Integral Human Pose Regression

Xiao Sun[1], Bin Xiao[1], Fangyin Wei[2], Shuang Liang[3(✉)], and Yichen Wei[1]

[1] Microsoft Research, Beijing, China
{xias,Bin.Xiao,yichenw}@microsoft.com
[2] Peking University, Beijing, China
weifangyin@pku.edu.cn
[3] Tongji University, Shanghai, China
shuangliang@tongji.edu.cn

Abstract. State-of-the-art human pose estimation methods are based on heat map representation. In spite of the good performance, the representation has a few issues in nature, such as non-differentiable post-processing and quantization error. This work shows that a simple *integral* operation relates and unifies the heat map representation and joint regression, thus avoiding the above issues. It is differentiable, efficient, and compatible with *any* heat map based methods. Its effectiveness is convincingly validated via comprehensive ablation experiments under various settings, specifically on 3D pose estimation, for the first time.

Keywords: Integral regression · Human pose estimation
Deep learning

1 Introduction

Human pose estimation has been extensively studied [3,24,28]. Recent years have seen significant progress on the problem, using deep convolutional neural networks (CNNs). Best performing methods on 2D pose estimation are all detection based [2]. They generate a likelihood heat map for each joint and locate the joint as the point with the maximum likelihood in the map. The heat maps are also extended for 3D pose estimation and shown promising [37].

Despite its good performance, a heat map representation bears a few drawbacks in nature. The "taking-maximum" operation is not differentiable and prevents training from being end-to-end. A heat map has lower resolution than that of input image due to the down sampling steps in a deep neural network. This causes inevitable quantization errors. Using image and heat map with higher resolution helps to increase accuracy but is computational and storage demanding, especially for 3D heat maps.

From another viewpoint, pose estimation is essentially a regression problem. A regression approach performs end-to-end learning and produces continuous output. It avoids the issues above. However, regression methods are not as effective as well as detection based methods for 2D human pose estimation.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11210, pp. 536–553, 2018.
https://doi.org/10.1007/978-3-030-01231-1_33

Among the best-performing methods in the 2D pose benchmark [2], only one method [7] is regression based. A possible reason is that regression learning is more difficult than heat map learning, because the latter is supervised by dense pixel information. While regression methods are widely used for 3D pose estimation [14,21,30–32,35,42,43,55,56], its performance is still not satisfactory.

Existing works are either detection based or regression based. There is clear discrepancy between the two categories and there is little work studying their relation. This work shows that a simple operation would relate and unify the heat map representation and joint regression. It modifies the "taking-maximum" operation to"taking-expectation". The joint is estimated as the integration of all locations in the heat map, weighted by their probabilities (normalized from likelihoods). We call this approach *integral regression*. It shares the merits of both heat map representation and regression approaches, while avoiding their drawbacks. The integral function is differentiable and allows end-to-end training. It is simple and brings little overhead in computation and storage. Moreover, it can be easily combined with *any* heat map based methods.

The integral operation itself is not new. It has been known as *soft-argmax* and used in the previous works [27,45,52]. Specifically, two contemporary works [29, 34] also apply it for human pose estimation. Nevertheless, these works have limited ablation experiments. The effectiveness of integral regression is not fully evaluated. Specifically, they only perform experiments on MPII 2D benchmark, on which the performance is nearly saturated. It is yet unclear whether the approach is effective under other settings, such as 3D pose estimation. See Sect. 3 for more discussions.

Because the integral regression is parameter free and only transforms the pose representation from a heat map to a joint, it does not affect other algorithm design choices and can be combined with any of them, including different *tasks*, *heat map and joint losses*, *network architectures*, *image and heat map resolutions*. See Fig. 1 for a summarization. We conduct comprehensive experiments to investigate the performance of integral regression under all such settings and find consistent improvement. Such results verify the effectiveness of integral representation.

Our main contribution is applying integral regression under various experiment settings and verifying its effectiveness. Specifically, we firstly show that integral regression significantly improves the 3D pose estimation, enables the mixed usage of 3D and 2D data, and achieves state-of-the-art results on Human3.6M [24]. Our results on 2D pose benchmarks (MPII [3] and COCO [28]) is also competitive. Code[1] will be released to facilitate future work.

2 Integral Pose Regression

Given a learnt heat map $\mathbf{H}_k$ for k^{th} joint, each location in the map represents the probability of the location being the joint. The final joint location coordinate

[1] https://github.com/JimmySuen/integral-human-pose.

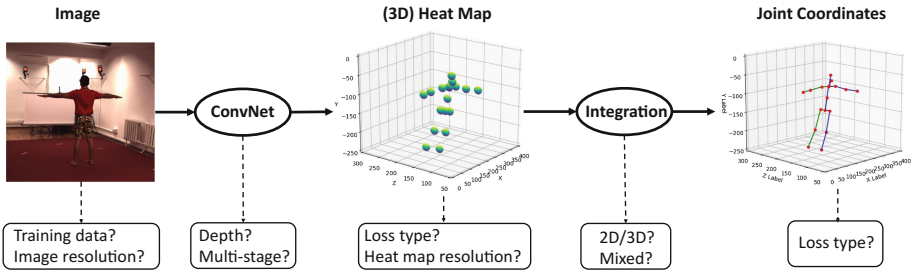

Fig. 1. Overview of pose estimation pipeline and all our ablation experiment settings.

$\mathbf{J}_k$ is obtained as the location $\mathbf{p}$ with the *maximum likelihood* as

$$\mathbf{J}_k = \arg\max_{\mathbf{p}} \mathbf{H}_k(\mathbf{p}). \tag{1}$$

This approach has two main drawbacks. First, Eq. (1) is *non-differentiable*, reducing itself to a post-processing step but not a component of learning. The training is not end-to-end. The supervision could only be imposed on the heat maps for learning.

Second, the heat map representation leads to *quantization error*. The heat map resolution is much lower than the input image resolution due to the down sampling steps in a deep neural network. The joint localization precision is thus limited by the quantization factor, which poses challenges for accurate joint localization. Using larger heat maps could alleviate this problem, but at the cost of extra storage and computation.

Regression methods have two clear advantages over heat map based methods. First, learning is *end-to-end* and driven by the goal of joint prediction, bridging the common gap between learning and inference. Second, the output is *continuous* and up to arbitrary localization accuracy, in principle. This is opposed to the quantization problem in heat maps.

We present a unified approach that transforms the heat map into joint location coordinate and fundamentally narrows down the gap between heat map and regression based method. It brings principled and practical benefits.

Our approach simply modifies the *max* operation in Eq. (1) to take expectation, as

$$\mathbf{J}_k = \int_{\mathbf{p}\in\Omega} \mathbf{p} \cdot \tilde{\mathbf{H}}_k(\mathbf{p}). \tag{2}$$

Here, $\tilde{\mathbf{H}}_k$ is the normalized heat map and Ω is its domain. The estimated joint is the integration of all locations $\mathbf{p}$ in the domain, weighted by their probabilities.

Normalization is to make all elements of $\tilde{\mathbf{H}}_k(\mathbf{p})$ non-negative and sum to one. [34] has already discussed it and we use softmax in this paper as

$$\tilde{\mathbf{H}}_k(\mathbf{p}) = \frac{e^{\mathbf{H}_k(\mathbf{p})}}{\int_{\mathbf{q}\in\Omega} e^{\mathbf{H}_k(\mathbf{q})}}. \tag{3}$$

The discrete form of Eq. (2) is

$$\mathbf{J}_k = \sum_{p_z=1}^{D} \sum_{p_y=1}^{H} \sum_{p_x=1}^{W} \mathbf{p} \cdot \tilde{\mathbf{H}}_k(\mathbf{p}), \tag{4}$$

By default, the heat map is 3D. Its resolution on depth, height, and width are denoted as D, H, and W respectively. $D = 1$ for 2D heat maps.

In this way, any heat map based approach can be augmented for joint estimation by appending the integral function in Eq. (4) to the heat map $\mathbf{H}_k$ and adopting a regression loss for $\mathbf{J}_k$. We call this approach *integral pose regression*.

Integral pose regression shares all the merits of both heat map based and regression approaches. The integral function in Eq. (4) is differentiable and allows end-to-end training. It is simple, fast and non-parametric. It can be easily combined with any heat map based methods, while adding negligible overhead in computation and memory for either training or inference. Its underlying heat map representation makes it easy to train. It has continuous output and does not suffer from the quantization problem.

2.1 Joint 3D and 2D Training

A lack of diverse training data is a severe problem for 3D human pose estimation. Several efforts have been made to combine 3D and 2D training [31,41,43,51,55]. Since integral regression provides a unified setting for both 2D and 3D pose estimation, it is a simple and general solution to facilitate joint 3D and 2D training so as to address this data issue in 3D human pose estimation.

Recently, Sun et al. [42] introduce a simple yet effective way to mix 2D and 3D data for 3D human pose estimation and show tremendous improvement. The key is to separate the 2D part (xy) of the joint prediction $\mathbf{J}_k$ from the depth part (z) so that the xy part can be supervised by the abundant 2D data.

Integral regression can naturally adopt this mixed training technique, thanks to the *differentiability* of integral operation in Eq. (4). We also obtain enormous improvement from this technique in our experiments and this improvement is feasible due to the integral formulation.

However, the underlying 3D heat map still can not be supervised by the abundant 2D data. To address this problem, we further decompose the integral function Eq. (4) into a two-step version to generate separate x, y, z heat map target. For example, for the x target, we first integrate the 3D heat map into 1D x heat vectors Eq. (5)

$$\tilde{\mathbf{V}}_k^x = \sum_{p_z=1}^{D} \sum_{p_y=1}^{H} \tilde{\mathbf{H}}_k(\mathbf{p}), \tag{5}$$

and then, further integrate the 1D x heat vector into x joint coordinate Eq. (6)

$$\mathbf{J}_k^x = \sum_{p_x=1}^{W} \mathbf{p} \cdot \tilde{\mathbf{V}}_k(\mathbf{p}). \tag{6}$$

Corresponding y and z formulation should be easy to infer. In this way, the x, y, z targets are separated at the first step, allowing the 2D and 3D mixed data training strategy. We obtain significant improvements from both direct and two-step integral regression for 3D pose estimation.

3 Methodology for Comprehensive Experiment

The main contribution of this work is a comprehensive methodology for ablation experiments to evaluate the performance of the integral regression under various conditions. Figure 1 illustrates the overview of the framework and the decision choices at each stage.

The related works [29,34] only experimented with 2D pose estimation on MPII benchmark [2]. They also have limited ablation experiments. Specifically, [29] provides only system-level comparison results without any ablation experiments. [34] studies the heat map normalization methods, heat map regularization and backbone networks, which is far less comprehensive than ours.

Tasks. Our approach is general and is ready for both 2D and 3D pose estimation tasks, indistinguishably. Consistent improvements are obtained from both tasks. Particularly, 2D and 3D data can be easily mixed simultaneously in the training. The 3D task benefits more from this technique and outperforms previous works by large margins.

Network Architecture. We use a simple network architecture that is widely adopted in other vision tasks such as object detection and segmentation [19,20]. It consists of a deep convolutional *backbone* network to extract convolutional features from the input image, and a shallow *head* network to estimate the target output (heat maps or joints) from the features.

In the experiment, we show that our approach is a flexible component which can be easily embedded into various backbone networks and the result is less affected by the network capacity than the heat map. Specifically, *network designs* ResNet [20] and HourGlass [33], *network depth* ResNet18, 50, 101 [20], *multi-stage* design [7,49] are investigated.

Heat Map Losses. In the literature, there are several choices of loss function for heat maps. The most widely adopted is mean squared error (or *L2* distance) between the predicted heat map and ground-truth heat map with a 2D Gaussian blob centered on the ground truth joint location [5,6,10,12,13,33,48,49]. In this work, the Gaussian blob has standard deviation $\sigma = 1$ as in [33]. Our baseline with this loss is denoted as H1 (H for heat map).

The recent Mask RCNN work [19] uses a one-hot $m \times m$ ground truth mask where only a single location is labeled as joint. It uses the cross-entropy loss over an m^2-way softmax output. Our baseline with this loss is denoted as H2.

Another line of works [22,36,38] solve a per-pixel binary classification problem, thus using binary cross-entropy loss. Each location in each heat map is classified as a joint or not. Following [22,38], the ground truth heat map for each joint is constructed by assigning a positive label 1 at each location within

15 pixels to the ground truth joint, and negative label 0 otherwise. Our baseline with this implementation is denoted as H3.

In the experiment, we show that our approach works well with any of these heat map losses. Though, these manually designed heat map losses might have different performances on different tasks and need careful network hyper-parameter tuning individually, the integral version (I1, I2, I3) of them would get prominent improvement and produce consistent results.

Heat Map and Joint Loss Combination. For the joint coordinate loss, we experimented with both *L1* and *L2* distances between the predicted joints and ground truth joints as loss functions. We found that *L1* loss works consistently better than *L2* loss. We thus adopt *L1* loss in all of our experiments.

Note that our integral regression can be trained with or without intermediate heat map losses. For the latter case, a variant of integral regression method is defined, denoted as I*. The network is the same, but the loss on heat maps is not used. The training supervision signal is only on joint, not on heat maps. In the experiment, we find that integral regression works well with or without heat map supervisions. The best performance depends on specific tasks. For example, for 2D task I1 obtains the best performance, while for 3D task I* obtains the best performance.

Image and Heat Map Resolutions. Due to the quantization error of heat map, high image and heat map resolutions are usually required for high localization accuracy. However, it is demanding for memory and computation especially for 3D heat map. In the experiment, we show that our approach is more robust to the image and heat map resolution variation. This makes it a better choice when the computational capabilities are restricted, in practical scenarios.

4 Datasets and Evaluation Metrics

Our approach is validated on three benchmark datasets.

Human3.6M [24] is the largest 3D human pose benchmark. The dataset is captured in controlled environment. It consists of 3.6 millions of video frames. 11 subjects (5 females and 6 males) are captured from 4 camera viewpoints, performing 15 activities. The image appearance of the subjects and the background is simple. Accurate 3D human joint locations are obtained from motion capture devices. For evaluation, many previous works [4,8,25,31,32,37,41,44,46,51,53,54,56] use the mean per joint position error (*MPJPE*). Some works [4,8,32,41,51,54] firstly align the predicted 3D pose and ground truth 3D pose with a rigid transformation using *Procrustes Analysis* [18] and then compute MPJPE. We call this metric *PA MPJPE*.

MPII [3] is the benchmark dataset for single person 2D pose estimation. The images were collected from YouTube videos, covering daily human activities with complex poses and image appearances. There are about $25k$ images. In total, about $29k$ annotated poses are for training and another $7k$ are for testing. For evaluation, Percentage of Correct Keypoints (PCK) metric is used. An estimated keypoint is considered correct if its distance from ground truth keypoint

is less than a fraction α of the head segment length. The metric is denoted as PCKh@α. Commonly, PCKh@0.5 metric is used for the benchmark [2]. In order to evaluate under high localization accuracy, which is also the strength of regression methods, we also use PCKh@0.1 and AUC (area under curve, the averaged PCKh when α varies from 0 to 0.5) metrics.

The *COCO* Keypoint Challenge [28] requires "in the wild" multi-person detection and pose estimation in challenging, uncontrolled conditions. The COCO train, validation, and test sets, containing more than 200k images and 250k person instances labeled with keypoints. 150k instances of them are publicly available for training and validation. The COCO evaluation defines the object keypoint similarity (OKS) and uses the mean average precision (AP) over 10 OKS thresholds as main competition metric [1]. The OKS plays the same role as the IoU in object detection. It is calculated from the distance between predicted points and ground truth points normalized by the scale of the person.

5 Experiments

Training. Our training and network architecture is similar for all the three datasets. ResNet [20] and HourGlass [33] (ResNet and HourGlass on Human3.6M and MPII, ResNet-101 on COCO) are adopted as the backbone network. ResNet is pre-trained on ImageNet classification dataset [16]. HourGlass is trained from scratch. Normal distribution with 1e−3 standard deviation is used to initialize the HourGlass and head network parameters.

The head network for heat map is fully convolutional. It firstly use deconvolution layers (4×4 kernel, stride 2) to upsample the feature map to the required resolution (64×64 by default). The number of output channels is fixed to 256 as in [19]. Then, a 1×1 conv layer is used to produce K heat maps. Both heat map baseline and our integral regression are based on this head network.

We also implement a most widely used regression head network as a regression baseline for comparison. Following [7,42,55,56], first an average pooling layer reduces the spatial dimensionality of the convolutional features. Then, a fully connected layer outputs $3K$ ($2K$) joint coordinates. We denote our regression baseline as R1 (R for regression).

We use a simple multi-stage implementation based on ResNet-50, the features from conv3 block are shared as input to all stages. Each stage then concatenates this feature with the heat maps from the previous stage, and passes through the conv4 and conv5 blocks to generate its own deep feature. The heat map head is then appended to output heat maps, supervised with the ground truth and losses. Depending on the loss function used on the heat map, this multi-stage baseline is denoted as MS-H1(2, 3).

MxNet [9] is used for implementation. Adam is used for optimization. The input image is normalized to 256×256. Data augmentation includes random translation ($\pm 2\%$ of the image size), scale ($\pm 25\%$), rotation ($\pm 30^{circ}$) and flip. In all experiments, the base learning rate is 1e−3. It drops to 1e−5 when the loss on the validation set saturates. Each method is trained with enough number of

iterations until performance on validation set saturates. Mini-batch size is 128. Four GPUs are used. Batch-normalization [23] is used. Other training details are provided in individual experiments.

For integral regression methods (I1, I2, I3, and their multi-stage versions), the network is pre-trained only using heat map loss (thus their H versions) and then, only integral loss is used. We found this training strategy working slightly better than training from scratch using both losses.

5.1 Experiments on MPII

Since the annotation on MPII test set is not available, all our ablation studies are evaluated on an about $3k$ validation set which is separated out from the training set, following previous common practice [33]. Training is performed on the remaining training data.

Table 1. Comparison between methods using heat maps, direct regression, and integral regression on MPII validation set. Backbone network is ResNet-50. The performance gain is shown in the subscript

Metric	R1	H1	H2	H3	I*	I1	I2	I3
@0.5	84.6	86.8	86.4	83.0	86.0$_{\uparrow 1.4}$	**87.3**$_{\uparrow 0.5}$	86.9$_{\uparrow 0.5}$	86.6$_{\uparrow 3.6}$
@0.1	25.0	17.2	17.6	12.6	28.3$_{\uparrow 3.3}$	29.3$_{\uparrow 12.1}$	**29.7**$_{\uparrow 12.1}$	29.1$_{\uparrow 16.5}$
AUC	54.1	52.9	53.1	46.3	56.6$_{\uparrow 2.5}$	**58.3**$_{\uparrow 5.4}$	**58.3**$_{\uparrow 5.2}$	57.7$_{\uparrow 11.4}$

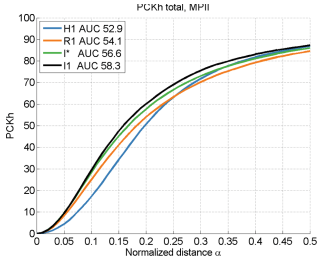

Fig. 2. Curves of PCKh@α of different methods while α varies from 0 to 0.5.

Effect of Integral Regression. Table 1 presents a comprehensive comparison. We first note that all integral regression methods (I1, I2, I3) clearly outperform their heat map based counterpart (H1, H2, H3). The improvement is especially significant on PCKh@0.1 with high localization accuracy requirement. For example, the improvement of I1 to H1 is +0.5 on PCKh@0.5, but +12.1 on PCKh@0.1. The overall improvement on AUC is significant (+5.4). Among the three heat map based methods, H3 performs the worst. After using integral regression (I3), it is greatly improved, eg., AUC from 46.3 to 57.7 (+11.4). Such results show that *joint training of heat maps and joint is effective.* The significant improvement on localization accuracy (PCKh@0.1 metric) is attributed to the joint regression representation.

Surprisingly, I* performs quite well. It is only slightly worse than I1/I2/I3 methods. It outperforms H1/H2/H3 on PCKh@0.1 and AUC, thanks to its regression representation. It outperforms R1, indicating that integral regression is better than direct regression, as both methods use exactly the same supervision and almost the same network (actually R1 has more parameters).

Fig. 3. Example results of regression baseline (R1), detection baseline (H1) and integral regression (I1).

From the above comparison, we can draw two conclusions. First, integral regression using an underlying heat map representation is effective (I* > H, I* > R). It works even without supervision on the heat map. Second, joint training of heat maps and joint coordinate prediction combines the benefits of two paradigms and works best (I > H, R, I*).

As H3 is consistently worse than the other two and hard to implement for 3D, it is discarded in the remaining experiments. As H1 and I1 perform best in 2D pose, they are used in the remaining 2D (MPII and COCO) experiments. Figure 2 further shows the PCKh curves of H1, R1, I* and I1 for better illustration.

Figure 3 shows some example results. Regression prediction (R1) is usually not well aligned with local image features like corners or edges. On the contrary, detection prediction (H1) is well aligned with image feature but hard to distinguish locally similar patches, getting trapped into local maximum easily. Integral regression (H1) shares the merits of both heat map representation and joint regression approaches. It effectively and consistently improves both baselines.

Effect of Resolution. Table 2 compares the results using two input image sizes and two output heat map sizes.

Not surprisingly, using large image size and heat map size obtains better accuracy, under all cases. However, integral regression (I1) is much less affected by the resolution than heat map based method (H1). It is thus a favorable choice when computational complexity is crucial and a small resolution is in demand.

For example, when heat map is downsized by half on image size 256 (a to b), 1.1 G FLOPs (relative 15%) is saved. I1 only drops 0.6 in AUC while H1 drops 4.8. This gap is more significant on image size 128 (c to d). 0.3G FLOPs (relative 17%) is saved. I1 only drops 3.5 in AUC while H1 drops 12.5.

When image is downsized by half (b to d), 4.7 G FLOPs is saved (relative 76%). I1 only drops 11.1 in AUC while H1 drops 18.8.

Thus, we conclude that *integral regression significantly alleviates the problems of quantization error or needs of large resolution in heat map based methods.*

Table 2. For two methods (H1/I1), two input image→feature map (**f**) resolutions, and two heat map sizes (using either 3 or 2 upsampling layers), the performance metric (mAP@0.5, map@0.1, AUC), the computation (in FLOPs) and the amount of network parameters. Note that setting (*b*) is used in all other experiments

Size	×2, ×2, ×2	×2, ×2	Size	×2, ×2, ×2	×2, ×2
256 → 8	(*a*) → 16 → 32 → 64	(*b*) → 16 → 32	128 → 4	(*c*) → 8 → 16 → 32	(*d*) → 8 → 16
H1	86.7/28.0/57.7	86.8/17.2/52.9		81.6/13.6/46.6	75.4/5.6 /34.1
I1	86.6/32.1/58.9	87.3/29.3/58.3		83.2/20.6/50.7	80.9/16.1/47.2
FLOPs	7.3G	6.2G		1.8G	1.5G
params	26M	26M		26M	26M

Effect of Network Capacity. Table 3 shows results using different backbones on two methods. While all methods are improved using a network with large capacity, integral regression I1 keeps outperforming heat map based method H1.

While a large network improves accuracy, a high complexity is also introduced. Integral regression I1 using ResNet-18 already achieves accuracy comparable with H1 using ResNet-101. This makes it a better choice when a small network is in favor, in practical scenarios.

Table 3. PCKh@0.5, PCKh@0.1 and AUC metrics (top) of three methods, and model complexity (bottom) of three backbone networks. Note that ResNet-50 is used in all other experiments

	ResNet-18	ResNet-50	ResNet-101
H1	85.5/15.7/50.8	86.8/17.2/52.9	87.3/17.3/53.3
I1	86.0/25.7/55.6	87.3/29.3/58.3	87.9/30.3/59.0
FLOPs	2.8G	6.2G	11.0G
params	12M	26M	45M

Table 4. PCKh@0.5, PCKh@0.1 and AUC metrics of a multi-stage network with and without integral regression

Stage	MS-H1	MS-I1
1	86.8/17.2/52.9	87.3/29.3/58.3
2	86.9/17.6/53.4	87.7/32.0/59.5
3	87.1/17.8/53.7	87.8/32.4/59.9
4	87.4/17.8/54.0	88.1/32.3/60.1

Effect in Multi-stage. Table 4 shows the results of our multi-stage implementation with or without using integral regression. There are two conclusions. First, integral regression can be effectively combined with a multi-stage architecture and performance improves as stage increases. Second, integral regression outperforms its heat map based counterpart on all stages. Specifically, MS-I1 stage-2 result 87.7 is already better than MS-H1 state-4 result 87.4.

Conclusions. From the above ablation studies, we can conclude that *effectiveness of integral regression is attributed to its representation.* It works under different heat map losses (H1, H2, H3), different training (joint or not), different resolution, and different network architectures (depth or multi-stage). Consistent yet even stronger conclusions can also be derived from COCO benchmark in Sect. 5.2 and 3D pose benchmarks in Sect. 5.3.

Table 5. Comparison to state-of-the-art works on MPII

Method (heat map based)	Tompson [47]	Raf [39]	Wei [49]	Bulat [5]	Newell [33]	Yang [50]	Ours		
							H1	MS-H1	HG-H1
Mean (PCKh@0.5)	82.0	86.3	88.5	89.7	90.9	92.0	89.4	89.8	90.4
Method (regression)	Carreira [7]		Sun [42]		R1 (Ours)		I1	MS-I1	HG-I1
Mean (PCKh@0.5)	81.3		86.4		87.0		90.0	90.7	91.0

Result on the MPII Test Benchmark. Table 5 summarizes the results of our methods, as well as state-of-the-art methods. In these experiments, our training is performed on all $29k$ training samples. We also adopt the flip test trick as used in [33]. Increasing the training data and using flip test would increase about 2.5 mAP@0.5 from validation dataset to test dataset.

We first note that our baselines have good performance, indicating they are valid and strong baselines. H1 and MS-H1 in the heat map based section has 89.4 and 89.8 PCKh, respectively, already comparable to many multi-stage methods that are usually much more complex. R1 in regression section is already the best performing regression method.

Our integral regression further improves both baselines (I1>H1, MS-I1>MS-H1, 4 stages used) and achieves results competitive with other methods.

We also re-implement the HourGlass architecture [33], denoted as HG-H1. Consistent improvement is observed using integral regression HG-I1. While the accuracy of our approach is slightly below the state-of-the-art, we point out that the recent leading approaches [10,12,13,50] are all quite complex, making direct and fair comparison with these works difficult. Integral regression is simple, effective and can be combined with most other heat map based approaches, as validated in our baseline multi-stage and the HourGlass experiments. Combination with these approaches is left as future work.

Table 6. COCO **test-dev** results

	Backbone	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}	AP^{kp}_{M}	AP^{kp}_{L}
CMU-Pose [6]		61.8	84.9	67.5	57.1	68.2
Mask R-CNN [19]	ResNet-50-FPN	63.1	87.3	68.7	57.8	71.4
G-RMI [36]	ResNet-101(353×257)	64.9	85.5	71.3	62.3	70.0
Ours: H1	ResNet-101(256×256)	66.3	**88.4**	74.6	62.9	72.1
Ours: I1	ResNet-101(256×256)	**67.8**	88.2	**74.8**	**63.9**	**74.0**

5.2 Experiments on COCO

Person Box Detection. We follow a two-stage top-down paradigm similar as in [36]. For human detection, we use Faster-RCNN [40] equipped with deformable convolution [15]. We uses Xception [11] as the backbone network. The box detection AP on COCO test-dev is 0.49. For reference, this number in [36] is 0.487. Thus, the person detection performance is similar.

Following [36], we use the keypoint-based Non-Maximum-Suppression (NMS) mechanism building directly on the OKS metric to avoid duplicate pose detections. We also use the pose rescoring technique [36] to compute a refined instance confidence estimation that takes the keypoint heat map score into account.

Pose Estimation. We experimented with heat map based method (H1) and our integral regression methods (I1). All settings are the same as experiments on MPII, except that we use ResNet-101 as our backbone and use 3 deconvolution layers (4×4 kernel, stride 2) to upsample the feature maps.

Results. Table 6 summarizes the results of our methods, as well as state-of-the-art on COCO test-dev dataset. Our experiments are performed on COCO training data, no extra data is added. The baseline model (H1) is a one-stage ResNet-101 architecture. Our baseline model H1 is already superior to the state of the art top-down method [36]. Our integral regression further increases AP^{kp} by 1.5 points and achieves the state-of-the-art result.

5.3 Experiments on Human3.6M

In the literature, there are two widely used evaluation protocols. They have different training and testing data split.

Protocol 1. Six subjects (S1, S5, S6, S7, S8, S9) are used in training. Evaluation is performed on every 64th frame of Subject 11. *PA MPJPE* is used for evaluation.

Protocol 2. Five subjects (S1, S5, S6, S7, S8) are used in training. Evaluation is performed on every 64th frame of subjects (S9, S11). *MPJPE* is used for evaluation.

Two training strategies are used on whether use extra 2D data or not. *Strategy 1* only use Human3.6M data for training. For integral regression, we use Eq. (4). *Strategy 2* mix Human3.6M and MPII data for training, each mini-batch consists of half 2D and half 3D samples, randomly sampled and shuffled. In this strategy, we use the two-step integral function Eqs. (5) and (6) so that we can add 2D data on both heat map and joint losses for training as explained in Sect. 2.1.

Effect of Integral Regression. Table 7 compares the integral regression (I*,I1,I2) with corresponding baselines (R1, H1,H2) under two training strategies. Protocol 2 is used. Backbone is ResNet50. We observe several conclusions.

First, integral regression significantly improves the baselines in both training strategies. Specifically, without using extra 2D data, the integral regression (I*, I1, I2) improves (R1, H1, H2) by 6.0%, 13.2%, 17.7% respectively. I2 outperforms

Table 7. Comparison between methods using heat maps, direct regression, and integral regression. Protocol 2 is used. Two training strategies are investigated. Backbone network is ResNet-50. The relative performance gain is shown in the subscript

Training data strategy	R1	H1	H2	I*	I1	I2
Strategy 1	106.6	99.5	80.4	$100.2_{\downarrow 6.0\%}$	$86.4_{\downarrow 13.2\%}$	$\mathbf{66.2}_{\downarrow 17.7\%}$
Strategy 2	56.2	63.6	59.3	$\mathbf{49.6}_{\downarrow 11.7\%}$	$52.7_{\downarrow 17.1\%}$	$52.4_{\downarrow 11.6\%}$

Table 8. Comparison with Coarse-to-Fine Volumetric Prediction [37] trained only on Human3.6M. Protocol 2 is used. Evaluation metric is MPJPE. d_i denotes the z-dimension resolution for the supervision provided at the i-th hourglass component. Our I1 wins at both stages

Network architecture (HourGlass [33])	Coarse-to-fine. [37]	Ours H1	Ours I1
One stage ($d = 64$)	85.8	85.5	**78.7**
Two stage ($d_1 = 1, d_2 = 64$)	69.8	68.0	**64.1**

all previous works in this setting. When using extra 2D data, the baselines have already achieved very competitive results. Integral regression further improves them by 11.7%, 17.1%, 11.6%, respectively. I* achieves the new state-of-the-art in this setting and outperforms previous works by large margins, see Table 10(B). Second, all methods are significantly improved after using MPII data. This is feasible because of integral formulation Eqs. (5) and (6) generates x, y, z predictions individually and keep differentiable.

Effect of Backbone Network. [37] is the only previous work using 3D heat map representation. They use a different backbone network, multi-stage HourGlass. In Table 8, we follow exactly the same practice as in [37] for a fair comparison using this backbone network. Only Human3.6M data is used for training and Protocol 2 is used for evaluation.

We have several observations. First, our baseline implementation H1 is strong enough that is already better than [37] at both stages. Therefore, it serves as a competitive reference. Second, our integral regression I1 further improves H1 at both stages by 6.8 mm (relative 8.0%) at stage 1 and 3.9 mm (relative 5.7%) at stage 2. We can conclude that the integral regression also works effectively with HourGlass and multi-stage backbone on the 3D pose problem and our two-stage I1 sets the new state-of-the-art in this setting, see Table 11.

Effect of Resolution. Table 9 investigates the effect of input image and heat map resolution on 3D problem. We can also have similar conclusions as in Table 2. Integral regression (I2) is much less affected by the resolution than heat map based method (H2). It is thus a favorable choice when computational complexity is crucial and a small resolution is in demand.

Table 9. For two methods (H2/I2), two input image → feature map (**f**) resolutions, and two heat map sizes (using either 3 or 2 upsampling layers). Strategy 2 and Protocol 2 are used. Backbone network is ResNet-50

Size	×2, ×2, ×2	×2, ×2	Size	×2, ×2, ×2	×2, ×2
256 → 8	(*a*) → 16 → 32 → 64	(*b*) → 16 → 32	128 → 4	(*c*) → 8 → 16 → 32	(*d*) → 8 → 16
H2	59.3	61.5		66.6	86.4
I2	52.4	51.7		57.1	60.9

Table 10. Comparison with previous work on Human3.6M. All methods used extra 2D training data. Ours use MPII data in the training. Methods in Group A and B use Protocol 1 and 2, respectively. Ours is the best single-image method under both scenarios. Methods with * exploit temporal information and are complementary to ours. We even outperform them in Protocol 2

Method (A, Pro. 1)	Hossain [21]*	Dabral [14]*	Yasin [51]	Rogez [41]	Chen [8]	Moreno [32]	Zhou [54]	Martinez [30]	Kanazawa [26]	Sun [42]	Fang [17]	Ours
PA MPJPE	42.0	36.3	108.3	88.1	82.7	76.5	55.3	47.7	56.8	48.3	45.7	**40.6**

Method (B, Pro. 2)	Hossain [21]*	Dabral [14]*	Chen [8]	Tome [46]	Moreno [32]	Zhou [54]	Jahangiri [25]	Mehta [31]	Martinez [30]	Kanazawa [26]	Fang [17]	Sun [42]	Ours
MPJPE	51.9	52.1	114.2	88.4	87.3	79.9	77.6	72.9	62.9	88.0	60.4	59.1	**49.6**

Table 11. Comparison with previous work on Human3.6M. Protocol 2 is used. No extra training data is used. Ours is the best

Method	Zhou [53]	Tekin [44]	Xingyi [56]	Sun [42]	Pavlakos [37]	Ours
MPJPE	113.0	125.0	107.3	92.4	71.9	**64.1**

For example, when heat map is downsized by half on image size 256 (*a* to *b*). I2 even gets slightly better while H2 drops 2.2 mm on MPJPE. This gap is more significant on image size 128 (*c* to *d*). I2 only drops 3.8 mm in MPJPE while H2 drops 19.8 mm. When image is downsized by half (b to d). I2 only drops in 9.2 mm on MPJPE while H2 drops 24.9 mm.

Consistent yet even stronger conclusions are derived on 3D task, compared with Table 2 on 2D task.

Comparison with the State of the Art. Previous works are abundant with different experiment settings and fall into three categories. They are compared to our method in Table 10(A), (B) and Table 11 respectively.

Our approach is the best single-image method that outperforms previous works by large margins. Specifically, it improves the state-of-the-art, by 5.1 mm (relative 11.2%) in Table 10(A), 9.5 mm (relative 16.1%) in Table 10(B), and

7.8 mm (relative 10.8%) in Table 11. Note that Dabral et al. [14] and Hossain et al. [21] exploit temporal information and are complementary to our approach. Nevertheless, ours is already very close to them in Table 10(A) and even better in Table 10(B).

6 Conclusions

We present a simple and effective integral regression approach that unifies the heat map representation and joint regression approaches, thus sharing the merits of both. Solid experiment results validate the efficacy of the approach. Strong performance is obtained using simple and cheap baseline networks, making our approach a favorable choice in practical scenarios. We apply the integral regression on both 3D and 2D human pose estimation tasks and push the very state-of-the-art on MPII, COCO and Human3.6M benchmarks.

References

1. COCO Leader Board. http://cocodataset.org
2. MPII Leader Board. http://human-pose.mpi-inf.mpg.de
3. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: new benchmark and state of the art analysis. In: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition, pp. 3686–3693 (2014)
4. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it SMPL: automatic estimation of 3D human pose and shape from a single image. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 561–578. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_34
5. Bulat, A., Tzimiropoulos, G.: Human pose estimation via convolutional part heatmap regression. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 717–732. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_44
6. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. arXiv preprint arXiv:1611.08050 (2016)
7. Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4733–4742 (2016)
8. Chen, C.H., Ramanan, D.: 3D human pose estimation = 2D pose estimation + matching. arXiv preprint arXiv:1612.06524 (2016)
9. Chen, T., et al.: MxNet: a flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274 (2015)
10. Chen, Y., Shen, C., Wei, X.S., Liu, L., Yang, J.: Adversarial PoseNet: a structure-aware convolutional network for human pose estimation. arXiv preprint arXiv:1705.00389 (2017)
11. Chollet, F.: Xception: deep learning with depthwise separable convolutions. arXiv preprint arXiv:1610.02357 (2016)
12. Chou, C.J., Chien, J.T., Chen, H.T.: Self adversarial training for human pose estimation. arXiv preprint arXiv:1707.02439 (2017)
13. Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A.L., Wang, X.: Multi-context attention for human pose estimation. arXiv preprint arXiv:1702.07432 (2017)

14. Dabral, R., Mundhada, A., Kusupati, U., Afaque, S., Jain, A.: Structure-aware and temporally coherent 3D human pose estimation. arXiv preprint arXiv:1711.09250 (2017)
15. Dai, J., et al.: Deformable convolutional networks. arXiv preprint arXiv:1703.06211 (2017)
16. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255. IEEE (2009)
17. Fang, H.S., Xu, Y., Wang, W., Liu, X., Zhu, S.C.: Learning pose grammar to encode human body configuration for 3D pose estimation. In: AAAI (2018)
18. Gower, J.C.: Generalized procrustes analysis. Psychometrika **40**(1), 33–51 (1975)
19. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: International Conference on Computer Vision (2017)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
21. Hossain, M.R.I., Little, J.J.: Exploiting temporal information for 3D pose estimation. arXiv preprint arXiv:1711.08585 (2017)
22. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: DeeperCut: a deeper, stronger, and faster multi-person pose estimation model. In: European Conference on Computer Vision. pp. 34–50. Springer (2016)
23. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
24. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6m: large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1325–1339 (2014)
25. Jahangiri, E., Yuille, A.L.: Generating multiple hypotheses for human 3D pose consistent with 2D joint detections. arXiv preprint arXiv:1702.02258 (2017)
26. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. arXiv preprint arXiv:1712.06584 (2017)
27. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. J. Mach. Learn. Res. **17**(1), 1334–1373 (2016)
28. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
29. Luvizon, D.C., Tabia, H., Picard, D.: Human pose regression by combining indirect part detection and contextual information. arXiv preprint arXiv:1710.02322 (2017)
30. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3D human pose estimation. arXiv preprint arXiv:1705.03098 (2017)
31. Mehta, D., Rhodin, H., Casas, D., Sotnychenko, O., Xu, W., Theobalt, C.: Monocular 3D human pose estimation in the wild using improved CNN supervision. arXiv preprint arXiv:1611.09813 (2016)
32. Moreno-Noguer, F.: 3D human pose estimation from a single image via distance matrix regression. arXiv preprint arXiv:1611.09010 (2016)
33. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
34. Nibali, A., He, Z., Morgan, S., Prendergast, L.: Numerical coordinate regression with convolutional neural networks. arXiv preprint arXiv:1801.07372 (2018)

35. Nie, B.X., Wei, P., Zhu, S.C.: Monocular 3D human pose estimation by predicting depth on joints. In: Proceedings of IEEE International Conference on Computer Vision, pp. 3447–3455 (2017)
36. Papandreou, G., et al.: Towards accurate multi-person pose estimation in the wild. arXiv preprint arXiv:1701.01779 (2017)
37. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3D human pose. arXiv preprint arXiv:1611.07828 (2016)
38. Pishchulin, L., et al.: DeepCut: joint subset partition and labeling for multi person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4929–4937 (2016)
39. Rafi, U., Kostrikov, I., Gall, J., Leibe, B.: An efficient convolutional network for human pose estimation. In: BMVC, vol. 1, p. 2 (2016)
40. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
41. Rogez, G., Schmid, C.: MoCap-guided data augmentation for 3D pose estimation in the wild. In: Advances in Neural Information Processing Systems, pp. 3108–3116 (2016)
42. Sun, X., Shang, J., Liang, S., Wei, Y.: Compositional human pose regression. In: International Conference on Computer Vision (2017)
43. Tekin, B., Marquez Neila, P., Salzmann, M., Fua, P.: Learning to fuse 2D and 3D image cues for monocular body pose estimation. In: International Conference on Computer Vision (ICCV). No. EPFL-CONF-230311 (2017)
44. Tekin, B., Rozantsev, A., Lepetit, V., Fua, P.: Direct prediction of 3D body poses from motion compensated sequences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 991–1000 (2016)
45. Thewlis, J., Bilen, H., Vedaldi, A.: Unsupervised learning of object landmarks by factorized spatial embeddings. In: Proceedings of ICCV (2017)
46. Tome, D., Russell, C., Agapito, L.: Lifting from the deep: convolutional 3D pose estimation from a single image. arXiv preprint arXiv:1701.00295 (2017)
47. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 648–656 (2015)
48. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems, pp. 1799–1807 (2014)
49. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4724–4732 (2016)
50. Yang, W., Li, S., Ouyang, W., Li, H., Wang, X.: Learning feature pyramids for human pose estimation. In: The IEEE International Conference on Computer Vision (ICCV), vol. 2 (2017)
51. Yasin, H., Iqbal, U., Kruger, B., Weber, A., Gall, J.: A dual-source approach for 3D pose estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4948–4956 (2016)
52. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: LIFT: learned invariant feature transform. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 467–483. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_28

53. Zhou, X., Zhu, M., Leonardos, S., Derpanis, K.G., Daniilidis, K.: Sparseness meets deepness: 3D human pose estimation from monocular video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4966–4975 (2016)
54. Zhou, X., Zhu, M., Pavlakos, G., Leonardos, S., Derpanis, K.G., Daniilidis, K.: MonoCap: monocular human motion capture using a cnn coupled with a geometric prior. arXiv preprint arXiv:1701.02354 (2017)
55. Zhou, X., Huang, Q., Sun, X., Xue, X., Wei, Y.: Towards 3D human pose estimation in the wild: a weakly-supervised approach. In: International Conference on Computer Vision (2017)
56. Zhou, X., Sun, X., Zhang, W., Liang, S., Wei, Y.: Deep kinematic pose regression. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 186–201. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_17