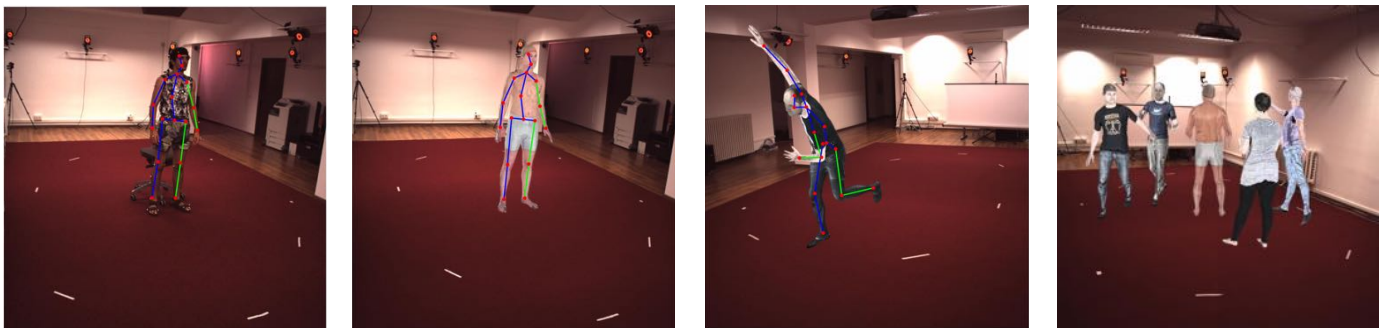


Fig. 7. Average 3D pose error in mm on Human3.6M protocol P1 with respect to the number K of anchor-poses (a) and the 2 PPI thresholds (b). Note that results on in (a) are reported for NMS with/without rigid alignment for a model regressing 13 joints and trained during 100k iterations. Results in (b) are obtained after rigid alignment with our best architecture trained to regress 17 joints.

pose aligned with a rigid transformation (Align.), but also report the absolute error (Abs.). The second protocol, denoted as P2, is used in [11], [40], [43]. All the frames from subjects S9 and S11 are used for testing and only S1, S5, S6, S7 and S8 are used for training. We evaluate only on every 5^{th} frame as in [43], *i.e.*, on a test set of 110k images, as we did not observe a significant impact on performance when evaluating on all the frames. The last protocol P3, introduced by Bogo *et al.* [25], uses the same subjects for training and testing as P2. However, evaluation is performed only on sequences from camera 3 / trial 1 after rigid alignment.

Anchor-poses. We select a subset of the training set, *i.e.*, 190k images and the corresponding 3D poses as in [8], to build a set of anchor-poses by clustering the 3D poses using K-means. Figure 7a shows the performance obtained when varying the number K of anchor-poses with a simple NMS, *i.e.*, taking the top scoring pose proposal as 3D pose estimate. Best performance is obtained for K=100. When K is too small, for instance if K=1 which corresponds to a standard regression, the number of anchor-poses might not be sufficient to cover the pose space. When K becomes



(a) (b) (c) (d)

Fig. 8. Human3.6M real and synthetic training data. We show a training image from protocol 2 with the overplayed 2D pose in (a). In (b), we show a synthetic "surreal" type of image obtained after rendering the SMPL model using the Human3.6M 3D pose from (a) and a randomly picked body shape and texture map. Note that for more realism, the surreal image is rendered at the exact same 3D location in the MoCap room, using the camera parameters and background from the real image (a). In (c), we show an example of image synthesized using a 3D pose from the CMU motion capture dataset. In (d), we show a multi-person image generated using 5 poses from the CMU MoCap dataset.

too large, the error also increases since the anchor-poses are too similar, resulting in ambiguities in the classification. We select K=100 classes for the remaining experiments on Human3.6M.

Additional synthetic training data. One of the conclusions in the earlier version of this work [14] was that LCR-Net required a significant amount of training data that could be generated through synthesis. To augment the training set with synthetic images with associated 3D poses, we render the SMPL 3D human mesh model [60]. For more realism, we render these images in the Human3.6M capture room using background images and camera parameters provided with the data (see Figure 8). We generate images for a same quantity of poses and consider two sets of 3D MoCap poses: a) the same poses from Human3.6M to add appearance variations (Figure 8b) and b) poses from the CMU dataset [6] to add variations both in terms of appearance and poses (Figure 8c-d). To ensure a balanced training set, we sample CMU poses in areas of the pose space that are less populated by Human3.6M poses. In both cases, we use the SMPL body parameters and texture maps from [53]. The SMPL kinematic model is somehow different from the Human3.6M 3D model: some of the 17 joints from Human3.6M poses do not correspond exactly to their SMPL counterparts (e.g., head, hips and shoulders) while others are simply missing (neck and torso). To tackle this issue, we trained a regressor from SMPL to Human3.6M poses using the body parameters estimated by [53], to "correct" the misplaced or missing joints for the CMU-based images for which we do not have Human3.6M-like pose annotations. We obtained satisfactory 17-joint poses for all the synthesized images. See examples in Figure 8c-d. In total, we obtained a training set of 557k images and trained the different models for 500k iterations (roughly 8 epochs) using SGD, 300k iterations at a learning rate of 10^{-3} and 200k with a learning rate of 10^{-4}.

Impact of PPI. We merge poses that are (a) highly overlapping in 2D, *i.e.* for which the bounding boxes intersection over union is over the IoU threshold and (b) close in 3D pose space, *i.e.* whose 3D Euclidean distance is below T_{3D}. We experimentally set T_{3D} to 125 mm and found that the IoU threshold has no influence on the performance for this dataset (see Figure 7b), as only one individual is observed and all highly scored proposals are localized on the subject. In most cases, the highest scoring pose proposal (NMS) is already an accurate estimation but, on average, the improvement achieved by our PPI over the NMS estimates is non negligible. On protocol P1, we obtain an average error of 54.2 mm after NMS and 53.5 mm after PPI (43.7 mm and 43.1 mm after rigid alignment) when evaluating on 17 joints. In Figure 9, we show some qualitative results where examples are sorted by increasing 3D pose error. A green upward peak with respect to the blue curve corresponding to PPI indicates an important improvement by the PPI, whereas a red peak downward indicates poses where the rigid alignment helps correct the most. For the 928 test frames of protocol P1, less than 20 have an error equal to or greater than 90 mm. This occurs in cases of unseen poses in the training set, see rightmost example in Figure 9.

Ablative analysis. Our complete method, called LCR-Net++, significantly improves over the initial and simpler version of LCR-Net (as published in [14]), *i.e.*, a model trained without synthetic data and an architecture that does not include iterative refinement, RoI alignment and rescoring of the pose proposals. To better understand the origin of this improvement, an ablative analysis is provided in Table 1 for a model trained to estimate 13 joints. We can see that the biggest improvements are obtained when adding synthetic images to the training set. By adding variability in terms of appearance, *i.e.*, adding synthetic data rendered using Human3.6M poses, we decrease the 3D error by 15 mm. We can see that the gap between Abs. and Align. results is smaller (10.9 mm vs 16.1 mm without using synthetic data), meaning that we better estimate the camera viewpoint. Adding synthetic training images rendered from new poses (CMU) further improves the performance by another 5 mm. This validates the fact that our approach requires a large and varied training set in terms of pose and appearance. The RoI alignment and iterative process do not help improve the performance on Human3.6M significantly as the 3D estimations are already quite accurate. Rescoring the pose proposals following Equation 6 (in Section 3.6) allows the NMS to select better pose proposals and the PPI to produce better pose estimates after integration, *i.e.*, the 3D error decreases by 3 mm. In Figure 10, we report the Percentage of Correct Keypoints (PCK), *i.e.*, the ratio of joints for which the error is below a threshold, on Human3.6M protocol P1. When computing the upper bound, *i.e.*, taking the pose proposal closest to ground-truth pose and thus simulating a perfect scoring, we observe a boost in performance, both before and after rigid alignment. This indicates that even after applying our rescoring function, the top scoring pose proposals are not always the ones that best explain the input images. In some

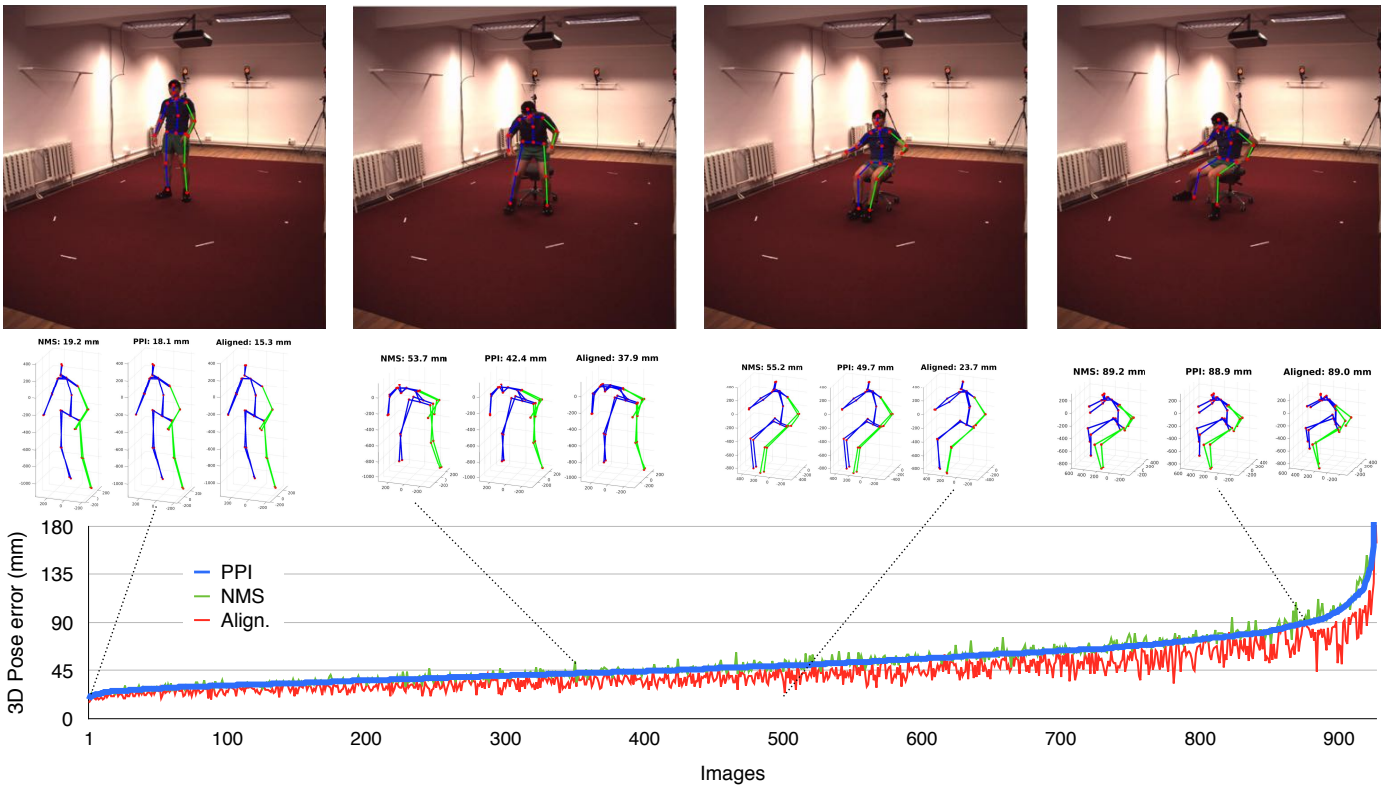


Fig. 9. Average 3D pose error on Human3.6M test images (protocol P1). We order the examples by increasing error of PPI results (blue) and also report the performance with a simple NMS (green) and after rigid alignment of the PPI estimation (red). We show qualitative results for 4 particular cases, from left to right: (a) an image where NMS estimation is already accurate, thus PPI and alignment do not further improve, (b) a case in which the PPI achieves an accurate pose estimate, (c) a case where PPI does not improve over NMS but the alignment helps to correct the pose estimate and (d) a failure case where the pose is not satisfactory, even after rigid alignment. For each case, we show the image with the estimated 2D pose (with PPI). We also show the 3D poses estimated by NMS, PPI and after alignment overlaid with the ground-truth 3D pose.

cases, information from previous frames could help disambiguate and adequately rescore the pose proposals. In future work, our method could be extended to leverage such additional temporal information, which should further improve the performance.

	NMS (Abs.)	PPI (Abs.)	PPI (Align.)
LCR-Net [14]	89.8	87.7	71.6
+synth H3.6M	73.3	73.9	63.2
+synth CMU	68.5	68.9	59.3
+RoI align	68.3	69.3	59.6
+iterative estimation	67.7	68.7	59.3
+rescoring (**LCR-Net++**)	**66.8**	**65.8**	**56.4**

TABLE 1
Ablative analysis on Human3.6M protocol P2 (evaluating on 13 joints). We evaluate the performance of LCR-Net when adding the different modifications introduced in this work compared to the simpler version published in [14], *i.e.*, with a RoI pooling layer and trained on Human3.6M training set only. For each tested model/architecture, the average absolute 3D pose error (mm) is reported for NMS, and also PPI before (Abs.) and after rigid 3D alignment (Align.)

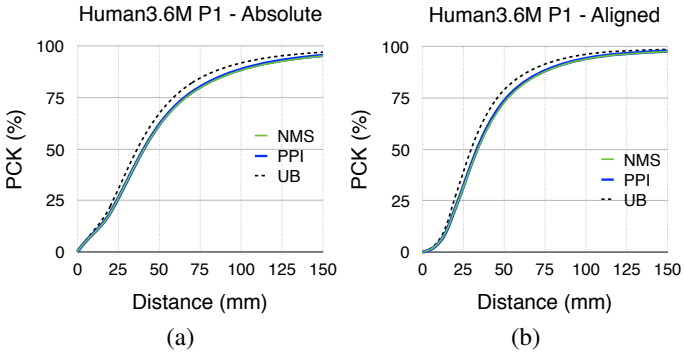


Fig. 10. Average Percentage of Correct Keypoints PCK (%) on Human3.6M protocol P1. Detection rate with respect to the distance to ground truth 3D joints is given for PPI, NMS and the Upper bound (UB), *i.e.*, taking the pose proposal closest to ground-truth pose. Performances are given before (a) and after (b) rigid alignment to the ground-truth poses.

Detailed comparison with the state of the art. We now extensively compare our method with the state of the art. First, Table 3 compares our complete method (LCR-Net++) to other recent competing approaches on the three protocols P1, P2 and P3. We also compare with the simpler version of LCR-Net [14]. For a fair comparison, we group methods that consider 13-14 or 16-17 joints and do not include results where a different model was trained for each action. First, we can observe that the average 3D pose error obtained by our LCR-Net++ decreases when considering more keypoints. These additional joints, *i.e.*, pelvis, back, torso and neck keypoints, are easier to estimate compared to extremities of limbs such as wrists and ankles. Adding them in the computation of the pose error artificially improves the performance. We outperform all other methods for the 3 protocols of the literature and significantly improve over our previous results, especially on protocol P2 (65.8 mm vs 87.7 mm

Method (num. joints)	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	Walk	WalkD.	WalkT.	Avg.
Rogez & Schmid [8] (13 jts)	94.5	110.4	109.3	143.9	125.9	160.3	95.5	89.8	134.2	179.2	123.8	133.0	77.4	129.5	91.3	121.2
Chen & Ramanan [29] (14 jts)	89.9	97.6	90.0	107.9	107.3	139.2	93.6	136.1	133.1	240.1	106.7	106.2	114.1	87.8	90.6	114.2
Rogez & Schmid [54] (13 jts)	87.7	100.7	93.6	139.6	107.9	155.2	88.1	78.9	119.0	171.9	107.4	130.7	71.6	114.6	83.1	110.6
Nie *et al.* [31] (13 jts)	90.1	88.2	85.7	95.6	103.9	92.4	90.4	117.9	136.4	98.5	103.0	94.4	86.0	90.6	89.5	97.5
Moreno-Noguer [30] (14 jts)	67.5	79.0	76.5	83.1	97.4	**74.6**	72.0	102.4	116.7	**87.7**	100.4	94.6	75.2	82.7	74.9	85.6
LCR-Net [14] (13j)	76.2	80.2	75.8	83.3	105.7	92.2	79.0	71.7	105.9	127.1	88.0	83.7	64.9	86.6	84.0	87.7
LCR-Net++ (13j)	**53.4**	**59.1**	**61.8**	**59.6**	**72.3**	78.3	**54.1**	**55.7**	**95.6**	99.5	**68.7**	**59.4**	**47.1**	**66.3**	**56.4**	**65.8**
Sanzari *et al.* [64] (17 jts)	**48.8**	56.3	96.0	84.8	96.5	105.6	66.3	107.4	116.9	129.6	97.8	65.9	92.6	130.5	102.2	93.1
Tome *et al.* [48] (17 jts)	65.0	73.5	76.8	86.4	86.3	110.7	68.9	74.8	110.2	173.9	84.9	85.8	71.4	86.3	73.1	88.4
Pavlakos *et al.* [42] (17 jts)	67.4	71.9	66.7	69.1	71.2	77.0	65.0	68.3	83.7	96.5	71.7	65.8	59.1	74.9	63.2	71.9
Tekin *et al.* [47] (17 jts)	53.9	62.2	61.5	66.2	80.1	79.5	64.6	83.2	**70.9**	107.9	70.4	68.0	52.8	77.8	63.1	70.8
Katircioglu *et al.* [65] (17 jts)	54.9	63.3	**57.3**	62.3	70.3	77.4	56.7	57.1	79.0	97.1	64.3	61.9	49.8	67.1	62.3	65.4
Zhou *et al.* [50] (16 jts)	54.8	60.7	58.2	71.4	**62.0**	**65.5**	53.8	55.6	75.2	111.6	64.15	66.05	63.2	51.4	55.3	64.9
Martinez *et al.* [32] (17 jts)	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	**62.3**	59.1	49.5	65.1	**52.4**	62.9
LCR-Net++ (17j)	50.9	**55.9**	63.3	**56.0**	65.1	70.7	**52.1**	**51.9**	81.1	**91.7**	64.7	**54.6**	**44.7**	**61.1**	53.7	**61.2**

TABLE 2
Per-class results on Human3.6M protocol P2 (without pose alignment). We report 3D pose error results (mm) for recently published works that provide per-class performance and employ a single "general" model, *i.e.*, a single model covering the 15 actions.

Methods (num. joints)	P1 (Abs.)	**P1 (Align.)**	**P2 (Abs.)**	P2 (Align.)	**P3 (Align.)**
Bo & Sminchisescu [61] (14 jts)	-	117.9	-	-	-
Kostrikov & Gall [59] (14 jts)	-	115.7	-	-	-
Iqbal *et al.* [33] (14 jts)	-	108.3	-	-	-
Du *et al.* [62] (14 jts)	-	-	126.5	-	-
Bogo *et al.* [25] (14 jts)	-	-	-	-	82.3
Rogez & Schmid [8] (13 jts)	126	88.1	121.2	87.3	-
Chen & Ramanan [29] (14 jts)	-	82.7	114.2	-	-
Rogez & Schmid [54] (13 jts)	116.7	90.1	110.6	-	-
Nie *et al.* [31] (13 jts)	-	79.5	97.5	-	-
Moreno-Noguer [30] (14 jts)	-	74.0	85.6	-	81.5
LCR-Net [14] (13 jts)	63.2	53.4	87.7	71.6	72.7
LCR-Net++ (13 jts)	**56.8**	**48.3**	**65.8**	**56.4**	**57.2**
Li *et al.* [12] (17 jts)	-	-	136.5	-	-
Li *et al.* [40] (17 jts)	-	-	122	-	-
Tekin *et al.* [11] (17 jts)	-	-	125.0	-	-
Park *et al.* [10] (17 jts)	-	-	117.3	-	-
Zhou *et al.* [43] (17 jts)	-	-	113.0	-	-
Zhou *et al.* [63] (17 jts)	-	-	107.26	-	-
Sanzari *et al.* [64] (17 jts)	-	-	93.1	-	-
Tome *et al.* [48] (17 jts)	-	70.7	88.4	-	79.6
Mehta *et al.* [49] (17 jts)	72.8	-	74.14	-	-
Pavlakos *et al.* [42] (17 jts)	-	-	71.9	51.9	-
Tekin *et al.* [47] (17 jts)	-	-	70.81	50.1	-
Katircioglu *et al.* [65] (17 jts)	-	-	65.4	-	-
Zhou *et al.* [50] (16 jts)	-	-	64.9	-	-
Martinez *et al.* [32] (17 jts)	-	-	62.9	**47.7**	-
Sun *et al.* [44] (16 jts)	-	48.3	-	-	-
Kinauer *et al.* [66] (16 jts)	-	45.9	-	54.5	-
LCR-Net++ (17 jts)	**53.5**	**43.1**	**61.2**	49.4	**50.5**

TABLE 3
Comparison with state-of-the-art results on Human3.6M for 3 different protocols. The average 3D pose error (mm) is reported before (Abs.) and after rigid 3D alignment (Align.) for protocols P1 and P2. We group the methods according to the number of joints that are evaluated (13-14 or 16-17). The errors are globally higher with P2 and P3 that provide less training subjects and have a larger and more varied test set.

with 13 joints) which is the most difficult one as less training subjects are available and a larger and more varied test set is considered. On this protocol, we establish a new state-of-the-art performance both with 13 (65.8 mm) and 17 joints (61.2 mm) and outperform all previously published methods, including very recent work, despite the fact that (a) we also perform localization, in contrast to most methods such as [8], [42], [50] that assume a bounding box annotation of the human and (b) we propose an end-to-end architecture trained with Human3.6M images only while other methods rely on off-the-shelf 2D pose detectors [29], [30], [32], [47]. Note that we do not include in this table the results reported by [44] on P2 Abs. (59.1 mm) as the authors did not follow the exact same protocol and evaluated on a much smaller subset of the test images, 9.6k randomly sampled images instead of 110k, making the comparison unfair. When adding a rigid transformation for protocol P2, the method of [32] achieves a slightly better performance than ours, whereas LCR-Net++ performs better without alignment. This means that their estimation of the camera viewpoint is less accurate than ours and that aligning the poses in 3D helps to correct this lack of accuracy. We present a per-class comparison on protocol P2 in Table 2. Compared to methods estimating 13-14 joints, LCR-Net++ is state of the art for 13 out of 15 actions and only performs lower than [30] for "taking photo" and "sit down" actions. In the case of 16-17 joints, our method is state of the art for 8 out of 15 actions. The methods from [30], [32], [47], [50] or [65] report better performance for the remaining 7 actions. Katircioglu *et al.* [65] leverage temporal information. All the other methods rely on heatmaps or 2D joints detected by [20] or [3] while our architecture is trained end-to-end using only the Human3.6M training set and synthetic data.

4.2 2D and 3D pose detection on MPII

Datasets and evaluation protocols. We now present experimental results for 2D and 3D pose detection in real-world images. We use the challenging MPII human pose dataset [58] that consists of around 40k annotated 2D poses in around 25k images (17,4k for training and 7k for testing). It contains a large variety of camera viewpoints and poses, originating from around 400 different actions. Each scene can contain multiple people, that are often occluded or truncated by the image boundary. This makes the dataset challenging for human pose estimation. While most other papers on 3D pose estimation only show qualitative examples on real images, we analyze our results on a validation set of 1k images that we used for both single (1,088 poses) and multi-person (209 groups) protocols. This set is obtained by randomly splitting the training dataset to create a training set of 16,4k images and a validation set of 1k images, making sure that images from the same video all belong to the same set. For training, we also use the annotated images from LSPE as in [8], [22], a subset of 17k images from Human3.6M as in [8] and the training set of the MS Coco dataset [67]. After mirroring, we obtain a training set of 161k images with around 290k annotated humans. To understand the influence of the training data on the performance, we further increased the size of the training set by adding synthetic images rendered using the CMU Mocap dataset as in Section 4.1. This time, we synthesized 40k multi-person images with 186k humans

(an average of 4-5 persons per image) as the example depicted in Figure 8d. We trained the different models for 500k iterations (roughly 20 epochs), 300k iterations at a learning rate of 10^{-3} and 200k with a learning rate of 10^{-4}. For single-person pose estimation, we report the results using the PCKh metric that measures the ratio of estimated joints for which the distance to the ground-truth is below a threshold. The standard threshold is set to half the size of the head, *i.e.*, PCKh@0.5. In this setting, most methods use person localization information before computing the pose. In our case, we detect the poses for the entire image and use the localization information only for evaluation, *i.e.*, to select the pose that corresponds to each ground-truth. For multi-person evaluation, we follow the standard protocol and evaluate the average precision (AP). Both PCKh and AP are averaged over 14 joints, the 14^{th} joint (top of the head) being extrapolated from our 13-joint pose.

Dealing with truncation. To deal with truncations by the image boundary, we double the number of clusters by considering also upper-body region proposals. More precisely, for the K anchor-poses, we adjust the full-body anchor-pose such that only the upper-body covers the candidate box but we still regress the full-body pose. This process allows us to "hallucinate" valid full-body poses even when only the upper-body is visible. At training, we define an upper-body ground-truth box for each annotated pose plus a fully-body ground-truth box when at least one joint from the lower limbs is visible. By this process, we obtain 476k upper-body and 415k full-body poses in our training set.

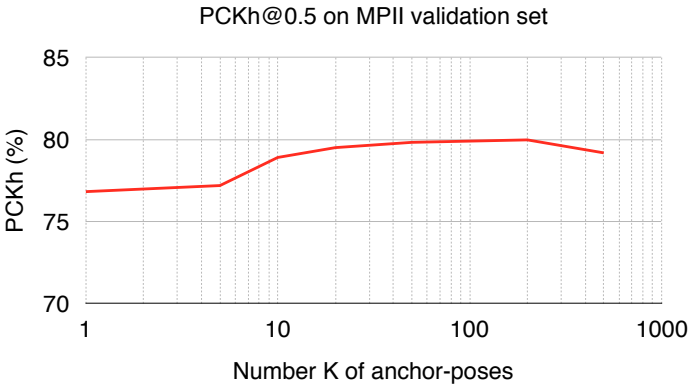


Fig. 11. PCKh@0.5 (%) on MPII validation set with respect to the number K of anchor-poses. Results are reported after PPI with T_{3D}=0.2 m and IoU=0.2.

Pseudo ground-truth 3D pose and anchor-poses. LCR-Net requires 3D ground-truth poses associated with each training image. For MPII, LSPE and MS Coco images, we infer them using the proposed nearest neighbor (NN) search on the annotated joints, see Section 3.7. We consider the CMU MoCap dataset as 3D pose source, as in [8], [33]. However, both MPII and LSPE datasets present rare poses (*e.g.*, gymnastic) that are absent from this dataset. To cover a wider set of poses, we merged several MoCap datasets available on the internet, such as Pose Prior [23] and HDM05 [68], and observed a 13% reduction in the matching error, *i.e.*, distance between the query 2D pose and the best match, when using this augmented dataset. The set of anchor-poses is obtained by running K-means on the 3D poses of the extended MoCap dataset. In Figure 11, we show PCKh when varying the number K of anchor-poses. Compared to Human3.6M, the diversity in pose is significantly higher and we found that an optimum number is reached for K=200. We keep K=200 anchor-poses for the remaining experiments in the following.

Impact of PPI. We experimentally set T_{3D} to 130 mm and IoU to 0.12 to evaluate PCKh, see Figure 12. As expected, the IoU threshold has greater impact on multi-person AP than on single person PCKh: with a small IoU, pose proposals corresponding to different persons with a high spatial overlap in 2D can be accidentally merged if they correspond to similar 3D poses. A group of people moving together (*e.g.* dancers) is a typical failure case. With T_{3D}=130 mm and IoU=0.12, we obtain PCKh@0.5=82.16%. With T_{3D}=30 mm and IoU=0.54, we obtain AP=54.31%.

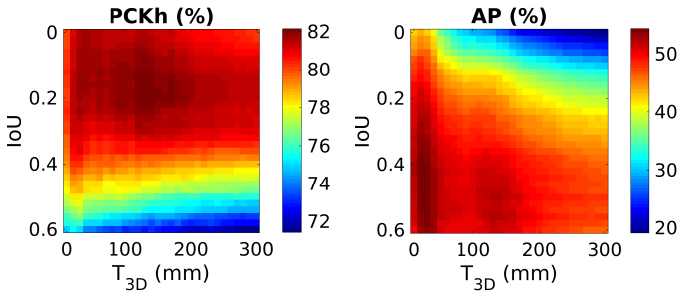


Fig. 12. Single-person PCKh@0.5 (left) and multi-person AP (right) on MPII validation set when varying IoU and T_{3D}.

	NMS	PPI
LCR-Net [14]	69.87	75.21
+ MS-Coco training set	74.84	79.95
+ Synthetic data	76.30	80.79
+ RoI align	78.36	81.32
+ iterative estimation	78.76	81.78
+ rescoring (**LCR-Net++**)	**80.30**	**82.16**

TABLE 4
Ablative analysis on MPII validation set. We evaluate the PCKh@0.5 (%) of our architecture when adding the different modifications introduced in this work compared to the version of LCR-Net published in [14] with a RoI pooling layer and trained on MPII+LSPE+Human3.6M images. For each tested model, the PCKh@0.5 (%) is reported for NMS and after PPI with T_{3D}=130 mm and IoU=0.12.

Ablative analysis. We provide an ablative analysis for single person pose estimation on the MPII validation set in Table 4. On our validation set, the initial version of LCR-Net [14] (trained on MPII, LSPE and Human3.6M) obtains 75.21% for a standard PCKh@0.5. We can see that PPI (with T_{3D}=130 mm and IoU=0.12) improves with respect to NMS by 5.34%. When adding annotated images from MS-Coco [67], *i.e.*, approximately doubling the size of the training set, a significant improvement in performance is obtained, PCKh@0.5=79.95% on the validation set. This confirms that LCR-Net requires a large amount of training data. While using additional synthetic data had a strong impact on the performance in the experiments on the Human3.6M dataset, the improvement is less substantial when evaluating on MPII validation set (+0.84%). A possible explanation is that generating useful synthetic data is much harder in-the-wild than in the controlled Human3.6M scenario where no occlusions, no object manipulations and no multi-persons scenes are observed. We can see that the RoI alignment greatly improves the quality of the region features, since the gap between NMS and PPI results

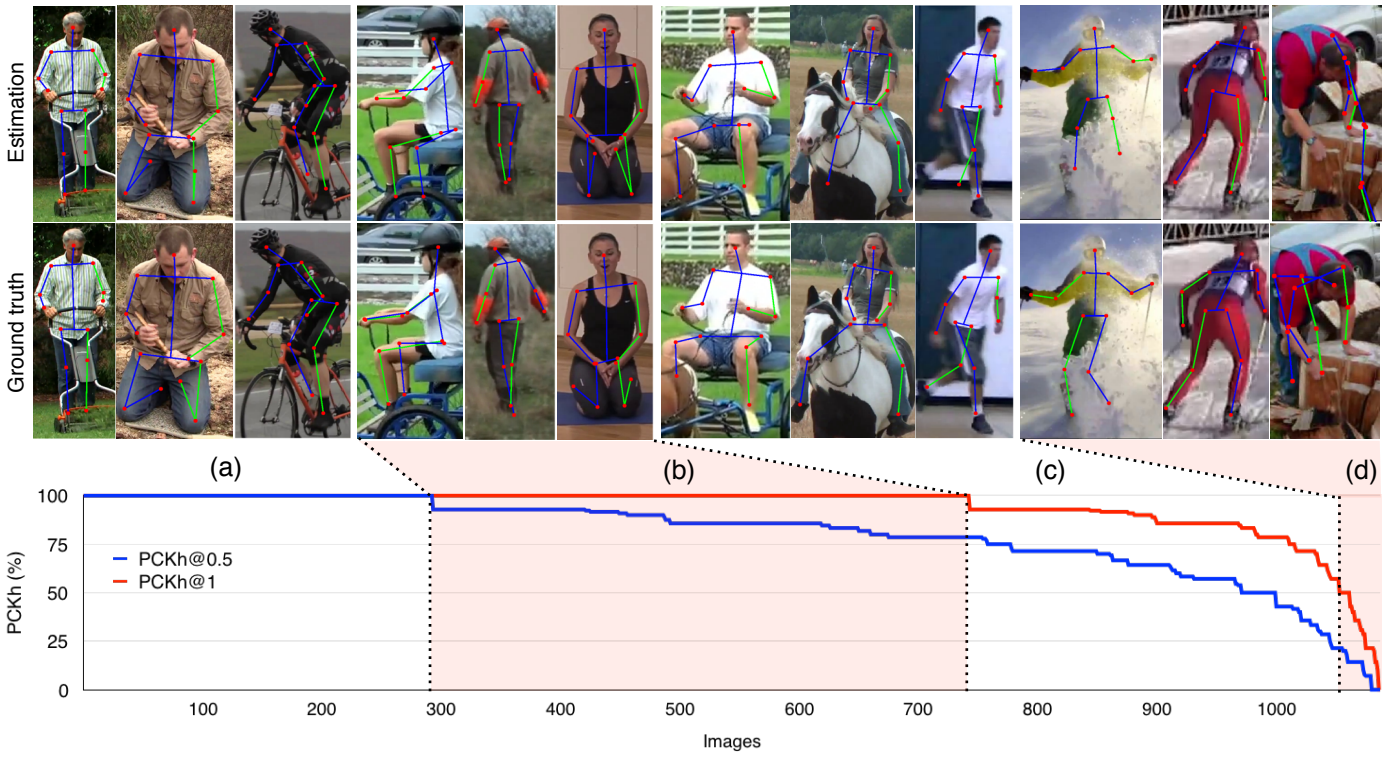


Fig. 13. Qualitative analysis on MPII validation set. The average "per-pose" PCKh@0.5 and PCKh@1 is represented (bottom) when ordering the poses with respect to PCKh score. This helps visualize (from left to right) the poses that are (a) perfectly recognized (30% of the poses), (b) correct but imprecise (42.5%), (c) partly incorrect (25%) and (d) miss-detected (2.5%). For each category, we show 3 exemples of estimated poses (top) and the corresponding ground truth annotations (middle). Note that in average, PCKh@0.5=82.16 and PCKh@1=94.74.

decreases from 4.49% to 2.96%. The iterative refinement improves the performance by another 0.46%. Finally, the rescoring of the pose proposals (Equation 6 in Section 3.6) helps to improve the NMS estimates by 1.54% but has a marginal influence on PPI results (+0.38%). LCR-Net++ produces significantly more accurate propose proposals than its initial version and the PPI post-processing stage still improves over the simpler NMS but to a lesser extent (1.86%).

Impact of regression target. Since 2D and 3D poses are regressed together, inaccurate 3D annotations could negatively impact 2D pose estimation. To evaluate the effect of the pseudo ground-truth on 2D performance, we train a version of the architecture to predict the 2D poses only (see Table 5). We observe a decrease of the NMS performance obtaining a PCKh of 74.61% compared to 76.30% with 2D+3D regression. Adding the 3D pose regression actually helps to improve the performance in 2D. Finally, we evaluate PCKh when only considering full-body classes and observed a lower performance after NMS and PPI, validating that adding upper-body classes to the full-body classes improves performance on the MPII validation set.

Detailed analysis. While we outperform the state of the art in 3D human pose estimation in a controlled environment, our 2D performance on real images is below the state of the art on the MPII test set, as reported in Table 6. Note that in contrast to most other approaches, our holistic method also gives an estimation of the occluded joints that is not evaluated. Figure 13 shows the "per pose" PCKh on the validation set for PCKh@0.5 and PCKh@1. The poses are ordered with respect to PCKh score. We can see (from left to right) that 72.5% of the poses are globally correct (30% of the poses are perfectly recognized and 42.5% are simply imprecise) while 25% are partly incorrect, e.g. a limb is poorly estimated, and 2.5% of the poses are miss-detected, *i.e.*, PCKh@1$\leq$50%. These misdetections are often due to a right-left inversion in the estimation (or in the ground-truth annotations) leading to very poor PCKh scores as visualized in the examples. We can see on Figure 14a that PCKh@1 approaches 95%. Although globally correct, our pose estimations can lack precision on the limb extremities resulting in lower PCKh score in

	NMS	PPI
Baseline ([14] + MS-Coco train + Synth)	**76.30**	**80.79**
Regressing 2D pose only	74.61	-
Using full-body classes only	74.42	78.40

TABLE 5
Additional analysis on MPII validation set. We evaluate the performance of LCR-Net (a) when predicting only the 2D poses and (b) when using only full-body classes, *i.e.*, no upper-body classes. For each tested model, the PCKh@0.5 (%) is reported for NMS and after PPI with T_{3D}=130 mm and IoU=0.12.

Method	Human3.6M 2D pose error (pix)	MPII PCKh@0.5 (%)
Wei *et al.* [20]	10.04	**88.5**
LCR-Net	7.4	74.2
LCR-Net ++	**5.9**	75.3

TABLE 6
2D pose estimation results on Human3.6M and MPII test sets compared to state-of-the-art 2D method [20].

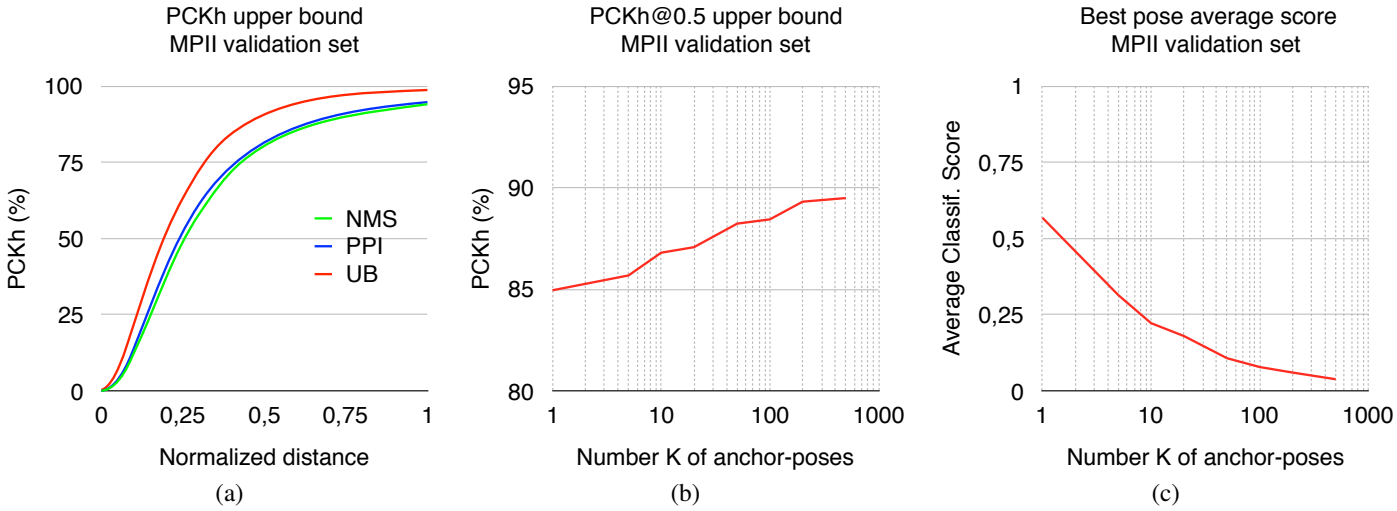


Fig. 14. Upper bound on MPII validation set. (a) Detection rate with respect to the normalized distance in PCKh computation for PPI, NMS and the Upper bound (UB), *i.e.*, taking the pose proposal closest to ground-truth pose. Results are reported for K=200 anchor-poses. (b) Upper bound of the PCKh@0.5 when varying K, the number of anchor-poses. (c) Average score of the pose proposals used to compute the upper bound.

2D. One explanation is that we use a fully-connected layer for the regression. This could be improved by using fully convolutional architecture with deconvolution or upsampling [3]. Another possible explanation is that the pose proposals are not correctly scored. On Figure 14a, we can see that if we compute the upper bound on the PCKh, *i.e.*, computed with the closest pose proposals from ground-truth annotations, we can obtain greater performances: PCKh@0.5=90.64 and PCKh@1=98.72. This indicates that the classification score is not always representative of the quality of the regressed 2D and 3D poses. Some high scoring poses can in fact be imprecise while others with lower scores are more accurate. In Figure 14b, we show this upper bound of the PCKh@0.5 when varying the number K of anchor-poses. Adding more anchor-poses clearly helps generate better pose proposals but their score decreases when augmenting K as shown in Figure 14c. The anchor-poses become probably too similar and harder to distinguish, resulting in ambiguities in the classification. Another reasons for this observation could be the amount of training data available for each class that also decreases when increasing K. We proposed a rescoring function that helps to improve both NMS and PPI performances but a better scoring function should be investigated in future work.

Multi-person pose detection. For multi-person evaluation, our validation set contains 209 groups of multiple people in 187 images. We follow the standard protocol and evaluate AP averaged over joints. We obtain 54.3% for a standard mAP@0.5 and over 62% for a mAP@1. This is below state-of-the-art, e.g., [69] reports mAP@0.5=77.5% on the test set, but we also estimate the 3D poses unlike all existing multi-person approaches who only focus on 2D pose estimation. Examples of multi-person pose detection are shown in Figure 15. Our method is able to detect multiple people even if they overlap (second row, second column). It is also robust to unusual poses (top right), truncation (top row, third column) or important occlusions (top row, second column).

5 Conclusion

This paper introduces a Localization-Classification-Regression network (LCR-Net) for joint 2D and 3D human pose detection in natural images. We demonstrate the benefit of an end-to-end architecture which relies on pose proposals that are hypothesized at different locations in the image, scored by classification and refined by regression. The final pose estimation is obtained by integrating over neighboring pose hypotheses. We outperform the state of the art in 3D pose estimation in controlled environments and show promising results on real images.

The upper bound performance shows that there is room for improvement and that a considerable boost could be obtained by adequately scoring the pose proposals. Our first attempt at rescoring them has shown encouraging results in that direction. Another line of improvement concerns the training data. In this work, we proposed a solution to automatically annotate 2D images with "pseudo" ground-truth 3D poses. Our ongoing research indicates that better 3D and 2D performances could be obtained with LCR-Net if more accurate training data was available.

Acknowledgements. This work was supported by ERC advanced grant Allegro and an Amazon Academic Research Award. We thank NVIDIA for donating the GPUs used for this research.

References

[1] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *CVPR*, 2017. 1, 2, 3, 4
[2] A. Bulat and G. Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," in *ECCV*, 2016. 1, 2
[3] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *ECCV*, 2016. 1, 2, 4, 9, 12
[4] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *IJCV*, 2010. 1, 3
[5] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. PAMI*, 2014. 1, 3, 6
[6] "CMU motion capture dataset. http://mocap.cs.cmu.edu. the database was created with funding from nsf eia-0196217." 1, 3, 7
[7] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen, "Synthesizing training images for boosting human 3D pose estimation," in *3DV*, 2016. 1, 3
[8] G. Rogez and C. Schmid, "MoCap-guided data augmentation for 3D pose estimation in the wild," in *NIPS*, 2016. 1, 3, 6, 9, 10
[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015. 1, 2, 4, 5

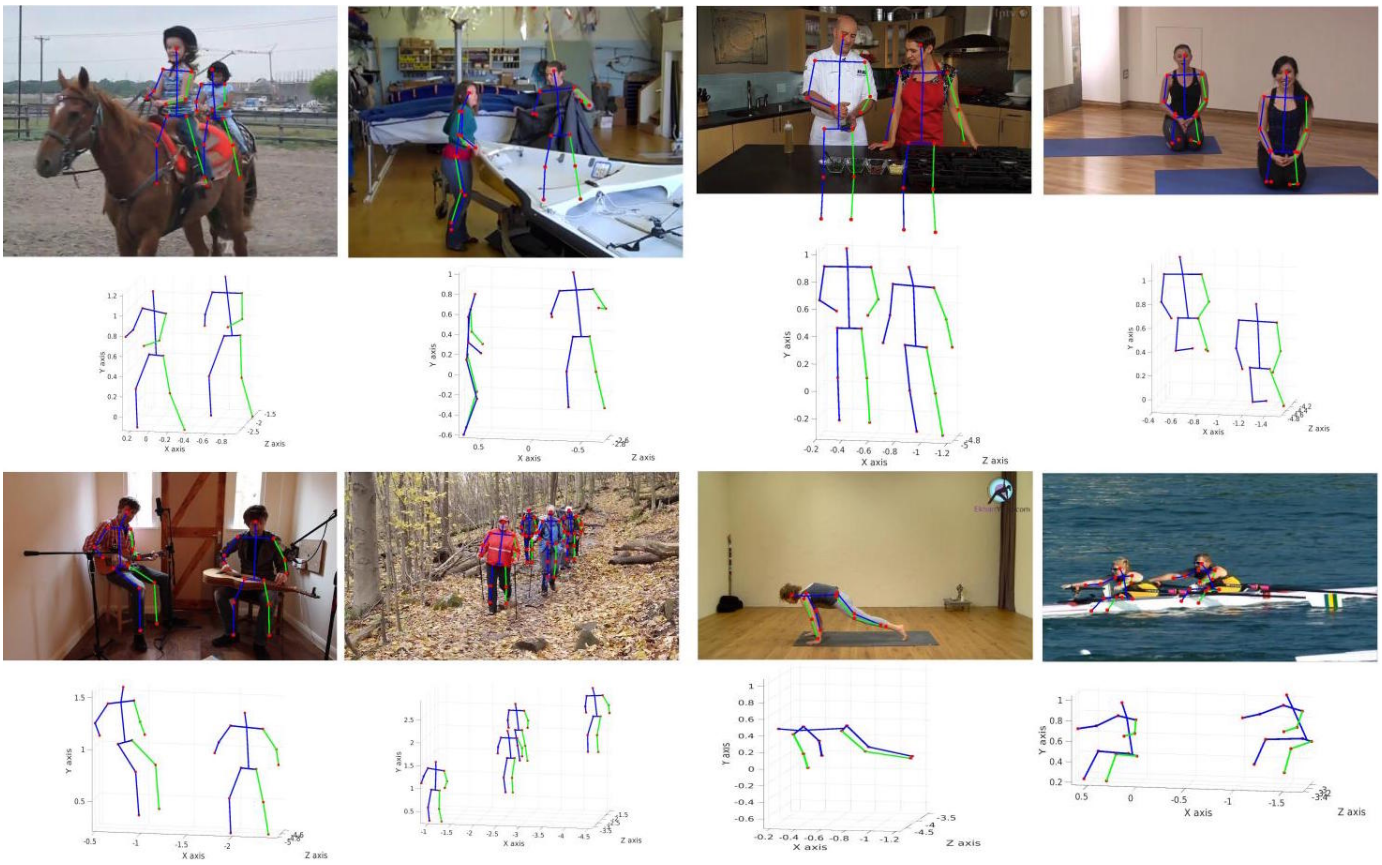


Fig. 15. Qualitative examples. LCR-Net outputs multiple 2D and 3D poses, the 3D poses being expressed in a camera reference system centered on the torso. To represent the 3D poses in a common coordinate system, we find for each of them the appropriate 3D displacements in front of the camera. This is obtained using a least square minimization of the reprojection error, *i.e.*, the distance between 2D pose and reprojected 3D pose. When the camera is unknown, hypothesizing an orthographic camera leads to acceptable qualitative results as shown in these examples.

[10] S. Park, J. Hwang, and N. Kwak, "3D human pose estimation using convolutional neural networks with 2D pose information," in *ECCV Workshop*, 2016. 1, 9

[11] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua, "Direct prediction of 3D body poses from motion compensated sequences," in *CVPR*, 2016. 1, 3, 6, 9

[12] S. Li and A. B. Chan, "3D human pose estimation from monocular images with deep convolutional neural network," in *ACCV*, 2014. 1, 3, 9

[13] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *CVPR*, 2014. 1, 2, 3

[14] G. Rogez, P. Weinzaepfel, and C. Schmid, "LCR-Net: Localization-Classification-Regression for Human Pose," in *CVPR*, 2017. 2, 7, 8, 9, 10, 11

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017. 2, 5

[16] X. Chen and A. L. Yuille, "Articulated pose estimation by a graphical model with image dependent pairwise relations," in *NIPS*, 2014. 2

[17] X. Fan, K. Zheng, Y. Lin, and S. Wang, "Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation," in *CVPR*, 2015. 2, 3

[18] W. Ouyang, X. Chu, and X. Wang, "Multi-source deep learning for human pose estimation," in *CVPR*, 2014. 2

[19] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *NIPS*, 2014. 2

[20] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016. 2, 9, 11

[21] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," in *CVPR*, 2017. 2, 3

[22] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," *CVPR*, 2016. 3, 9

[23] I. Akhter and M. Black, "Pose-conditioned joint angle limits for 3D human pose reconstruction," in *CVPR*, 2015. 3, 10

[24] X. Fan, K. Zheng, Y. Zhou, and S. Wang, "Pose locality constrained representation for 3D human pose reconstruction," in *ECCV*, 2014. 3

[25] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image," in *ECCV*, 2016. 3, 6, 9

[26] E. Simo-Serra, A. Ramisa, G. Alenyà, C. Torras, and F. Moreno-Noguer, "Single image 3D human pose estimation from noisy observations," in *CVPR*, 2012. 3

[27] C. Wang, Y. Wang, Z. Lin, A. L. Yuille, and W. Gao, "Robust estimation of 3D human poses from a single image," in *CVPR*, 2014. 3

[28] V. Ramakrishna, T. Kanade, and Y. A. Sheikh, "Reconstructing 3D Human Pose from 2D Image Landmarks," in *ECCV*, 2012. 3

[29] C. Chen and D. Ramanan, "3D human pose estimation = 2D pose estimation + matching," in *CVPR*, 2017. 3, 5, 9

[30] F. Moreno-Noguer, " 3D human pose estimation from a single image via distance matrix regression," in *CVPR*, 2017. 3, 9

[31] B. Xiaohan Nie, P. Wei, and S.-C. Zhu, "Monocular 3D human pose estimation by predicting depth on joints," in *ICCV*, 2017. 3, 9

[32] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3D human pose estimation," in *ICCV*, 2017. 3, 9

[33] H. Yasin, U. Iqbal, B. Krüger, A. Weber, and J. Gall, "A dual-source approach for 3D pose estimation from a single image," in *CVPR*, 2016. 3, 5, 6, 9, 10

[34] M. Lin, L. Lin, X. Liang, K. Wang, and H. Cheng, "Recurrent 3D pose sequence machines," in *CVPR*, 2017. 3

[35] A. Agarwal and B. Triggs, "3D Human Pose from Silhouettes by Relevance Vector Regression," in *CVPR*, 2004. 3

[36] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. H. S. Torr, "Randomized trees for human pose detection," in *CVPR*, 2008. 3

[37] C. Sminchisescu, A. Kanaujia, Z. Li, and D. N. Metaxas, "Generative

Modeling for Continuous Non-Linearly Embedded Visual Inference," in *CVPR*, 2005. 3

[38] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Fast Algorithms for Large Scale Conditional 3D Prediction," in *CVPR*, 2008. 3

[39] G. Shakhnarovich, P. Viola, and T. Darrellr, "Fast pose estimation with parameter-sensitive hashing," in *CVPR*, 2003. 3

[40] S. Li, W. Zhang, and A. B. Chan, "Maximum-margin structured learning with deep networks for 3D human pose estimation," in *ICCV*, 2015. 3, 6, 9

[41] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua, "Structured Prediction of 3D Human Pose with Deep Neural Networks," in *BMVC*, 2016. 3

[42] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis., " Coarse-to-fine volumetric prediction for single-image 3D human pose," in *CVPR*, 2017. 3, 9

[43] X. Zhou, M. Zhu, S. Leonardos, K. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3D human pose estimation from monocular video," in *CVPR*, 2016. 3, 6, 9

[44] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression," in *ICCV*, 2017. 3, 9

[45] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer, "A joint model for 2D and 3D pose estimation from a single image," in *CVPR*, 2013. 3

[46] F. Zhou and F. D. la Torre, "Spatio-temporal matching for human detection in video," in *ECCV*, 2014. 3

[47] B. Tekin, P. Marquez-Neila, M. Salzmann, and P. Fua, "Learning to fuse 2D and 3D image cues for monocular body pose estimation," in *ICCV*, 2017. 3, 9

[48] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3D pose estimation from a single image," in *CVPR*, 2017. 3, 9

[49] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3D human pose estimation in the wild using improved cnn supervision," in *3DV*, 2017. 3, 9

[50] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Towards 3D human pose estimation in the wild: A weakly-supervised approach," in *ICCV*, 2017. 3, 9

[51] C. R. de Souza, A. Gaidon, Y. Cabon, and A. Lopez, "Procedural Generation of Videos to Train Deep Action Recognition Networks," in *CVPR*, 2017. 3

[52] S. Huang and D. Ramanan, "Expecting the Unexpected: Training Detectors for Unusual Pedestrians With Adversarial Imposters," in *CVPR*, 2017. 3

[53] G. Varol, J. Romero, X. Martin, N. Mahmood, M. Black, I. Laptev, and C. Schmid, " Learning From Synthetic Humans," in *CVPR*, 2017. 3, 7

[54] G. Rogez and C. Schmid, "Image-based synthesis for deep 3d human pose estimation," *to appear in IJCV*, 2018. 3, 9

[55] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler, "Unite the people: Closing the loop between 3D and 2D human representations," in *CVPR*, 2017. 3

[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015. 5

[57] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009. 5

[58] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state-of-the-art analysis," in *CVPR*, 2014. 6, 9

[59] I. Kostrikov and J. Gall, "Depth sweep regression forests for estimating 3D human pose from images," in *BMVC*, 2014. 6, 9

[60] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics*, 2015. 7

[61] L. Bo and C. Sminchisescu, "Twin Gaussian processes for structured prediction," *IJCV*, 2010. 9

[62] Y. Du, Y. Wong, Y. Liu, F. Han, Y. Gui, Z. Wang, M. Kankanhalli, and W. Geng, "Marker-less 3D human motion capture with monocular image sequence and height-maps," in *ECCV*, 2016. 9

[63] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei, "Deep kinematic pose regression," in *ECCV Workshop on Geometry Meets Deep Learning*, 2016. 9

[64] M. Sanzari, V. Ntouskos, and F. Pirri, "Bayesian image based 3D pose estimation," in *ECCV*, 2016. 9

[65] I. Katircioglu, B. Tekin, M. Salzmann, V. Lepetit, and P. Fua, "Learning latent representations of 3D human pose with deep neural networks," *IJCV*, 2018. 9

[66] S. Kinauer, R. Guler, S. Chandra, and I. Kokkinos, "Structured output prediction and learning for deep monocular 3d human pose estimation," in *EMMCVPR*, 2017. 9

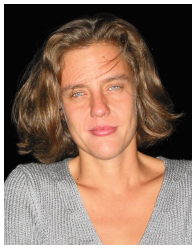
[67] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *ECCV*, 2014. 9, 10

[68] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation MoCap database HDM05," Universität Bonn, Tech. Rep. CG-2007-2, 2007. 10

[69] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," in *NIPS*, 2017. 12

Grégory Rogez holds a M.Eng. in physics from the Ecole Nationale Sup. de Physique de Marseille (ENSPM, now Centrale Marseille), France, a M.Sc. degree in biomedical engineering and a Ph.D. degree in computer vision both from the University of Zaragoza, Spain. Dr. Rogez was a visiting student (2007-2008) and research fellow (2009-2010) in the Computer Vision group of Oxford Brookes University. His work on monocular human body pose analysis received the (biennial) best Ph.D. thesis award from the Spanish Association on Pattern Recognition (AERFAI) in 2013. Dr. Rogez was awarded a competitive Marie Curie Fellowship to visit the University of California between 2013 and 2015. He is currently a Research Scientist with the THOTH team at Inria Grenoble Rhône-Alpes. His research interests include computer vision and machine learning, with a special focus on understanding people from visual data, *i.e.*, human detection, 2D/3D pose estimation, action recognition and object manipulation.

Philippe Weinzaepfel received a M.Sc. degree from Université Grenoble Alpes, France, and Ecole Normale Supérieure de Cachan, France, in 2012. He was a PhD student in the Thoth team, at Inria Grenoble and LJK, from 2012 until 2016, and received a PhD degree in computer science from Université Grenoble Alpes in 2016. He is currently a Research Scientist at NAVER LABS Europe, France, in the computer vision group. His research interests include computer vision and machine learning, with special interest in video understanding and action recognition.



Cordelia Schmid Cordelia Schmid holds a M.S. degree in computer science from the University of Karlsruhe and a Doctorate, also in computer science, from the Institut National Polytechnique de Grenoble (INPG). Her doctoral thesis received the best thesis award from INPG in 1996. Dr. Schmid was a post-doctoral research assistant in the Robotics Research Group of Oxford University in 1996–1997. Since 1997 she has held a permanent research position at Inria Grenoble Rhône-Alpes, where she is a research director and directs an INRIA team. Dr. Schmid has been an Associate Editor for IEEE PAMI (2001–2005) and for IJCV (2004–2012), editor-in-chief for IJCV (2013—), a program chair of IEEE CVPR 2005 and ECCV 2012 as well as a general chair of IEEE CVPR 2015 and ECCV 2020. In 2006, 2014 and 2016, she was awarded the Longuet-Higgins prize for fundamental contributions in computer vision that have withstood the test of time. She is a fellow of IEEE. She was awarded an ERC advanced grant in 2013, the Humbolt research award in 2015 and the Inria & French Academy of Science Grand Prix in 2016. She was elected to the German National Academy of Sciences, Leopoldina, in 2017.