PCLs: Geometry-aware Neural Reconstruction of 3D Pose with Perspective Crop Layers

Frank Yu[1] Mathieu Salzmann[2] Pascal Fua[2] Helge Rhodin[1]

[1]UBC, Vancouver, Canada
[2]EPFL, Lausanne, Switzerland
{frankyu, rhodin}@cs.ubc.ca

Abstract

Local processing is an essential feature of CNNs and other neural network architectures—it is one of the reasons why they work so well on images where relevant information is, to a large extent, local. However, perspective effects stemming from the projection in a conventional camera vary for different global positions in the image. We introduce Perspective Crop Layers (PCLs)—a form of perspective crop of the region of interest based on the camera geometry— and show that accounting for the perspective consistently improves the accuracy of state-of-the-art 3D pose reconstruction methods. PCLs are modular neural network layers, which, when inserted into existing CNN and MLP architectures, deterministically remove the location-dependent perspective effects while leaving end-to-end training and the number of parameters of the underlying neural network unchanged. We demonstrate that PCL leads to improved 3D human pose reconstruction accuracy for CNN architectures that use cropping operations, such as spatial transformer networks (STN), and, somewhat surprisingly, MLPs used for 2D-to-3D keypoint lifting. Our conclusion is that it is important to utilize camera calibration information when available, for classical and deep-learning-based computer vision alike. PCL offers an easy way to improve the accuracy of existing 3D reconstruction networks by making them geometry-aware. Our code is publicly available at github.com/yu-frank/PerspectiveCropLayers.

1. Introduction

Convolutional neural networks (CNNs) have proven highly effective for image-based prediction tasks because of their translation invariance and the locality of the computation they perform. For 3D pose estimation, this allows them to focus on image locations that carry information about the pose while discarding other ones [43, 30, 6, 27, 32, 39, 44, 42, 20, 51, 17, 47].

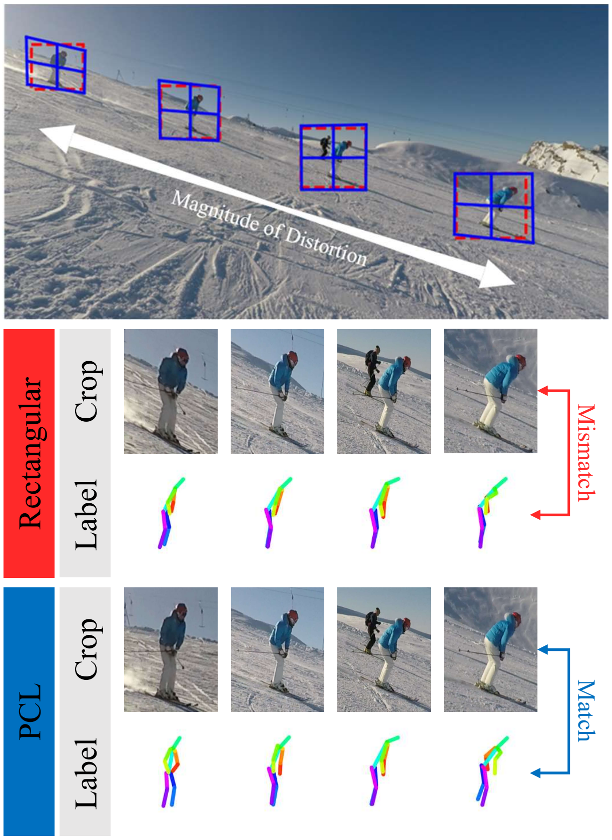

Figure 1: **Perspective effects and correction with PCL crops.** The skier looks as if she turns, from front to backwards facing, although recorded with a static camera and going straight. PCLs correct the stretching originating from the projection onto the image plane and matches the 3D pose label to the local view direction of the crop.

Convolutions in the image plane, however, ignore the perspective effects caused by projecting a 3D scene in 2D. For example, as shown in Figure 1, a person captured by static camera in a fixed pose and moving in a constant direction is seen from different angles as their image location changes. Applying the same convolutional filter at the top-left image corner and at the bottom-right one will therefore yield different features, even though the pose is the same. In practice, this is typically tackled by increasing the width and depth of the network, so that different filters and layers can model the same 3D pose perspective-distorted in different ways. The effectiveness of this procedure, however, strongly depends on the availability of large amounts of training data, which is far from being a given for pose estimation in the wild. Notably, two-stage approaches that lift 3D pose from 2D pose estimates using multilayer perceptrons (MLPs) [2, 28, 8, 23, 22, 34, 31, 26] rely also on translational invariance by centering the 2D pose on a root joint, thereby losing important cues on perspective distortion too.

In this paper, we therefore introduce Perspective Crop Layers (PCLs) to explicitly account for perspective distortion within CNNs and other neural networks. Specifically, we use a homography to map the input image to a virtual camera with pre-defined intrinsic parameters that point to the region of interest (RoI). The homography parameters are functions of the RoI's location and scale. Hence, this yields a synthetic view in which the location-dependent perspective deformations are undone. The 3D pose inferred from this synthetic view can then be projected into the original image. This requires *a priori* knowledge about the intrinsic camera parameters, which is rarely a problem in real-world situations because they either are readily available from the camera specifications or can be inferred from the input images alone [45, 11]. We will further show that our PCLs are robust to calibration inaccuracies. Ultimately, all the operations performed by our PCLs are differentiable, and thus amenable to end-to-end learning, while removing the need for the CNN to learn the already known perspective geometry. Our contributions can be summarized as follows:

- We showcase the influence of perspective effects on 3D pose estimates that increases for poses away from the image center, which is disregarded by virtually all state-of-the-art algorithms;
- We derive the equations to compensate for these effects across the image in a location-dependent manner;
- We encapsulate our formalism into generic NN layers, dubbed PCLs, that naturally integrate into existing deep learning frameworks.

We demonstrate the benefits of our PCLs for 3D human pose estimation of both rigid objects and articulated people. PCLs yield a consistent boost in performance, of $2 - 10\%$ on average and up to 25% at the image boundary where perspective effects are strongest. Notably, the improvements attributable to our PCLs are consistent across the baseline we seek to improve, which validates our claim that even the most-advance deep networks do not learn these perspective effects on the existing datasets. This includes a PCL variant that undoes the perspective effect on 2D keypoints, thus allowing us to showcase the benefits of our approach on state-of-the-art 3D pose estimation methods that lift 2D keypoint detections to 3D poses [22, 33]. Our code is publicly available at github.com/yu-frank/PerspectiveCropLayers.

2. Related Work

In this section, we discuss existing ways of handling image distortions and review the existing attention window mechanisms upon which PCLs are built.

Handling perspective effects. Many works sidestep perspective effects by training and testing on synthetic renderings [1, 7, 49, 48] or real images [10, 49] where the object of interest is centered manually. However, these methods are not applicable to natural images where the object can be at an arbitrary location. If the object location is known in advance, perspective distortion can be undone in a preprocessing stage. For instance, [24] propose to rotate locally inferred 3D poses back to the camera frame. This strategy has later been adopted by [14], but neither of these works undistorts the input images or input 2D pose. [38, 37] apply an image correction, however, only approximating the homography with an affine transformation. In other words, the above-mentioned approaches neither model the perspective correction geometrically accurately nor formulate it as a differentiable layer. However, differentiability is an important prerequisite for end-to-end training on natural images, particularly for unsupervised approaches, that deal with unknown object locations.

Radial undistortion. Fisheye cameras and others with a large field of view yield large deformations when mapped to a rectangular pixel grid. They are better represented with spherical images, thereby avoiding location-dependent deformation entirely. This, however, gives rise to challenges when one wants to process the resulting non-rectangular pixel grids with convolutions. [3] compute convolutions on spherical harmonics, but such frequency-domain networks do not yet reach the accuracy of regular CNNs. A common workaround is to unfold the spherical images along the azimuth and longitude dimensions, which leads to lesser artifacts than perspective projection to a planar image. Nevertheless, extreme stretching at the sphere poles remains. This deformation has been handled by learning filters that have the same response as processing local planar patches [40] or by using Deformable Convolutional Networks [5]. However, any convolution is location invariant and misses the geometric position that caused the deformation. To counteract this, [19] propose to add the pixel coordinates as an input feature. In our preliminary experiments, however, we observed this to lead to overfitting and degraded results.

Convolution can be defined directly on the sphere, by sampling points reflecting the sphere curvature [15, 4]. This leads to high accuracy and position invariance, but is computationally expensive because non-regular convolution kernels are needed at each neural network layer, which hampers parallelization and cache efficiency. By contrast, we target a single undistortion layer that works in harmony with regular CNNs, MLPs, and attention mechanisms.

Attention windows. Processing RoIs instead of the entire input image leads to computationally more efficient and more accurate models. Most prominent and related to our approach are Spatial Transformer Networks (STN) [13] that learn invariance to translation, scale, rotation and more generic warping by spatially transforming the feature maps with an affine or free-form deformation. Multiple STNs have also been stacked [18] to model more complex transformations. STNs proceed in two steps: First, a grid of sample points is defined in the original image, either by direct regression or by predicting the parameters of a restricted family of transformations, such as a 3×3 matrix for affine transformations. Second, the pixel value at each grid point is mapped to the target by bilinear interpolation of the neighboring image pixels. This yields differentiability and enables end-to-end training as an ordinary layer within deep network architectures. It also applies to 3D transformations [48]. In this work, we generate a sampling grid that undoes perspective effects in the RoI and use the STN to maintain differentiability with respect to the RoI position and scale.

3. Perspective Crop Layer

We start our derivation by formulating local processing and existing cropping solutions mathematically as an affine transformation between a real and virtual camera of fixed orientation. Subsequently, we derive the perspective transformation underlying PCL, which corresponds to a rotation of the virtual camera frame, and finally introduce the implementation of PCL via two neural network layers that sandwich the backbone prediction network.

3.1. Motivation and Rectangular Crops

Each point $\hat{\mathbf{q}}$ of a rescaled rectangular or trapezoidal image patch can be expressed in terms of the original image coordinates $\mathbf{q}$. This affine transformation can be written in projective coordinates as

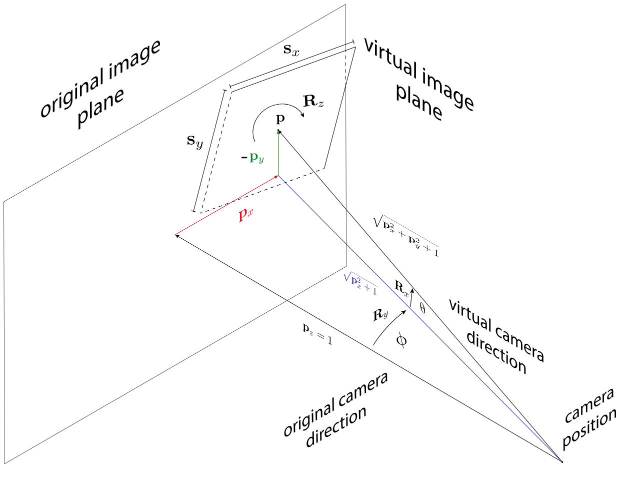

Figure 2: **Virtual camera.** Underlying to PCL is the projection from the original image plane onto a virtual camera pointing at the crop location. This figure visualizes the various quantities needed to infer the mapping.

$$\hat{\mathbf{q}} = \mathbf{C}\mathbf{q} \ , \quad \text{with } \mathbf{C} = \begin{bmatrix} s_x & c_x & a_x \\ c_y & s_y & a_y \\ 0 & 0 & 1 \end{bmatrix} , \tag{1}$$

where $\mathbf{a} = [a_x, a_y]$ defines a 2D translation, $\mathbf{s} = [\mathbf{s}_x, \mathbf{s}_y]$ are scalings in two different directions, and $\mathbf{c} = [c_x, c_y]$ are skew parameters. Therefore, as shown in Fig. 2, a cropped image can be thought of as being taken by a virtual camera with intrinsic parameters $\mathbf{K}^{\text{virt}} = \mathbf{CK}$, where $\mathbf{K}$ is the true 3×3 matrix of intrinsic parameters. As the translation $\mathbf{a}$ is usually chosen so that the patch contains an object of interest, the optical center of the virtual camera depends on the target location, which means that objects projected far from the image center are deformed differently from those near it, as shown in Fig. 1. To remedy this, our goal is to design a crop operation such that the optical center of the virtual camera is always at the center of the patch, which makes perspective distortion independent from image location.

The centering of 2D human pose commonly done in the state-of-the-art 2D-to-3D lifting approaches is a form of rectangular cropping, too. A pose is root-centered by multiplication with an affine matrix $\mathbf{C}$, in which $\mathbf{a}$ is the pelvis/root position, $\mathbf{s} = 1$ and $\mathbf{c} = 0$. The subsequent root-centered processing with an MLP has the same downsides as cropping in STNs and convolution in CNNs in that information about the image location is removed while being affected by position-dependent perspective effects.

3.2. Defining a Virtual Camera

We introduce a virtual camera with the same optical center as the real one but whose optical axis points at the center of the target patch $\mathbf{p} = [\mathbf{p}_x, \mathbf{p}_y]$ and whose focal length is chosen to zoom onto the region of interest with factor $\mathbf{s} = [\mathbf{s}_x, \mathbf{s}_y]$, as shown in Fig. 2. Below, we derive the virtual extrinsic parameters, in the form of rotation matrix $\mathbf{R}_{\text{virt}\rightarrow\text{real}}$, and virtual intrinsic parameter matrix, $\mathbf{K}^{\text{virt}}$, such that these constraints are fulfilled.

Cropping as a change of camera perspective. We aim to find camera parameters such that mapping a pixel from the original image to the cropped patch can be done by multiplying an image coordinate in homogeneous coordinates, $(u, v, 1)^\top$, by the matrix

$$\mathbf{\Gamma}(u, v, \mathbf{s}, \mathbf{K}) = \mathbf{K}^{\text{virt}} \mathbf{R}^{-1}_{\text{virt}\rightarrow\text{real}} \mathbf{K}^{-1} \ . \tag{2}$$

This transformation undoes the original projection using $\mathbf{K}^{-1}$, rotates the resulting point to the virtual camera with $\mathbf{R}^{-1}_{\text{virt}\rightarrow\text{real}}$ and projects it using $\mathbf{K}^{\text{virt}}$. As for the typical rectangular cropping defined in Eq. 1, mapping from the image to the patch remains a warp and does not depend on the generally unknown scene geometry. By contrast to rectangular cropping, this warp is non-linear.

Extrinsic Parameters. Let $\mathbf{R}_{\text{virt}\rightarrow\text{real}}$ be the 3×3 rotation matrix that defines the virtual camera orientation. It can be written as $\mathbf{R}_y \mathbf{R}_x \mathbf{R}_z$, where $\mathbf{R}_x$, $\mathbf{R}_y$, and $\mathbf{R}_z$ are the Euler rotation matrices that rotate counter-clockwise around the x, y, and z axes of the original camera coordinate system, as depicted by Fig. 2. Two degrees of freedom of $\mathbf{R}_{\text{virt}\rightarrow\text{real}}$, $\mathbf{R}_x$ and $\mathbf{R}_y$, are determined by pinning the center of the virtual camera to the backprojected point $\mathbf{p} = \mathbf{K}^{-1}(u, v, 1)^\top$ in the real camera. Formally, we compute

$$\mathbf{R}_{\text{virt}\rightarrow\text{real}} = \begin{bmatrix} \frac{1}{\sqrt{1+\mathbf{p}_x^2}} & \frac{-\mathbf{p}_x\mathbf{p}_y}{\sqrt{(1+\mathbf{p}_x^2+\mathbf{p}_y^2)(1+\mathbf{p}_x^2)}} & \frac{\mathbf{p}_x}{\sqrt{1+\mathbf{p}_x^2+\mathbf{p}_y^2}} \\ 0 & \frac{\sqrt{1+\mathbf{p}_x^2}}{\sqrt{1+\mathbf{p}_x^2+\mathbf{p}_y^2}} & \frac{\mathbf{p}_y}{\sqrt{1+\mathbf{p}_x^2+\mathbf{p}_y^2}} \\ \frac{-\mathbf{p}_x}{\sqrt{1+\mathbf{p}_x^2}} & \frac{-\mathbf{p}_y}{\sqrt{(1+\mathbf{p}_x^2+\mathbf{p}_y^2)(1+\mathbf{p}_x^2)}} & \frac{1}{\sqrt{1+\mathbf{p}_x^2+\mathbf{p}_y^2}} \end{bmatrix}, \tag{3}$$

The details are provided in the appendix.

The yaw angle around the optical axis is unconstrained. We set it to zero (pointing upwards) in our experiments. Instead, $\mathbf{R}_z$ could be controlled, to normalize subject orientation to pose human subjects upright in the virtual view.

Intrinsic Parameters. Let

$$\mathbf{K}^{\text{virt}} = \begin{bmatrix} \mathbf{f}_x^{\text{virt}} & 0 & t_x^{\text{virt}} \\ 0 & \mathbf{f}_y^{\text{virt}} & t_y^{\text{virt}} \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

be the 3×3 matrix of intrinsic parameters of the virtual camera. Putting the optical center in the middle of the patch means that $t_x^{\text{virt}} = t_y^{\text{virt}} = 0.5$. The virtual focal lengths are $\mathbf{f}^{\text{virt}} = [\mathbf{f}_x^{\text{virt}}, \mathbf{f}_y^{\text{virt}}] = \frac{\mathbf{h}^{\text{virt}}}{\mathbf{s}}$, where $\mathbf{h}^{\text{virt}}$ is a function of the original focal length in the horizontal and vertical direction stored in $\mathbf{K}$, and $\mathbf{s}$ determines the crop scale in relation to the full image. Together with $\mathbf{p}$, it defines the area of interest and is the input to PCL.

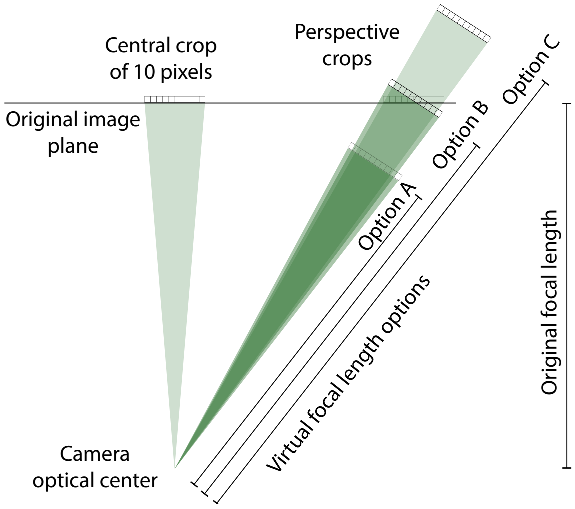

Figure 3: **Focal length settings.** We propose three different variants for inferring the focal length of the camera to match the pixel scales in the original and transformed view. The intersection of the green cones with the image plane is the fraction of pixels that is cropped by PCL. Only Option C maintains a consistent scaling between center and off-center positions.

There is no universal way for choosing $\mathbf{h}^{\text{virt}}$, the virtual camera's focal length, without scaling to the smaller crop size. We propose the following three alternatives and evaluate their influence empirically in Section 4:

A. Setting $\mathbf{h}^{\text{virt}}$ to $\mathbf{f}$, the original focal length.

B. Setting $\mathbf{h}^{\text{virt}}$ to $\mathbf{f}\|\mathbf{p}\|$, so that the virtual image plane intersects with the real one at $\mathbf{p}$.

C. Setting $\mathbf{h}_x^{\text{virt}} = \mathbf{f}_x \|\mathbf{p}\| \sqrt{\mathbf{p}_x^2 + 1}$ and $\mathbf{h}_y^{\text{virt}} = \mathbf{f}_y \frac{\|\mathbf{p}\|^2}{\sqrt{\mathbf{p}_x^2+1}}$ to preserve pixel scales.

Although the nature of such a perspective transformation cannot maintain scale in all parts of the image, the last choice of parameters guarantees that scale between the original and virtual image is preserved along the vertical and horizontal axis, while scaling non-linearly in the diagonal direction. Fig. 1 visualizes this behavior, and Fig. 3 illustrates the crop width after with each of the three choices.

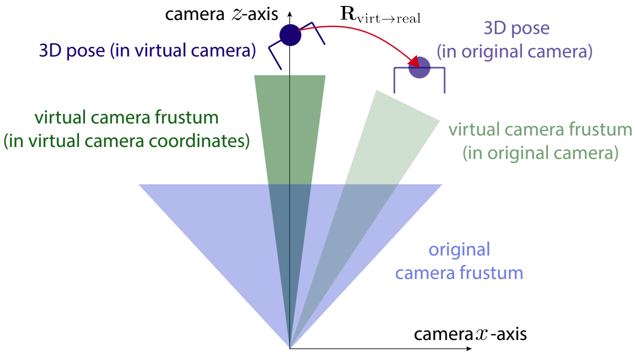

Figure 4: **3D Pose change.** Because networks equipped with a PCL layer operate in a virtual camera, the 3D pose prediction living in these coordinates is transformed back from virtual to original camera coordinates in the PCL_inv layer by applying $\mathbf{R}_{\text{virt}\to\text{real}}$.

3.3. Differentiable Network Layer

PCLs are designed to facilitate perspective correction within existing deep neural network architectures. We propose the two forms that are depicted in Fig. 5. Two layers are involved, the projection to the virtual camera and the transformation of the reconstruction to the original camera.

PCL for lifting 2D keypoints to 3D with MLPs. For networks taking 2D keypoints as input, such as the locations of the human body parts detected in the image plane, Eq. 2 can be applied directly on every 2D coordinate and becomes a simple pre-processing that normalizes the 2D pose for perspective effects. The target center location $\mathbf{p}$ can be chosen as the mean of all joints, or a root joint. We use the pelvis location as crop target for human pose estimation.

PCL for CNNs. Applying PCL to CNNs requires a two-stage CNN architecture. First, one or more RoIs $(\mathbf{p}, \mathbf{s})_{i=1}^{N}$ are predicted from the input image $\mathbf{I} \in \mathbb{R}^{W \times H \times F}$ using a detection network. Subsequently, the input is cropped to focus the attention of the subsequent reconstruction network. PCL replaces the cropping by implementing Eq. 2 and Eq. 3. The pixels are warped using bilinear interpolation, as in the conventional STNs [13] we introduced in the related work section. Because the transformation $\mathbf{\Gamma}$, the definition of $\mathbf{R}_{\text{virt}\to\text{real}}$, and the virtual camera matrix $\mathbf{K}^{\text{virt}}$ rely on simple algebraic operations, the entire process is analytically differentiable. To improve efficiency and numerical stability, we parametrize $\mathbf{R}_{\text{virt}\to\text{real}}$ in terms of length measures (Eq. 3) instead of angles and computationally-expensive trigonometric functions in the general definition of Euler angles. The derivation and relation of both are detailed in the appendix.

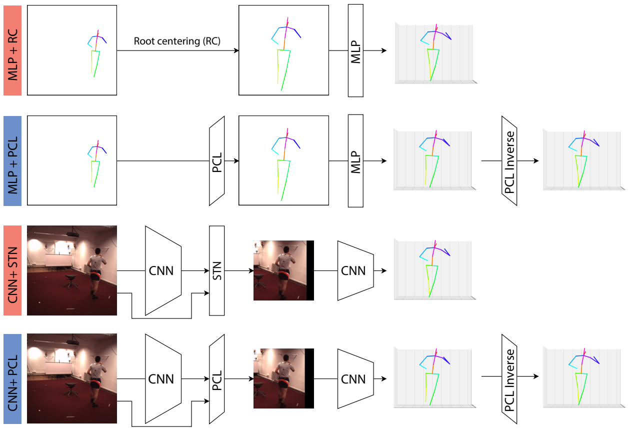

Figure 5: **PCL integration into existing architectures.** PCLs are applied in pairs, sandwiching the original neural network backbone with the PCL mapping from original to virtual image space and the PCL_inv that maps 3D predictions back to the original camera coordinates.

PCL_inv: back-transformation to the real camera. The derived perspective crop has the regular grid structure of a normal image or feature map. Any neural image processing steps can be performed on it as is. However, the derived quantities will live in the coordinates of the virtual camera. As for classical attention windows, if the spatial context is important, the processed crop needs to be translated back to the original camera coordinates. For 3D quantities, such as 3D human pose, this amounts to applying $\mathbf{R}_{\text{virt}\to\text{real}}$ on the reconstruction, as shown in Fig. 4. This PCL_inv layer is the same for MLPs and CNNs.

4. Experiments

We evaluate the improvements brought about by PCL on the task of 3D human pose estimation from either images or 2D keypoints, and show that they hold for neural networks of diverse complexity. The benefits of PCL for the 2D to 3D lifting task on Human 3.6 Million dataset [12] and MPI-INF-3DHP dataset [25] are shown qualitatively in both Fig. 6 and in additional experiments in the supplemental video.

Baselines. We integrate PCL into the three neural network architectures for 3D pose estimation discussed below, and compare the resulting networks with the original ones.

MLP+RC: As a first baseline, we use the four-layer MLP from [21] with root centering. To ensure a fair comparison, we scale the 2D input of the baseline by the crop scales $\mathbf{s}$ that are used in PCL.

T-CNN: Our second baseline consists of the temporal convolution 2D to 3D lifting approach of Pavllo et al. [33],

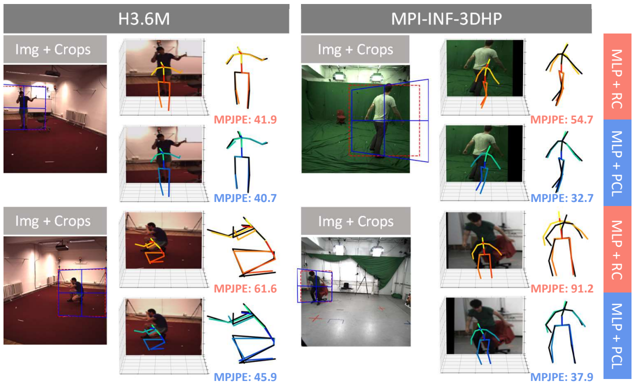

Figure 6: **Qualitative examples.** For both, H3.6M (left) and MPI-INF-3DHP (right), PCL improves 3D pose estimation significantly by predicting the orientation of limbs more precisely. The MLP+PCL output is shown in blue and the baseline w/o PCL in red. Individual MPJPE scores (in mm) are reported for each, which relates the visual and quantitative improvements.

which operates on pose sequences. To date, it is the most accurate method in its class.

CNN+STN: ResNet [9] is the most widely used backbone for predicting 3D pose from images. As baseline, we use an STN that takes a 265 × 265 image as input and outputs a 128 × 128 patch. We do almost all tests with ground-truth crop locations determined by bounding box annotations in the dataset. This guarantees that the differences in performance that we measure during evaluation are entirely due to the type of cropping and not differences in a detector network. We test a version with an an additional L2 loss on the crop location predicted by a ResNet-18 detector network, trained end-to-end with the 3D reconstruction objective. We test ResNet backbones of depth 18 and 50.

Each baseline is extended with PCLs as follows, and as sketched in Fig. 5.

MLP+PCL: Our method replaces the traditional hip-centering in [21] with the PCL and PCL_inv layers.

T-CNN+PCL: We apply PCL to [33] by transforming the sequence of frames to the virtual camera pointing to the image coordinates in the middle of the sequence. Note that this centering is a significant difference to the original [33], which works with absolute 2D positions as input. For simplicity, we assume the optical center is at the image center.

CNN+PCL: We simply replace the rectangular STN crop with our PCL layer, as detailed in Section 3.3.

Datasets. **H3.6M**: We evaluate the effectiveness of PCL on the popular Human 3.6 Million dataset [12] that features eleven subjects performing 14 different actions and provides ground-truth 3D poses and camera calibration. We use the established train/validation/test split, 17-joint skeleton, and the pre-processing of [29]. We set the rectangular and PCL crop location to the pelvis 2D joint and compute the crop scale as the width and height of a tightly-fitting bounding box. We also experiment with using the GT depth for scale estimation. We compare variants using 2D detections from [41, 46] and ground truth as input for 2D to 3D lifting. When we compare to [33], we use their preprocessing and train/validation/test split since consecutive frames are required. For simplicity, we assume that the image size is 1000 × 1000, although size varies from 1000 to 1002.

MPI-INF-3DHP: We also evaluate our approach on the MPI-INF-3DHP dataset [25], which, compared to H3.6M, contains more extreme poses, outdoor environments, and is shot with wide field-of-view cameras, leading to stronger perspective effects. The cameras are calibrated, and all frames are labeled with 3D pose. We use the color augmentation from [29] and the official test set and training subjects 1-8 for training, while withholding the first sequence of subject 4 and the last sequence of subject 8 for validation. We set the rectangular and PCL crop location to the pelvis joint for 2D to 3D lifting and at the mean of the 2D poses for the image-based variants. The crop scale is computed as the width and height of a tightly fitting bounding box.

ToyCube: We introduce a synthetic dataset containing images of a rendered cube of edge length 0.5 m. We use this toy example to ablate individual factors of variation, such as the effect of illumination and pose distribution.

Training setup. The 3D pose is trained on an L2 loss using Adam [16] with a learning rate of 0.001 for the 2D to 3D lifting models and 0.0005 for the image to 3D networks. The temporal convolution networks are trained using Amsgrad [36] with an initial learning of 0.001 and a learning rate decay factor of 0.95 applied after each epoch. We train 2D to 3D lifting methods for 200 epochs and batch size 64 and the temporal convolution networks for 80 epochs and batch size 1024 with up to 243 frames in each batch element. Lastly, image to 3D networks using a ResNet-50 backbone are trained for 40 and 150 epochs on H3.6m and MPI-INF-3DHP respectively. The ResNet-18 backbone trained on H3.6M is trained for 60 epochs.

Metrics. To quantitatively evaluate 3D pose accuracy, we use the Mean Per Joint Position Error (MPJPE), computed as the average Euclidean distance of the predicted 3D joints to the ground-truth ones, where both poses are centered at the pelvis. All MPJPE results are reported in millimeters. We also report the percentage of correct keypoints (PCK), encoding the proportion of joints whose distance to the ground truth is less than a threshold, using thresholds of 50 and 100 millimeters.

Input	Model	H3.6M MPJPE ↓	 PCK @ 50mm ↑	 PCK @ 100mm ↑	MPI-INF-3DHP MPJPE ↓	 PCK @ 50mm ↑	 PCK @ 100mm ↑
2D GT + 3D Root GT	MLP + RC ([21])	45.3 ± 0.2	66.8 ± 0.3	91.4 ± 0.2	69.4 ± 0.4	46.4 ± 0.5	77.0 ± 0.3
2D GT + 3D Root GT	MLP + PCL (Ours)	$\mathbf{40.1 \pm 0.5}$	$\mathbf{72.8 \pm 0.5}$	$\mathbf{93.3 \pm 0.2}$	$\mathbf{45.6 \pm 0.5}$	$\mathbf{70.1 \pm 0.5}$	$\mathbf{89.7 \pm 0.3}$
2D GT	MLP + RC ([21])	48.4 ± 0.4	62.9 ± 0.5	90.3 ± 0.2	74.1 ± 0.5	43.0 ± 0.3	74.7 ± 0.7
2D GT	MLP + PCL (Ours)	$\mathbf{43.8 \pm 0.1}$	$\mathbf{68.3 \pm 0.1}$	$\mathbf{92.2 \pm 0.0}$	$\mathbf{50.1 \pm 0.2}$	$\mathbf{65.5 \pm 0.2}$	$\mathbf{87.8 \pm 0.1}$
2D Detection	MLP + RC ([21])	69.7 ± 0.2	46.3 ± 0.5	80.5 ± 0.1	-	-	-
2D Detection	MLP + PCL (Ours)	$\mathbf{67.0 \pm 0.1}$	$\mathbf{48.7 \pm 0.2}$	$\mathbf{82.1 \pm 0.0}$	-	-	-
Image	CNN (ResNet50) + STN	96.5	32.7	64.6	117.7	33.6	60.1
Image	CNN (ResNet50) + PCL (Ours)	**94.1**	**34.1**	**65.8**	**109.5**	**40.3**	**66.2**
Image	CNN (ResNet18) + STN	95.9	35.4	65.6	-	-	-
Image	CNN (ResNet18) + PCL (Ours)	**93.9**	**37.1**	**66.6**	-	-	-

Table 1: Shown are the reported MPJPE in millimeters as well as the PCK for 2D to 3D keypoint lifting tests performed on H3.6M. The reported mean and standard deviation is computed over three runs with varying random seed. For MPJPE, lower values are better and for PCK, higher values are better. We can see from the table that our method significantly outperforms the baselines that do not use PCL. We bold the best performing models in each category.

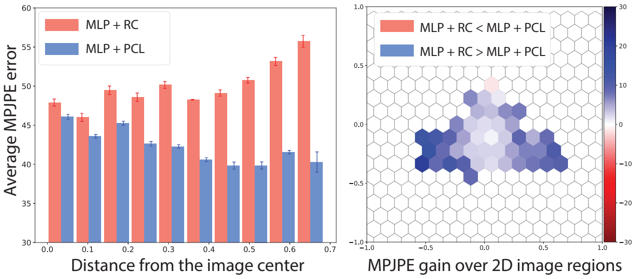

Figure 7: **Improvement of reconstruction error,** binned with respect to the image position. **Left:** MLP+RC suffers from perspective effects away from the center, while MLP+PCL effectively compensates these leading to improvements of up to 25%. **Right:** The consistent difference of MLP+RC and MLP+PCL is also reflected over a 2D tiling, showing the average MPJPE error difference of cells with 10 or more frames on the validation set.

Model	Original f	New f
T-CNN	$\mathbf{47.3 \pm 0.0}$	72.7 ± 0.5
T-CNN + RC	51.5 ± 0.1	51.5 ± 0.1
T-CNN + PCL (known f)	48.8 ± 0.3	$\mathbf{48.9 \pm 0.2}$

Table 2: **Temporal CNN tests**, computed as the MPJPE over two runs with varying seed on H3.6M. While the baseline performs the best using the original camera, it is unable to generalize to new camera settings. The PCL equipped version strikes the best compromise.

4.1. PCL for 2D to 3D Keypoint Lifting

The results for the 2D to 3D lifting task on H3.6M and MPI-INF-3DHP are provided in Table 1. For H3.6M, MLP+PCL achieves an MPJPE of 67.0 mm vs. 69.8 mm MPJPE of the MLP+RC baseline [21] when using 2D detections from [41, 46] as input, a 4% improvement. Even larger improvements are achieved when using the GT 2D pose as input and when using the 3D root joint position for scale estimation. Notably, our method with a scale computed from the 2D pose still outperforms the STN baseline using 3D ground truth for scale prediction.

We obtain even larger improvements with PCL on the MPI-INF-3DHP dataset, with 2.4 cm in MPJPE and 22 PCK points. This dramatic improvement is no surprise since the larger field of view (smaller focal length f) of the MPI-INF-3DHP cameras leads to stronger perspective effects and, therefore, a larger difference between the corrected and uncorrected views. These experiments also show that PCL is not specific to any particular focal length.

In Figure 7, we analyze the position dependent effect of our method on H3.6M. As shown by the plot, the baseline MPJPE increases with the distance from the subject to the image center, hinting at the negative effect of perspective distortion. PCLs undo this effect, leading to a more stable MPJPE and outperforming the baseline by a growing margin as the distance increases. PCL even decreases with the distance to the center, which is surprising. We believe this is because the most complex poses in H3.6M, such as sitting and lying on the ground, are performed in the image center while walking dominates for the off-center ones.

4.2. PCL for Temporal CNNs

As shown in Table 2 incorporating PCL into the temporal convolutional network of [33] does not improve results on their original implementation. Our following analysis shows that this is due to [33] already learning position-dependent effects by operating on unnormalized 2D pose. This, however, overfits to the camera used at training time. By contrast, PCL generalizes perfectly when the camera changes at test time so long as its properties are known approximately.

To analyze the effect, we artificially change the focal length of the test sequences by multiplying all 2D testing

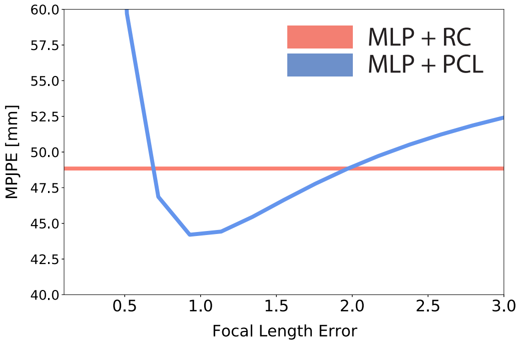

Figure 8: **Evaluating the effectiveness of PCL when the focal length of the camera is approximated.** We can see from the figure that for reasonable approximations of the focal length, PCL still outperforms the rectangular cropping.

poses by a factor of 2/3. This simulates a camera with a smaller focal length and larger field of view. T-CNN generalizes poorly, as it seemingly overfits to the global position and scale of the training set, while PCL can adapt perfectly to the new capture setup without loss in performance.

To facilitate a fair comparison, we created a second baseline, T-CNN+RC. It centers and scales the input pose sequence by subtracting the root joint of the central frame and scaling by its horizontal and vertical size; this is the same procedure that is used for the other RC baselines. This maintains the root motion while removing absolute scale and the position in the image. This variant generalizes much better than T-CNN does but has an overall higher error than T-CNN+PCL. In summary, PCL strikes the best compromise between accuracy on the original domain while being applicable to new camera geometries and capture setups.

4.3. PCL for CNN Architectures

As shown in the last four rows of Table 1, when using images as input to a ResNet regressing 3D pose on H3.6M, the baselines achieves an MPJPE of 96.5 mm, while our model with PCL yields 94.1 mm, a 2.5% reduction. The improvement is lower compared to 2D to 3D lifting, likely because the overall higher error for image to 3D pose prediction compared to 2D to 3D lifting is dominated by other error factors.

On the MPI-INF-3DHP dataset, PCLs' improvement is more pronounced, improving by 8mm and 6 PCK points, which further validates the previous findings that perspective effects are stronger on MPI-INF-3DHP, therefore leading to a clear improvement despite higher total errors.

4.4. Ablation Studies

Robustness to errors in focal length. Although PCL requires knowledge of the camera's focal length, it is often known or can be estimated approximately from image features [45, 11]. To evaluate the robustness of PCL to erroneous estimates, we experiment with an artificial disturbance to the true focal length at test time. The 2D input poses to the MLP+PCL network are deformed as if they would stem from a camera with different zoom. As shown in Fig. 8, PCL is relatively robust when the estimated focal length ranges between 0.7 and 1.5 times the true one.

Network capacity, post-processing, and generalization. In addition to the main results, we i) show that the improvement by PCL is as prominent as doubling the neural network capacity from two million to four million parameters; ii) show that it is important to apply PCL at training time instead of on pre-trained models; and iii) analyze the generalizability of PCL to new unseen positions within the camera frame on a synthetic cube dataset. Details for all these tests can be found in the supplemental document.

5. Limitations

The gained improvements come at the cost of requiring an estimate of the focal length f. Yet, the robustness towards errors in f shows that improvements can still be obtained with a rough guess. It is worth to note that PCL compensates for location-dependent perspective effects. Those effects that originate from varying distance of the object to the camera, e.g., selfie vs. third person picture, can not be resolved with image warps but would require knowledge of the 3D geometry. Data-driven approaches have been proposed to compensate these [35, 50].

6. Conclusion

We have presented a drop-in replacement for rectangular cropping and root centering that removes location-dependent perspective effects. It is fully differentiable, lends itself to end-to-end training, is efficient, does not impose additional network parameters, and the empirical evaluation demonstrates significant improvements for 3D pose estimation. Notably, the strong influence of perspective effects on the reconstruction accuracy is widely overlooked in the 3D pose reconstruction literature and these improvements are observed irrespective of the network architecture. PCL is therefore an important contribution to pushing state-of-the-art 3D reconstruction methods further.

Acknowledgements. We would like to thank Dushyant Mehta and Jim Little for valuable discussion and feedback. This work was funded in part by Compute Canada and a Microsoft Swiss JRC project. Frank Yu was supported by NSERC-CGSM.

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[2] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7035–7043, 2017.

[3] Taco Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Convolutional networks for spherical signals. *arXiv preprint arXiv:1709.04893*, 2017.

[4] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–533, 2018.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[6] Yu Du, Yongkang Wong, Yonghao Liu, Feilin Han, Yilin Gui, Zhen Wang, Mohan Kankanhalli, and Weidong Geng. Marker-less 3D human motion capture with monocular image sequence and height-maps. In *European Conference on Computer Vision (ECCV)*, pages 20–36. Springer, 2016.

[7] H. Fan, H. Su, and L. Guibas. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[8] Haoshu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6821–6828. AAAI Press, 2018.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[10] G.E. Hinton, A. Krizhevsky, and S.D. Wang. Transforming Auto-Encoders. In *International Conference on Artificial Neural Networks*, pages 44–51, 2011.

[11] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matthew Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. A perceptual measure for deep single image camera calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2354–2363, 2018.

[12] C. Ionescu, I. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[13] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[14] A. Kanazawa, S. Tulsiani, A. Efros, and J. Malik. Learning Category-Specific Mesh Reconstruction from Image Collections. *European Conference on Computer Vision*, 2018.

[15] Renata Khasanova and Pascal Frossard. Graph-based classification of omnidirectional images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 869–878, 2017.

[16] D.P. Kingma and J. Ba. Adam: A Method for Stochastic Optimisation. In *International Conference on Learning Representations*, 2015.

[17] Shichao Li, Lei Ke, Kevin Pratama, Yu-Wing Tai, Chi-Keung Tang, and Kwang-Ting Cheng. Cascaded deep monocular 3d human pose estimation with evolutionary training data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[18] Chen-Hsuan Lin and Simon Lucey. Inverse compositional spatial transformer networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2568–2576, 2017.

[19] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, pages 9605–9616, 2018.

[20] Chenxu Luo, Xiao Chu, and Alan L. Yuille. Orinet: A fully convolutional network for 3d human pose estimation. In *British Machine Vision Conference (BMVC)*, page 92, 2018.

[21] J. Martinez, R. Hossain, J. Romero, and J.J. Little. A Simple Yet Effective Baseline for 3D Human Pose Estimation. In *International Conference on Computer Vision*, 2017.

[22] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3d human pose estimation. In *International Conference on Computer Vision (ICCV)*, 2017.

[23] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *International Conference on 3D Vision (3DV)*. IEEE, 2017.

[24] 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 *International Conference on 3D Vision*, 2017.

[25] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017.

[26] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. XNect: Real-time multi-person 3D motion capture with a single RGB camera. volume 39, 2020.

[27] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel,

Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. In *ACM Transactions on Graphics*, volume 36, 7 2017.

[28] Francesc Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[29] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. 3d human pose estimation with 2d marginal heatmaps. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1477–1485. IEEE, 2019.

[30] Sungheon Park, Jihye Hwang, and Nojun Kwak. 3d human pose estimation using convolutional neural networks with 2d pose information. In *European Conference on Computer Vision (ECCV)*, pages 156–169, 2016.

[31] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7307–7316, 2018.

[32] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1263–1272, 2017.

[33] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019.

[34] Mir Rayat Imtiaz Hossain and James J. Little. Exploiting temporal information for 3d human pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

[35] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 51–66, 2018.

[36] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

[37] H. Rhodin, M. Salzmann, and P. Fua. Unsupervised Geometry-Aware Representation for 3D Human Pose Estimation. In *European Conference on Computer Vision*, 2018.

[38] H. Rhodin, J. Spoerri, I. Katircioglu, V. Constantin, F. Meyer, E. Moeller, M. Salzmann, and P. Fua. Learning Monocular 3D Human Pose Estimation from Multi-View Images. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[39] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net: Localization-Classification-Regression for Human Pose. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1216–1224, Honolulu, United States, July 2017. IEEE.

[40] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360 imagery. In *Advances in Neural Information Processing Systems*, pages 529–539, 2017.

[41] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[42] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2602–2611, 2017.

[43] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3d human pose with deep neural networks. In *British Machine Vision Conference (BMVC)*, 2016.

[44] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[45] Scott Workman, Connor Greenwell, Menghua Zhai, Ryan Baltenberger, and Nathan Jacobs. Deepfocal: A method for direct focal length estimation. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1369–1373. IEEE, 2015.

[46] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2018.

[47] Jingwei Xu, Zhenbo Yu, Bingbing Ni, Jiancheng Yang, Xiaokang Yang, and Wenjun Zhang. Deep kinematics analysis for monocular 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[48] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016.

[49] J. Yang, S.E. Reed, M.-H. Yang, and H. Lee. Weakly-Supervised Disentangling with Recurrent Transformations for 3D View Synthesis. In *Advances in Neural Information Processing Systems*, pages 1099–1107, 2015.

[50] Yajie Zhao, Zeng Huang, Tianye Li, Weikai Chen, Chloe LeGendre, Xinglei Ren, Ari Shapiro, and Hao Li. Learning perspective undistortion of portraits. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7849–7859, 2019.

[51] Kun Zhou, Xiaoguang Han, Nianjuan Jiang, Kui Jia, and Jiangbo Lu. Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2344–2353, 2019.