Explicit Residual Descent for 3D Human Pose Estimation from 2D Joint Locations

Yangyuxuan Kang*[12]
kyyx@ios.ac.cn
Anbang Yao*†[3]
anbang.yao@intel.com
Shandong Wang[3]
shandong.wang@intel.com
Ming Lu[3]
ming1.lu@intel.com
Yurong Chen[3]
yurong.chen@intel.com
Enhua Wu†[124]
ehwu@umac.mo

[1] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] Intel Labs China
[4] Faculty of Science and Technology, University of Macau

Abstract

Recent studies show that the end-to-end learning paradigm based on well-designed lifting networks merely using 2D joint locations as the input can achieve impressive performance in handling 3D human pose estimation problem. However, in the viewpoint of optimization design, existing methods of this category have two drawbacks: (1) The inherent feature relation between the 2D pose input and the corresponding 3D pose estimate is not sufficiently explored. (2) The regression procedure is usually performed in a one-step manner. To address these two issues, this paper proposes an efficient yet accurate method called Explicit Residual Descent (ERD). Given an arbitrary lifting network which takes 2D joint locations in a single image as the input and generates an initial 3D pose estimate, our ERD learns a sequence of descent directions encoded with a shared lightweight differentiable structure, progressively refining the previous 3D pose estimate via adding in a 3D increment obtained from projecting the reconstructed 2D pose features onto each learnt descent direction. Extensive experiments on public benchmarks including Human3.6M dataset validate the superior performance of the proposed method against state-of-the-art methods. Code will be made publicly available.

1 Introduction

Estimating accurate 3D human pose from a single monocular image or video is essential for numerous applications such as action recognition, human robot interaction, augmented reality, animation and gaming. As a fundamental ill-posed inverse problem in computer vision, it is challenging especially when images or videos are taken with large variations in body appearance, lighting, occlusion, view-point and background clutter. To ease the problem,

many existing 3D pose estimation systems [5, 6, 20, 25, 41] adopt a two-stage pipeline: detecting 2D joints from input images, and estimating 3D pose given 2D joint locations. Moreover, they universally use powerful deep neural networks as their main building blocks, demonstrating significantly improved performance on standard benchmarks compared with the conventional methods based on hand-crafted features.

With the availability of large-scale pose datasets having rich high-quality 2D skeleton annotations, well-trained deep models for 2D pose estimation are accurate enough for real deployment. Recently, Martinez *et al*. [20] introduced a pioneering optimization scheme in which a fully connected network directly lifts the vectorial input of 2D joint locations to 3D pose space without using any additional cues such as source image/video data, multi-view cameras and pose-conditioned priors, showing surprisingly better accuracy than previous top-performing counterparts. Since then, a lot of methods have been proposed to improve this 2D-to-3D pose estimation scheme mainly in three technical directions: designing more powerful 2D-to-3D lifting networks [6, 19, 25, 40, 41], learning more effective 2D/3D pose representations [2, 7, 8, 29, 36], and extending it to address weakly-supervised/unsupervised learning usage scenarios [6, 13, 27, 35]. Despite substantial improvements in estimation accuracy, these methods usually follow the one-step regression strategy presented in [20], and rarely explore the improvement of inherent feature relation between the 2D pose input and its corresponding 3D pose estimate.

We notice though the prevalent one-step regression strategy builds a forward feature relation from the 2D pose input to its corresponding 3D pose estimate via a lifting network, it is straightforward and suboptimal. This is because smaller network parameter errors are not necessarily equivalent to smaller correspondence errors. Furthermore, there may exist a few 3D body skeletons corresponding to the same 2D pose input due to geometric projection ambiguity. Recent approaches [13, 27, 35] use either random or deterministic projection along with generative adversarial learning to enhance geometric consistency of synthetic/unlabeled pose data instead of real/labeled data, and thus they are tailored to weakly-supervised and unsupervised learning usage scenarios. In this paper, we investigate 2D-to-3D pose estimation from two aspects particulary driven by making the proposed method be applicable to boost a variety of existing lifting networks [6, 19, 20, 25, 40, 41] in the context of supervised learning. Given an arbitrary 2D-to-3D lifting network, we first propose a residual feedback scheme which projects the current 3D pose estimate back to 2D space, and computes the residual difference between the initial 2D pose input and the back-projected 2D pose estimate. We conjecture such a 2D pose residual may serve as a strong feature constraint to reduce 3D pose regression error via mapping it to be a 3D pose increment. Motivated by this, we then propose a new optimization formulation which minimizes an error function measuring the feature relation from 2D pose residuals to the corresponding 3D pose increments. During training, we learn a sequence of descent directions encoded with a shared lightweight differentiable structure over training data iteratively. In testing, given an unseen 2D pose sample, a 3D pose increment is generated by projecting the current sample-specific 2D pose residual onto each learnt descent direction progressively, refining 3D pose estimate from coarse to fine, in a very efficient manner. We call our method Explicit Residual Descent (ERD), which is illustrated in Figure 1 and elaborated in Section 3.

We evaluate the effectiveness and generalization of our method on two public 3D pose estimation datasets Human3.6M [10] and HumanEva [28]. Taking [20] as the base 2D-to-3D lifting network to generate initial 3D pose estimates, on the Human3.6M dataset, our ERD shows 33.9 mm Mean Per-Joint Position Error (MPJPE) using 2D ground truth poses as inputs and 49.3 mm MPJPE for 2D joint detections, which surpass all existing results.

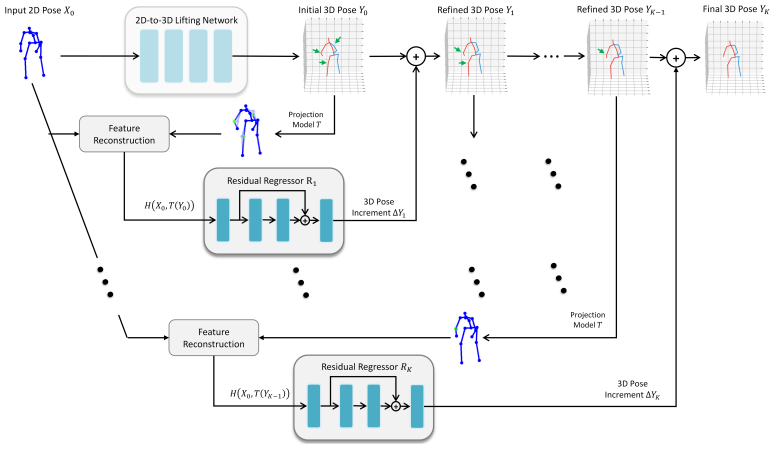

Figure 1: Schematic illustration of ERD. Given a 2D-to-3D lifting network and 2D pose input, ERD projects the previous 3D pose estimate back to 2D image space, reconstructs 2D joint features and refines the 3D pose estimate by the pose increments learnt by the residual regressors progressively in an additive manner.

Meanwhile our ablative studies show that ERD brings consistent accuracy improvements to three state-of-the-art lifting networks [6, 20, 41] when using ground truth 2D joint locations or detections as inputs. Furthermore, our model has only about 0.6 million of parameters, allowing for flexible use in real-time application scenarios.

2 Related Work

This section briefly summarizes recent advances in using deep learning techniques to address 3D pose estimation problem.

End-to-end pose estimation. This category of approaches leverages Convolutional Neural Networks (CNNs) to directly estimate 3D human pose from the input images. Some works [21, 22, 31] integrate the volumetric representation built upon 2D feature maps with different supervision strategies or regression schemes for single-view 3D pose estimation. Iskakov *et al*. [10] extend the volumetric representation to handle multi-view cases. Bogo *et al*. [1] present the first end-to-end CNN framework to estimate a 3D full body mesh from a single unconstrained image via fitting a statistical body shape model called SMPL [18] to 2D joint heatmaps. Several improved variants of this framework have been proposed in [14, 24]. As feature representation plays a critical role in an end-to-end framework, a couple of works [9, 23, 33, 37, 38] focus on engineering more powerful CNN models dedicated to 3D pose estimation. Despite their great success, training high-performance end-to-end models is hungry for 3D annotated pose data due to the large amount of learnable parameters. However, capturing pose data with 3D annotations is challenging and expensive. Consequently, there exist many learning based methods to address the problem of limited 3D annotated data, such as using generative models [12, 16] and adversarial models [26, 39] to augment the training given available labeled and unlabeled datasets.

Two-stage pose estimation. This category of approaches builds 3D pose estimators on top of 2D joint detectors whose outputs in image space are then lifted to 3D pose space.

Martinez *et al*. [20] introduce the first lifting network to directly regress 3D pose taking 2D joint locations as the input. Since 2D joint detection is relatively mature, many approaches attempt to improve the 2D-to-3D pose lifting framework. Instead of using Fully Connected Network (FCN) presented in [20], some recent works focus on designing more effective and efficient lifting networks, such as the semantic Graph Convolution Network (GCN) [41], the Locally Connected Network (LCN) [6], the temporal convolutional network [25] and the chirality nets [40]. There also exist a lot of works [2, 7, 8, 36] trying to learn more effective pose representations via utilizing kinematics and deformation knowledge regarding human body configuration. Recently, many weakly-supervised and unsupervised two-stage methods [9, 13, 27, 35] have been proposed to address the lack of sufficient 3D annotated data, especially in multi-view scenarios.

Our method falls under the two-stage category, and we focus on the latter stage in the context of supervised learning. To the best of our knowledge, this paper is the first work exploring the way to build a novel residual feedback optimization scheme for progressively reducing the estimation error of any existing 2D-to-3D pose lifting network, merely using reconstructed 2D joint features.

3 Method

In this section, we describe our ERD method to the supervised 2D-to-3D human pose estimation problem.

3.1 Overall Design

The goal of ERD is to estimate a more accurate 3D pose given its initial estimate by a 2D-to-3D lifting network taking 2D joint locations as the input. We do not add any architectural constraint to the lifting network, intending to make ERD be readily applicable to different types of existing lifting networks [6, 19, 25, 40, 41].

Given a dataset including N human pose samples, let $X_0 = \{x_0^i\}_{i=1}^N$ be the gallery of 2D joints, and let $Y = \{y^i\}_{i=1}^N$ be the gallery of ground truth joints in a predefined 3D space, where $x_0^i \in R^{2J}$, $y^i \in R^{3J}$, and J is the number of joints for a body skeleton. In our experiments, x_0^i can be either ground truth 2D joint locations or outputs of a 2D joint detector. By using a lifting network F to X_0, we can obtain a gallery of initial 3D pose estimates denoted as $Y_0 = \{y_0^i\}_{i=1}^N$. We notice that the one-step regression strategy popularly adopted by existing 2D-to-3D lifting networks lacks a feedback mechanism in the optimization to compensate for potentially weak estimation results. To fill this gap, our ERD presents an effective yet efficient design from a perspective of learning a set of K residual regressors to progressively update 3D pose estimate. Briefly, ERD projects the previous 3D pose estimate Y_{k-1} back to 2D space, regresses a 3D pose increment ΔY_k from the reconstructed features in 2D space, and computes the current 3D pose estimate Y_k in an additive manner:

$$Y_k = Y_{k-1} + \Delta Y_k, \quad k = 1, \cdots, K. \tag{1}$$

The 3D pose increment ΔY_k is computed as

$$\Delta Y_k = R_k(H(X_0, T(Y_{k-1}))), \tag{2}$$

where R_k is the residual regressor updates the previous 3D pose estimate Y_{k-1} to the new estimate Y_k, T is a known projection model maps Y_{k-1} back to 2D space, and H represents the reconstructed features conditioned on the initial 2D joints X_0 and the back-projected 2D joints corresponding to Y_{k-1}. Note that the residual regressor R_k depends on both the projection model T and the reconstructed features H, which will be elucidated later. During training, ERD learns each residual regressor R_k by minimizing the following error function

$$\arg\min_{R_k} \|Y - (Y_{k-1} + R_k(H(X_0, T(Y_{k-1})))\| . \tag{3}$$

3.2 Residual Regressors

In the presence of large 2D pose variations and complex 2D-to-3D pose correspondences, one single residual regressor might be not optimal for reducing the estimation error. This is why ERD introduces a set of K residual regressors to progressively update 3D pose estimate. Intuitively, the early residual regressors compensate for large 3D pose error fluctuations, while the latter residual regressors perform minor adjustments, guaranteeing generalization and accuracy on large-scale datasets. In our experiments, ERD converges with no more than 4 residual regressors. To learn good residual regressors, how to reconstruct informative features in 2D pose space and how to construct the structure of residual regressors are critical.

Reconstructed 2D pose features. We use the residual difference between the initial 2D joints X_0 and the back-projected 2D joints from the previous 3D pose estimate Y_{k-1} as the input features to learn each residual regressor R_k. Formally, we have

$$H(X_0, T(Y_{k-1})) = X_0 - T(Y_{k-1}). \tag{4}$$

We find such kind of features is compact and discriminative as it explicitly encodes the discrepancy between the initial input and the back-projected estimate in 2D pose space, and transferring them into a 3D pose increment builds up a bidirectional feature relation in both 2D and 3D pose spaces. It shows much better results compared with other kinds of features, as validated by our ablative studies. Alternatively, we empirically find that simply applying conventional compression techniques like Principle Component Analysis (PCA) to transform both the 2D pose input and the reconstructed 2D pose residual features into a slightly more compact representation will bring additional accuracy gain (ranging from 1.1 mm to 2.4 mm on the Human3.6M dataset) to both the lifting network and our method. Similar experimental observations on 3D pose features are also reported in [36].

Structure of residual regressors. In ERD, each residual regressor actually learns a descent direction. By projecting reconstructed 2D pose features on the learnt descent direction of a residual regresor, the sample-specific 3D pose increment can be generated to refine the previous 3D pose estimate. We use a simple neural network as the shared structure for all residual regressors. The network only includes two Fully Connected (FC) layers (one increasing the dimensionality of the input to 256, and the other predicating a 3D pose vector) and a tiny residual block having two hidden layers (each one has 256 hidden nodes and is followed by dropout with a ratio of 0.25). The first FC layer is followed by the operations of batch normalization, Rectified Linear Unit (ReLU), and dropout with a ratio of 0.25. This network is very lightweight as it only has about 0.15 million of parameters. In the experiments, we did not pay extra attention to tune this structure owing to its satisfied accuracy and high efficiency.

3.3 Projection Model

Recall that to reconstruct sample-specific 2D residual features, ERD projects the previous 3D pose estimate back to 2D space by a projection model. For easy implementation, we use popular perspective projection as our projection model. Given a 3D pose estimate having J joints, let (p_x, p_y, p_z) be a joint location with respect to the root joint in camera coordinates, and let (q_x, q_y) be the corresponding joint location in image coordinates. With the perspective projection, we have

$$\begin{cases} q_x &= \dfrac{f_x\,(p_x + root_x)}{p_z + root_z} + c_x \\ q_y &= \dfrac{f_y\,(p_y + root_y)}{p_z + root_z} + c_y, \end{cases} \tag{5}$$

where focal length (f_x, f_y) and optical center (i.e., principal point) (c_x, c_y) are camera intrinsic parameters, and $(root_x, root_y, root_z)$ is the global position of the body root joint (the pelvis joint is commonly used as the root joint in the literature) relative to the camera. For prevailing commercial cameras even including low-end devices, the camera intrinsic parameters can be easily obtained from the EXIF metadata of images or videos. As for the global position of the body root joint, it can be well regressed from the training data by learning based algorithms as shown in recent works [2, 25, 34]. Since estimating the global position is not the focus of this paper, we simply apply Singular Value Decomposition (SVD) to directly solve the inverse problem of Eq. 5 taking the global position $(root_x, root_y, root_z)$ as the only unknown parameters regarding all joints of each pose. The resulting global position maybe be further refined by learning based algorithms.

4 Experiments

4.1 Datasets

We evaluate our method on two public datasets, Human3.6M [10] and HumanEva-I [28].

Human3.6M. Human3.6M is currently the largest dataset for 3D human pose estimation. It consists of about 3.6 million video frames collected from 11 subjects. Each subject performs 15 actions recorded with a motion capture system having four RGB cameras and 1 depth camera synchronized at 50Hz. Following previous works [2, 6, 8, 19, 20, 25, 36, 40, 41], we adopt a 17-joint body skeleton, use subjects S1, S5, S6, S7 and S8 for training, and subjects S9 and S11 for testing.

HumanEva-I. Compared with Human3.6M, HumanEva-I is a much smaller dataset. It is recorded with 3 subjects from 3 different camera views at 60Hz. Following [20, 25], we adopt a 15-joint body skeleton, and evaluate our method on 3 actions (Walk, Jog, Box) using the provided train/test partition. It should be noted that manually synchronization and sampling lead to inaccurate annotations for some samples, which makes HumanEva-I have relative worse annotations in comparison to Human3.6M. We use it to test the generalization ability of our method.

4.2 Implement Details

Following [6, 19, 20, 25, 40, 41], we also consider two application scenarios, using either ground truth 2D joint locations (denoted as GT2D in our results) or outputs of a 2D joint

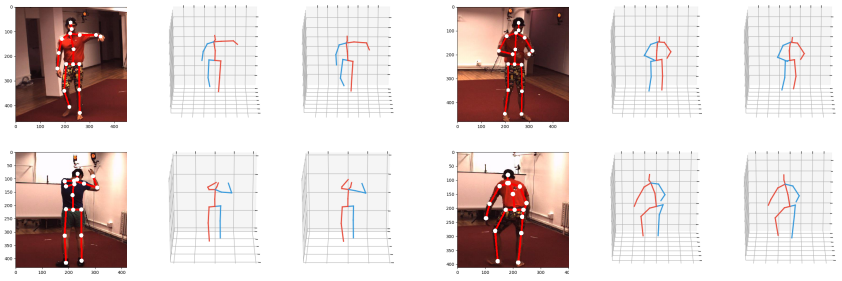

Figure 2: Illustrative result visualizations on the test set of Human3.6M. The left of each triplet shows the image overlayed with the detected 2D pose, the middle shows our 3D pose predication, and the right shows the ground truth 3D pose.

detector (denoted as DET2D in our results) as inputs in both training and testing. As for 2D joint detector, we use Cascaded Pyramid Network (CPN) [4] pre-trained on the MS-COCO dataset [17]. The pre-trained 2D joint detector is fine-tuned on each dataset. We normalize 2D joint inputs and 3D predications by subtracting the mean and dividing by the standard deviation. We follow the standard evaluation protocol, and report the Mean Per-Joint Position Error (MPJPE) in millimeters (mm) on both datasets. MPJPE measures the mean Euclidean distance between the estimated 3D joint locations and the ground truth. To better show the advantage of our method, in each experiment, we train one model for all actions instead of training a different model for each action.

2D-to-3D pose lifting networks. As our method is designed to boost the performance of existing 2D-to-3D pose lifting networks, we consider three state-of-the-art lifting networks [6, 20, 41] in our experiments. Specifically, we apply our method to the well-known lifting network [20] in our basic experiments, and consider two recently proposed lifting networks [6, 41] in an ablative study. We use the code released by the authors to train each lifting network from scratch on a single NVIDIA Titan X GPU. We train each lifting network for 200 epochs with a batch size of 200 and Adam optimizer. The learning rate starts with 0.001 and decays exponentially with a shrink factor of 0.96. Taking the outputs of each lifting network as the initial 3D pose estimates, we train 4 residual regressors progressively with the same settings except the number of training epochs is set to 5 as our method shows very fast convergence. As a result, our model has only about 0.6 million of parameters, running with an average speed of about 4000 fps on a single NVIDIA Titan X GPU (without considering the time cost of the lifting network).

4.3 Results on Human3.6M

Table 1 summarizes the results of our method and other recently proposed methods on the Human3.6M dataset, in which we report both averaged error over all actions and action-specific errors. For the GT2D input, it can be seen: (1) when the root joint (i.e., pelvis joint) location is not available, our model shows 35.8 mm error using estimated root joint locations, which is better than the previous best result; (2) using ground truth root joint locations, our model achieves 33.9 mm error, outperforming [6] by a margin of 2.4 mm. For the DET2D input, we can observe: (1) using estimated root joint locations, the error of our model is 53.0 mm, which is comparable to the state-of-the-art result obtained by a temporal convolution

Method	Dir.	Dis.	Eat	Gre.	Phon.	Phot.	Pose	Pur.	Sit	SitD.	Smo.	Wait	WalkD.	Walk	WalkT	Avg
Pavlakos *et al.* [25]	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Tekin *et al.* [32]	54.2	61.4	60.2	61.2	79.4	78.3	63.1	81.6	70.1	107.3	69.3	70.3	74.3	51.8	63.2	69.7
Zhou *et al.* [38]	54.8	60.7	58.2	71.4	62.0	65.5	53.8	55.6	75.2	111.6	64.2	66.1	51.4	63.2	55.3	64.9
Sun *et al.* [29]	52.8	54.8	54.2	54.3	61.8	67.2	53.1	53.6	71.7	86.7	61.5	53.4	61.6	47.1	53.4	59.1
Sun *et al.* [30]	63.8	64.0	56.9	64.8	62.1	59.8	60.1	71.6	91.7	60.9	70.4	65.1	63.2	51.3	55.4	64.1
Martinez *et al.* [20]	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Fang *et al.* [8]	50.1	54.3	57.0	57.1	66.6	73.3	53.4	55.7	72.8	88.6	60.3	57.7	62.7	47.5	50.6	60.4
Yang *et al.* [36]	51.5	58.9	50.4	57.0	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6
Zhao *et al.* [40]	47.3	60.7	51.4	60.5	61.1	49.9	47.3	68.1	86.2	55.0	67.8	61.0	42.1	60.6	45.3	57.6
Pavllo *et al.* [26]	47.1	50.6	49.0	51.8	53.6	61.4	49.4	47.4	59.3	67.4	52.4	49.5	55.3	39.5	42.7	**51.8**
Ci *et al.* [9]	46.2	49.9	50.0	50.5	56.0	67.3	49.1	47.4	63.4	71.6	52.7	50.3	55.9	40.8	45.9	53.1
Ours	46.7	50.8	49.0	50.7	56.8	66.8	48.9	47.8	62.7	73.0	52.6	51.0	55.0	39.7	43.9	53.0
Ci *et al.* [9](+)	46.8	52.3	44.7	50.4	52.9	68.9	49.6	46.4	60.2	78.9	51.2	50.0	54.8	40.4	43.3	52.7
Ours(+)	43.3	47.1	46.2	47.9	50.9	62.7	46.5	44.0	58.0	67.9	49.2	47.6	50.4	37.1	41.4	**49.4**
Martinez *et al.* [20]	37.7	44.4	40.3	42.1	48.2	54.9	44.4	42.1	54.6	58.0	45.1	46.4	47.6	36.4	40.4	45.5
Zhao *et al.* [40]	37.8	49.4	37.6	40.9	45.1	41.4	40.1	48.3	50.1	42.2	53.5	44.3	40.5	47.3	39.0	43.8
Pavllo *et al.* [26]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37.2
Ci *et al.* [9]	36.2	40.8	33.1	38.1	39.9	47.2	42.2	35.3	43.9	46.8	38.5	40.5	38.9	31.1	33.8	39.1
Ours	34.4	36.4	31.2	35.0	37.6	45.4	36.4	31.2	40.6	42.7	36.2	34.8	37.2	27.8	30.7	**35.8**
Ci *et al.* [9](+)	36.3	38.8	29.7	37.8	34.6	42.5	39.8	32.5	36.2	39.5	34.4	38.4	38.2	31.3	34.2	36.3
Ours(+)	34.1	33.8	28.8	33.4	35.6	42.3	35.1	28.7	37.8	39.0	34.2	33.2	35.3	26.8	30.0	**33.9**

Table 1: Results comparison on the Human3.6M dataset using the standard evaluation metric MPJPE (mm). The upper part of the table shows the results using the GT2D as the input during both training and testing, and the lower part shows the results using the DET2D as the input. As the default setting of our method, we use SVD to directly estimate global root joint locations for training and testing samples, while (+) indicates the ground truth root joint locations are used. Results of all the other methods are obtained from the original papers. For our method, we train a lifting network of [20] for generating initial 3D pose estimates, which also applies to the other experiments of this paper unless otherwise stated.

network (single frame testing) [25]; (2) when the root joint location is supposed to be known, our model achieves an impressive error of 49.3 mm.

Note that we train a lifting model of [20] to generate initial 3D pose estimates. By applying our method to refine its predications, we improve the results reported in [20] by over 9.5 mm margin in the context of using either ground truth 2D joint locations or outputs of a joint detector. Although we directly use SVD to estimate the root joint location for each pose sample and achieve very competitive results, we notice more accurate estimation results can be obtained by learning based algorithms [2, 25, 34], which we leave for our future work. Some illustrative 3D pose results of our method are shown in Figure 2.

4.4 Ablation Studies

We conduct extensive ablative experiments to evaluate the effectiveness of different components of our method, and its generalization ability to different lifting networks and datasets.

Residual regressor number. Firstly, we analyze the tradeoff between the number of residual regressors and training convergence. From the results shown in Figure 3, we can see the error curve converges at no more than 4 stages. Large error drop appears at the first three residual regressors under all four different training settings, and the latter stages show minor error adjustments.

Root joint for projection model. Secondly, considering 2D pose feature reconstruction depends on the projection model, we conduct experiments to compare the performance of our method using estimated root joint locations vs. the ground truth. Results are shown in Figure 3. It is not surprising that the more accurate the root location, the higher the estimation accuracy. The blue curves in both left and right sub-figures show the lowest boundary when the root location is sufficiently accurate. With DET2D and coarse root location predications, the improvement compromises on a small margin.

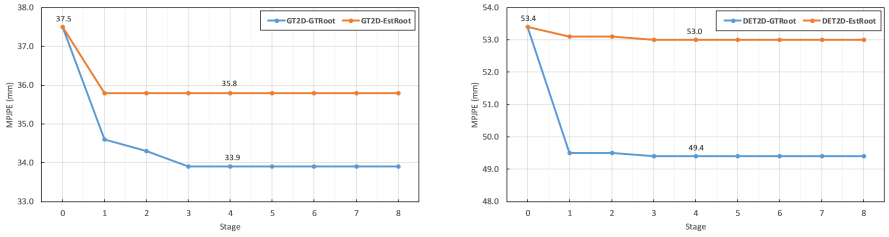

Figure 3: Error distribution of our method with different numbers of residual regressors on the Human3.6M dataset. Stage 0 denotes the baseline lifting model [20] reproduced by us.

Recon. Feature	Init. 2D Pose	Init. MPJPE(mm)	Final MPJPE(mm)	Gain(mm)
Init2D	GT2D	37.5	37.3	0.2
Proj2D	GT2D	37.5	37.4	0.1
Concat	GT2D	37.5	35.2	2.3
Residual	GT2D	37.5	33.9	3.6
Init2D	DET2D	53.4	53.4	0
Proj2D	DET2D	53.4	53.4	0
Concat	DET2D	53.4	50.5	2.9
Residual	DET2D	53.4	49.3	4.1

Table 2: Evaluation of our method on the Human3.6 dataset using different reconstructed 2D pose features. The root joint location is assumed as known. *Init2D* represents original 2D pose. *Proj2D* represents back-projected 2D pose estimate. *Concat* represents the concatenation of *Init.2D* and *Proj2D*. *Residual* represents our proposed 2D pose residual feature. We use our reproduced lifting model [20] as the baseline.

2D pose feature reconstruction. Thirdly, we study different choices of reconstructing 2D pose features fed into each residual regressor for predicating 3D pose increment. Specifically, we evaluate four choices: (1) original 2D pose denoted as *Init2D*; (2) back-projected 2D pose estimate denoted as *Proj2D*; (3) the concatenation of *Init2D* and *Proj2D* denoted as *Concat*; (4) the proposed one denoted as *Residual*, i.e., the residual difference between *Init2D* and *Proj2D*. Evaluation results are provided in Table 2. Comparatively, we can see *Init2D* and *Proj2D* have negligible improvements to baseline lifting models, and *Concat* performs much better by incorporating both *Init2D* and *Proj2D*, while our *Residual* is the best showing over 3.5 mm margins in both training settings.

2D-to-3D pose lifting networks. Fourthly, since our method intends to improve the performance of existing 2D-to-3D pose lifting networks, we conduct experiments to validate its effectiveness on different lifting baselines. Evaluation results in Table 3 show that our method also brings large improvements to two other state-of-the-art lifting networks [6, 41] as well as the well-known one [20]. Besides, we also explore the possibility of using the back-projected 2D pose from 3D estimate of a 2D-to-3D pose lifting network as the new input to directly refine its estimate. Taking ground truth 2D pose as the initial input for the example case, the back-projected 2D pose is less accurate than the initial input, thus feeding it to a 2D-to-3D pose lifting network will worsen 3D pose estimate. Specifically, in the experiments on Human3.6M, accuracy drop to [20], [6] and [41] is 4.8 mm, 3.4 mm and 4.2 mm, respectively. Similar drop is also observed on detected 2D pose input.

Lifting Network	Root Joint Locations	Error on GT2D	Error on DET2D
Martinez *et al.* [20]	GTRoot	33.9 (37.5)	49.4 (53.4)
Ci *et al.* [6]	GTRoot	35.3 (39.1)	50.0 (53.5)
Zhao *et al.* [41]	GTRoot	36.3 (39.9)	53.3 (57.2)
Martinez *et al.* [20]	EstRoot	35.8 (37.5)	53.0 (53.4)
Ci *et al.* [6]	EstRoot	37.2 (39.1)	53.2 (53.5)
Zhao *et al.* [41]	EstRoot	38.1 (39.9)	56.8 (57.2)

Table 3: Evaluation of our method on the Human3.6 dataset using different 2D-to-3D pose lifting networks. Results (mm) inside the bracket are for baselines, while outside ones are ours.

	Walk			Jog			Box		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
Palvakos[21]	22.3	19.5	29.7	28.9	21.9	23.8	-	-	-
Martinez[20]	19.7	17.4	46.8	26.9	18.2	18.6	-	-	-
Pavlakos[23]	18.8	**12.7**	29.2	23.5	15.4	**14.5**	-	-	-
Lee[15]	18.6	19.9	30.5	25.7	16.8	17.7	42.8	48.1	53.4
Martinez[20](*)	19.6	18.1	26.3	27.5	16.6	17.8	29.9	37.2	35.5
Ours	**15.3**	**12.7**	**22.3**	**22.8**	**13.5**	**14.5**	**27.7**	**35.2**	**30.7**

Table 4: Results comparison on the HumanEva-I dataset. MPJPE (mm) is computed after procrustes transformation. * represents the baseline lifting model [20] reproduced by us.

Generalization to HumanEva-I. Finally, we apply our method to HumanEva-I dataset to testify its generalization ability using the same lifting network [20] as on the Human3.6M dataset. We train our model using DET2D and estimated root point locations. The results are shown in Table 4 from which we can see: regarding different subjects over all three actions, our model brings at least 2.0 mm and at most 5.4 mm error reduction to the baseline lifting model. Our method mostly achieves lower errors compared with other methods.

5 Conclusions

In this paper, we present ERD, a fast and effective 3D human pose estimation method, which learns a set of residual regressors to progressively refine the predication of any existing 2D-to-3D pose lifting network merely using 2D joint features from a single image. Experiments on two public datasets validate the effectiveness of our method.

Acknowledgement

⋆ indicates equal contribution. † indicates corresponding authors. This work was done when Yangyuxuan Kang was an intern at Intel Labs China, supervised by Anbang Yao. This work was also supported by the National High Technology R&D Program of China (2017YFB1002701), NSFC(61632003, 61672502), and University of Macau Grant (MYRG2019-00006-FST).

References

[1] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 561–578, 2016.

[2] Ju Yong Chang, Gyeongsik Moon, and Kyoung Mu Lee. Absposelifter: Absolute 3d human pose lifting network from a single noisy 2d human pose. *arXiv preprint arXiv:1910.12029*, 2019.

[3] Xipeng Chen, Kwan-Yee Lin, Wentao Liu, Chen Qian, and Liang Lin. Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10895–10904, 2019.

[4] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018.

[5] Yu Cheng, Bo Yang, Bo Wang, Wending Yan, and Robby T Tan. Occlusion-aware networks for 3d human pose estimation in video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 723–732, 2019.

[6] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2262–2271, 2019.

[7] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. Learning 3d human pose from structure and motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 668–683, 2018.

[8] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[9] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt. In the wild human pose estimation using explicit 2d features and intermediate 3d representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10905–10914, 2019.

[10] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7): 1325–1339, 2013.

[11] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7718–7727, 2019.

[12] Ehsan Jahangiri and Alan L Yuille. Generating multiple diverse hypotheses for human 3d pose consistent with 2d joint detections. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 805–814, 2017.

[13] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1077–1086, 2019.

[14] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2252–2261, 2019.

[15] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–135, 2018.

[16] Chen Li and Gim Hee Lee. Generating multiple hypotheses for 3d human pose estimation with mixture density network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9887–9895, 2019.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[18] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[19] Chenxu Luo, Xiao Chu, and Alan Yuille. Orinet: A fully convolutional network for 3d human pose estimation. *arXiv preprint arXiv:1811.04989*, 2018.

[20] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017.

[21] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017.

[22] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3d human pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6988–6997, 2017.

[23] 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.

[24] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–468, 2018.

[25] 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.

[26] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[27] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 750–767, 2018.

[28] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 87(1-2):4, 2010.

[29] Jennifer J Sun, Jiaping Zhao, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, and Ting Liu. View-invariant probabilistic embedding for human pose. *arXiv preprint arXiv:1912.01001*, 2019.

[30] 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.

[31] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018.

[32] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3941–3950, 2017.

[33] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2500–2509, 2017.

[34] Márton Véges and András Lőrincz. Absolute human pose estimation with depth prediction network. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019.

[35] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7782–7791, 2019.

[36] Chunyu Wang, Haibo Qiu, Alan L Yuille, and Wenjun Zeng. Learning basis representation to refine 3d human pose estimations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8925–8932, 2019.

[37] Jue Wang, Shaoli Huang, Xinchao Wang, and Dacheng Tao. Not all parts are created equal: 3d pose estimation by modeling bi-directional dependencies of body parts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7771–7780, 2019.

[38] Min Wang, Xipeng Chen, Wentao Liu, Chen Qian, Liang Lin, and Lizhuang Ma. Dr-pose3d: Depth ranking in 3d human pose estimation. *arXiv preprint arXiv:1805.08973*, 2018.

[39] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5255–5264, 2018.

[40] Raymond Yeh, Yuan-Ting Hu, and Alexander Schwing. Chirality nets for human pose regression. In *Advances in Neural Information Processing Systems*, pages 8161–8171, 2019.

[41] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.

[42] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 398–407, 2017.