3D Human Pose Estimation using Iterative Conditional Squeeze and Excitation Networks


Published in:
IEEE Transactions on Cybernetics

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Publisher rights
©2020 IEEE.
This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.
Abstract—We propose a new method for single-camera real-world 3D human pose estimation. Our method uses multi-task training together with iterative pose refinement using a novel conditional attention mechanism. For iterative pose refinement, the output of each convolutional layer is conditioned on the latest pose estimate, using a Conditioned Squeeze-and-Excitation network architecture that incorporates novel feedback connections. Multi-task training on both an in-the-wild 2D pose data-set and a controlled 3D pose dataset allows for real-world 3D pose estimation without the need for a large-scale in-the-wild 3D pose dataset, which is unavailable. Experiments are performed on several real-world datasets, as well as the Human 3.6 Million and HumanEva-I datasets, to show that the combined attention mechanism, iterative refinement scheme and multi-task training allow us to achieve robust and competitive performance with only a simple network architecture. In addition, we show that our method is efficient enough to run on commodity hardware, producing pose estimates in real-time.

Index Terms—Conditioning Mechanism, Deep Neural Network, Human Pose Estimation

I. INTRODUCTION

Estimation of human 3D pose from a single image is a challenging problem with potential applications in areas ranging from security to video-games and human-computer interaction. The challenging aspect of this problem comes from the way the solution is underdetermined given the available evidence i.e., the full 3D position of each part of each body part must be estimated from an ambiguous and noisy 2D image. In general, the problem of recovering full 3D pose information from a 2D images does not have a unique solution, however, in the case of human pose estimation, the solution space is reduced due to the physical constraints that must be obeyed by the body [18].

In recent years the emergence of deep learning techniques, and the availability of large labelled datasets have quickly increased the accuracy of human pose estimation. In general, these approaches aim to learn a solution that combines the extraction of low-level image evidence together with high-level reasoning about how this evidence should be interpreted, given the physical constraints of the human body. This is usually done by learning from a large dataset containing person images and corresponding pose information.

In this paper we propose a novel neural network architecture incorporating a conditional-attention mechanism and iterative feedback, that directly regresses both 2D and 3D pose from a single image. Our approach is based on conditioning the output of each convolutional layer on both self-attention, and on the estimated pose from the previous iteration. While such self-attention mechanisms have been recently proposed to increase the expressive power of convolutional networks [11], [23], our conditional extension, along with iterative pose refinement, helps to reinforce the pose constraints of the human body during the low-level extraction of image evidence in an iterative manner. Furthermore, we propose the use of multi-task learning of both the 2D and 3D pose in order to improve the robustness of our system to in-the-wild scenarios. We show that our simple and novel architecture is capable of achieving competitive pose estimation error in controlled situations, where the pose estimate from our method improves with increasing iterations, and that is able to produce robust and realistic estimations in realistic settings. Our method has been tested on two challenging controlled datasets, the Human 3.6M dataset (H36M) [13] and the HumanEva-I dataset [29] and it achieves competitive results in each, as well as in several realistic datasets such as the MPII Human Pose Dataset (MPII) [1] and Oxford Town Centre [3].

II. RELATED WORK

There are several broad categories of approaches to estimating 3D human pose from a single image. The first group of approaches use the solution to the simpler 2D pose estimation
problem to bootstrap a solution to 3D pose estimation. The task of 2D pose estimation from a single image has the advantage that there is less ambiguity for visible body parts, which makes labelling visible joint locations in image space easier. Therefore this task has been extensively studied and there exist several datasets dedicated to 2D pose estimation [1], [14]. Human body joint locations in 2D image space can be predicted by direct regression to image coordinates [37], or by predicting a probabilistic heat-map for each joint, from which the image-space locations of the joints can be directly inferred [20].

Given a 2D pose estimate, obtained using one of the above methods, a constrained optimization process can then be used to find a 3D pose estimate compatible with the observed 2D evidence. Such an optimization problem can be tackled using a generative model, either as a part of separate optimization algorithm [41], [4] or using an inference network [30]. This method is explicitly used by [17], which first estimates 2D joint locations using existing methods, then learns to regress directly from 2D pose to the corresponding 3D pose. A related approach taken by [26], first proposes several different noisy pose estimates per-image, then regresses the most compatible joint 2D and 3D estimate given the earlier proposals. This family of approaches can sometimes suffer from the difficulty of jointly optimizing the separate 2D and 3D pose estimation processes in an end-to-end manner. The constrained optimization processes used can also be computationally costly, or can impose hard constraints regarding the kinds of pose and activities that can be recovered.

In contrast, a different family of methods exist for directly estimating 3D pose from a single image with no intermediate steps. These methods have the advantage of simplicity as there are no intermediate representations or separate algorithms to optimize. However, performing 3D pose estimation directly can be more difficult due to the lack of constraints. Recently, [32] explored an approach that fuses heat-maps and regression into a single network architecture. [31] proposes a new loss function for pose regression that reformulates the problem in terms of predicting bones, rather than joints, and achieves excellent results. [22] extends the idea of predicting a heat-map for each 2D joint location into 3D space by predicting a volumetric 3D heat-map, from which the 3D pose can be inferred.

Traditional neural networks have been feed-forward, once trained they simply apply a fixed set of transformations to their input. In recent years, the attention mechanism has emerged as a way for neural networks to change their behaviour based on their input. A self-attention mechanism was first proposed for neural machine translation [2], as a way for the decoder network to condition the representation of the input sentence on the decoder’s state. In effect, the decoder could re-examine the input sentence multiple times, each time focusing on information relevant to producing the next word of the translation. This mechanism was also applied to producing captions describing the contents of images [8]. More recently self-attention has emerged as a way of achieving similar results, while avoiding the quadratic complexity of traditional soft-attention mechanisms [38].

We propose a novel attention-based network architecture for iterative pose refinement that builds on the conditional-attention mechanism in [11]. Following on from previous iterative refinement schemes for human pose estimation [6], [7], [20], we extend the squeeze-and-excitation attention mechanism [11], by including additional task specific information when computing the attention weights. Specifically, our proposed architecture conditions the output of each convolutional layer both on information about the image content stored in the feature maps, and on the pose estimate from the previous iteration. A related approach is used in the FiLM network architecture [23], for visual question answering, which conditions the output of each convolutional layer on the question being asked. We present a novel extension of the above methods called conditional-squeeze-and-excitation, that enables this attention mechanism to be extended to perform iterative pose refinement. We show that this method can produce results comparable with the state-of-the-art on the Human 3.6 Million [13] and HumanEva-I [29] datasets while using a much simpler network design.

III. METHOD

In this paper we propose a novel convolutional network architecture, trained using multi-task learning, that is capable of real-time 3D human pose estimation in uncontrolled environments.

The proposed network performs regression on a person image to produce an estimate of both the 3D and 2D coordinates of each body part in world-space and image-space respectively. Our method makes use of a number of conditional squeeze-and-excite block (CSE-blocks), which will be discussed in Section III-C. The blocks condition the output of each convolutional layer on both a self-attention mechanism and on the current best pose estimate. Iterative refinement is used to obtain a pose estimate, where the predictions from the previous stage are used as input to condition to CSE-blocks in the next iteration. The CSE-blocks and iterative refinement process allow the network to modify its internal representation based on the current pose estimate.

To produce a network capable of real-world pose estimation in uncontrolled conditions, we make use of multi-task learning. The network is trained on several pose datasets: a very large, but controlled 3D pose dataset with limited appearance variability, and a smaller, but visually diverse 2D dataset. We show that by using multi-task learning, where the network is trained to perform both 3D and 2D pose estimation on both datasets, the network can generalise to real-world 3D pose estimation, despite not having access to a diverse real-world 3D pose dataset. We further show that our network is efficient enough to be run in real-time on commodity hardware.

A. Convolutional Network Architecture

The underlying network architecture we use is based on a residual network (ResNet) [10]. After the first convolutional layer of each residual block we add a conditional SE-layer (see Section III-B), as shown in Figure 2.
The family of residual network architectures provides the ability to create larger or smaller residual networks with a similar structure. Each member of the ResNet family has a differing number of residual blocks, containing convolutional filters, and hence a differing number of trainable parameters. This allows a network architecture to be selected based on the task and amount of training-data available.

In common with recent networks for object classification [10], [12] we employ a fully-convolutional architecture. In a fully-convolutional architecture, the feature maps of the final convolutional layer are passed through a global mean-pooling operation. This produces a vector, \( m \in \mathbb{R}^{C'} \), where \( C' \) is the number of feature maps in the final convolutional layer and element \( m_c \) is the global mean of feature map \( F_i \). As the number of feature maps output by the final convolutional layer is not equal to the number of joint positions we wish to predict, vector \( m \) is passed through a fully-connected layer to produce the pose prediction. The advantage of using a fully-convolutional architecture is that the total number of parameters is significantly reduced, compared to architectures with a fully-connected layer after the final convolution, and hence the network is less prone to over-fitting.

### B. Squeeze and Excite Layer

The standard SE-layer can be considered a type self-attention mechanism, where the feature-maps of a convolutional layer are re-weighted based on the mean values of all the feature-maps in that layer. This allows the SE-block to suppress or emphasize each feature-map, and hence suppress or emphasize a particular feature, by multiplying the corresponding feature-maps by an appropriate scalar. In this section we will explain the standard SE-layer and in Section III-C we will show how this layer can be extended as a conditional mechanism.

An SE-layer accepts a set of feature maps \( F \in \mathbb{R}^{H \times W \times C} \) as input, where feature map \( F_c \in \mathbb{R}^{W \times H} \) is the \( c \)’th feature map, there are a total of \( C \) feature maps, and each is of width \( W \) and height \( H \). The self-attention mechanism computes a vector \( k \in \mathbb{R}^C \) by taking the mean of each feature map i.e. the value at the \( c \)’th position in \( k \) is defined as:

\[
k_c = \frac{1}{HW} \sum_h \sum_w F_c(h, w)
\]

where \( F_c(h, w) \) is the value at position \((h, w)\) in feature map \( F_c \). The vector \( k \) is then passed through a small multi-layer-perceptron (MLP) consisting of two fully-connected layers with ReLU activations after the first layer and Sigmoid activations after the second layer. Hence the values of all the output neurons are between 0 and 1. The MLP calculating the self-attention weights and allows each feature map to influence all the other feature maps in the layer. The output of the MLP is a vector \( v \in \mathbb{R}^C \) i.e. equal in length to the number of input feature-maps. Self-attention is then applied by multiplying each of the input feature maps by the corresponding entry in \( v \) i.e.

\[
\hat{F}_c = F_c v_c
\]

(2) to produce a new, re-weighted set of feature maps \( \hat{F} \). The re-weighted feature maps are then added to the input of the residual block as is standard procedure the ResNet architecture [10].

### C. Conditional Squeeze and Excite Layer

In the standard SE-layer, described above, the self-attention mechanism is conditioned only on the mean values of the feature maps in the corresponding convolutional layer. The mean value of each feature-map encodes only limited information about image content. We propose to add a richer and higher-level source of information, and hence better guide the attention mechanism, by also providing the network output (the current pose estimate in this instance) to the SE-layer in each residual block.

Assuming we have available a pose estimate, we modify the architecture of the standard SE-layer to condition the attention mechanism on both the input feature-maps as well as on the current 2D and 3D pose estimate. This can be done by simply concatenating the pose estimate with the vector \( k \) (see Section III-B), containing the mean values of the input feature maps, to form the vector \( \hat{k} \). The conditional SE-block (CSE-block) now functions as before, and \( \hat{k} \) is passed through a small two-layer MLP to produce vector \( \hat{v} \) of length \( C \). Note that the size of the input layer of the MLP is increased to accommodate the longer vector \( \hat{k} \). As before, the input feature maps are scaled by the corresponding values in \( \hat{v} \) to calculate the output feature maps.

In our proposed network architecture a CSE-block is used at the start of each residual block, meaning that complex attention patterns can potentially be learned by combining the attention mechanisms from many layers together. The CSE-block is illustrated in Fig. 3.

### D. Iterative Pose Refinement

In the previous section, describing the conditional SE-block, we assumed there was a valid pose estimate available, as a bad estimate may potentially negatively affect learning. In order to provide the best possible estimate to guide the attention mechanism, iterative pose refinement is proposed. For iterative pose refinement, several pose estimation networks are concatenated together so that the pose estimate from network \( N \) is provided to the CSE-blocks of pose estimation.
network \( N + 1 \). Note that the first pose estimation network in the chain has no pose estimate available, therefore the zero vector is used in this case. The iterative pose refinement process is illustrated in Fig. 1.

In our proposed architecture, it is optional whether weights are shared between all the pose estimation networks or not. With shared weights our network functions similarly to a recurrent network with connections looping back from its output to internal layers, whereas with unshared weights it functions as a cascade of independent networks working together. If weights are shared, our architecture is efficient in terms of the number of parameters to be learned, as increasing the number of iterative refinement steps does not increase the number of parameters to be learned. In either case, the trade-off is that increasing the number of iterations linearly increases the computational cost. However, as we will show in Section IV-A2, only a small number of iterations are needed in practice to achieve good accuracy. During training, the cost function compares the output of each individual iteration of the pose estimation network with the ground truth. This encourages the network to produce the correct pose estimate after each iteration.

### E. In-the-Wild Pose Estimation using Multi-task Learning

It has been noted in the 3D human pose estimation literature that deep neural networks trained using only controlled datasets, such as the H36M or HumanEva datasets, do not generalise well to in-the-wild data [40]. This happens despite the fact that these datasets contain millions of accurately labelled images with a wide range of poses. The failure of networks trained on theses dataset to generalise is likely due to their very limited appearance variability. In terms of background, illumination, clothing and body-type, the variability of images form these datasets is tiny, compared to the huge variability of in the wild data. Hence, even though a network trained on these datasets has been exposed to millions of images with a wide range of poses, it has not been exposed to the full range of appearance variability need to learn a robust feature representation that generalises to real-world data. This problem can be seen as a form of dataset-bias [36], where even though the model does not over-fit in the classical sense, of having a large gap between training and validation error, there is a significant decrease in performance when the network is tested on data outside the training dataset. Similarly, training only on in-the-wild datasets may provide extensive appearance variability but limited pose variability (since sports and surveillance datasets mostly contain walking and running activities), as well as being mostly limited to 2D annotation only. Suitable and large 3D in the wild datasets do not currently exist, to the best of our knowledge.

In order to address the limitations of existing 3D pose datasets, and obtain a network that generalises to real-world data, we use multi-task learning, where the network is trained to perform several related tasks. In our case this means training the network to perform 2D and 3D pose estimation using controlled and in the wild datasets, based on the technique proposed in [40]. This technique can be considered as a form of regularization, where the network must learn an internal representation that generalises across tasks, rather than over-specializing in one task only. Multi-task learning may also be used where there is limited availability of training data for a task of interest, but data exists for several related tasks. By training a single network to perform several related tasks, the network’s performance may be better than separate networks trained on each task independently. This can happen due to knowledge transfer between the related tasks, where features useful for one task also help when performing a related task.

While our goal is to develop a network for real-world 3D human pose estimation, we do not have access to a real-world 3D human pose dataset. Instead, we aim to exploit the available 3D pose datasets with limited appearance variability in combination with diverse 2D pose datasets with limited pose variability. This is feasible since the tasks of 3D and 2D human pose estimation are closely related. They both require estimating the positions of the same human body parts, with the 3D pose task simply requiring an additional depth value to be predicted for each body part. By applying multi-task learning with such tasks, a single network is able to perform...
both 3D and 2D pose estimation, thus allowing us to take advantage of existing in-the-wild 2D pose datasets, which have extensive appearance variability, and the large 3D pose datasets, which have extensive pose variability. Our proposed system is capable of producing accurate 3D pose estimates for real world data, even though all real world images in the 2D dataset do not contain any 3D information.

An overview of our proposed multi-task learning approach is shown in Fig. 4. From our architecture in Fig. 2, all CSE blocks and convolutional layers are shared between the 2D and 3D pose estimation tasks. Sharing the early layers between both tasks encourages the network to learn an internal feature representation that encodes task and dataset agnostic pose information. The output of the shared part of the network is a vector. This vector can be thought of as containing a task-agnostic and dataset agnostic pose representation. The final 2D and 3D pose predictions are produced by using task-specific fully-connected layers to transform the generic pose representation from the final shared layer into the required 2D or 3D pose estimate.

The final task-specific fully-connected layers output the estimated 3D and 2D pose for each image, where 3D pose is represented in normalised world-coordinates and 2D pose is represented in normalised image coordinates. In normalised world-coordinates the 3D pose coordinates are zero-centred at the centre of the hips and scaled to lie within a unit cube. In normalised image coordinates, the 2D coordinates are scaled to lie within a unit square and zero-centred at the centre of the hips. During training, the cost function used depends on the dataset, and hence which modality has ground-truth available, as shown in Fig. 5. When a 2D dataset is used, and hence only 2D ground-truth is available, the error is only back-propagated through the 2D branch, and the network’s prediction for the other branch not penalised.

![Fig. 4: Overview of our multi-task approach to 3D pose prediction in realistic environments. The early network layers are shared between both tasks, encouraging these layer to learn a domain agnostic pose representation, while later layers are free to learn specialised functions for the 2D and 3D prediction tasks.](image)

![Fig. 5: How back-propagation is performed when, 5a, both 2D and 3D ground truth available and, 5b, when only 2D ground truth is available. When only one type of ground truth is available, back-propagation is only performed on the corresponding branch and the network’s prediction for the other branch not penalised.](image)

**F. Cost Function**

The $\ell_1$ distance i.e. absolute difference, is used to compare the network’s output with the ground truth for both the 2D and 3D predictions. The $\ell_1$ distance can be defined as:

$$\ell_1(a, b) = \sum_k | a_k - b_k |$$

where $a$ and $b$ are vectors containing, for example, the ground-truth and estimated pose prediction, and where $a_k$ and $b_k$ address individual elements of each respective vector.

We can define the complete cost function combining multi-task learning, with iterative feedback, used to compare the network’s 2D and 3D predictions with the ground truth for a particular training example as follows:

$$C(a, b) = \sum_i \ell_1(p_{2d,i}^g, g_{2d})I_{2D} + \ell_1(p_{3d,i}^g, g_{3d})I_{3D}$$

where $p_{2d,i}^g$ and $p_{3d,i}^g$ are the network’s prediction for the 2D and 3D pose from iteration $i$ respectively, where $g_{2d}$ and $g_{3d}$ are the ground-truth 2D and 3D pose respectively, and where $I_{2D}$ and $I_{3D}$ are indicator variables that take the value 1 if ground-truth 2D or 3D data is available for a particular training example and 0 otherwise.
IV. Experiments

In this section, we evaluate our pose estimation method on two of the main datasets for human 3D pose estimation, H36M [13] and HumanEva-I [29], as well as in two 2D datasets, the MPII Dataset [1] and the Oxford Town Centre Dataset [3]. The network was trained for 32 epochs using stochastic-gradient-descent (SGD) with a learning rate of 1e-2 and batch-size of 16. Training was carried out on a single Nvidia GTX-980. When using a network based on the ResNet-18 architecture [10] training took around three days to complete with no feedback. Use of iterative feedback linearly increased the training time with each iteration i.e. training with one iteration of feedback increased training time to 6 days. Our network was implemented using pytorch [21]. For all experiments, unless otherwise stated, the network parameters were shared between all iterations. We study the effect of using non-shared parameters in Section IV-A2.

Data Preparation The input image to the network is a square crop, centred at the person’s hips, showing the complete body. Before being input to the network, images were re-size to 128 × 128 pixels before input and the pixel intensity values were re-scaled to lie between 0 and 1. RGB colour images were used. If necessary, images were padded with zeros to maintain a square aspect-ratio, should the person approach the edge of the original frame. The 3D joint locations provided with the datasets were transformed into camera-coordinates and normalized to place the hips at the origin.

Data Augmentation Data Augmentation is used to artificially increase the size and diversity of the training-set. This technique exposes the network to many transformed versions of each training-image. The transformations attempt to simulate the greater diversity of images that may be encountered in realistic data compared to the limited diversity of data in the training-set. While training our system, online data augmentation was performed. Each individual training image was transformed using a random selection of the following augmentation methods: colour, channel augmentation, rotation, translation, mirroring, and scaling. In addition the parameters of each augmentation method were randomly selected from the following distributions: Rotation: ±15°, Cropping (Translation): ±16 pxls, Color: channels randomly permuted, Mirroring: 50% probability. This meant that the network is unlikely to see the same image repeated during training. Individual augmentation methods were applied with 50% probability, thus further increasing variability as the set of augmentations applied was varied for each image. Note that for each augmentation method, the ground-truth 3D and 2D pose data was also transformed to correspond with the transformation of the image.

Datasets The Human 3.6 Million (H36M) dataset [13] includes 3.6 Million images of 7 persons performing 15 different activities. The dataset was recorded from four camera viewpoints at 50 Hz. When comparing with the literature we use protocol 1 [19], the most challenging protocol, where all frames from subjects 1, 5, 6, 7, 8 are used for training and all frames from subjects 9 and 11 are used for testing. Individual images from all camera viewpoints were used during both training and testing. In order to make training our network more computationally tractable on this dataset, we temporally sub-sampled the training-data by two i.e. we used every other frame for training. This halves the training time required for a given number of epochs, and should have only a negligible effect on the final result as temporally adjacent frames are highly visually correlated, so may not contribute much diversity to the training-set.

The HumanEva-I dataset [29] consists of three persons performing five activities: walking, jogging, gestures, boxing and throw/catch. The dataset was captured using three cameras at 60 Hz. As per the literature, we use the provided training/validation split. Results are reported for the walking, jogging and boxing activities, as per the standard evaluation protocol used in [19].

The MPII Human Pose Dataset [1] consists of approx. 25,000 images containing approx. 40,000 persons with 2D pose ground-truth labels. The images have been extracted from online videos of people performing many diverse activities. We extract all persons from the dataset using the ground truth labels to centre the extracted bounding box on the centre of the person’s hips. We use 95% of the images for training and 5% for validation. The Oxford Town Centre dataset [3] contains 1080p surveillance video of pedestrians walking in a busy street in Oxford city centre. There are an average of 16 persons in the frame at any time and the video lasts several minutes. Ground-truth and HoG [25] pedestrian bounding boxes are provided with the dataset.

In the evaluation, 3D pose error was measured using average Euclidean distance, in mm, between the predicted pose and the ground truth. The 2D pose error was measured using average Probability of a Correct Pose (PCP) “A body part returned by the algorithm is considered correct if its segment endpoints lie within 50% of the length of the ground-truth segment from their annotated location.” [9].

A. Ablation Studies of the Attention Mechanism

In this section we perform three ablation experiments on the H36M dataset to validate and justify the individual components of our proposed attention mechanism.

1) Attention: As our first ablation study we disabled the SE-connections and trained the network to perform pose estimation on the H36M dataset. The network with SE-connections disabled is equivalent to a standard Resnet-34. This network was compared with our CSE-Resnet-34 with the SE-connection enabled. Both networks were trained with zero feedback iterations. Results are shown in Table I.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>60.67</td>
<td>70.97</td>
<td>66.70</td>
<td>89.75</td>
<td>74.11</td>
<td>62.56</td>
<td>62.72</td>
<td>121.49</td>
</tr>
<tr>
<td>Enabled</td>
<td>61.60</td>
<td>70.28</td>
<td>64.94</td>
<td>89.48</td>
<td>72.88</td>
<td>61.13</td>
<td>62.62</td>
<td>119.77</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SE-Conn.</th>
<th>Sit</th>
<th>Smk</th>
<th>Phot</th>
<th>Wait</th>
<th>WkL.D</th>
<th>WkL.T</th>
<th>WkL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>83.60</td>
<td>71.38</td>
<td>92.23</td>
<td>83.73</td>
<td>70.43</td>
<td>62.00</td>
<td>49.83</td>
</tr>
<tr>
<td>Enabled</td>
<td>81.88</td>
<td>70.71</td>
<td>88.54</td>
<td>82.86</td>
<td>70.16</td>
<td>63.68</td>
<td>48.08</td>
</tr>
</tbody>
</table>

TABLE I: Pose estimation error (mm) on the H36M dataset comparing the network’s performance when the SE-connections are enabled or disabled.
TABLE II: Pose estimation error (mm) on the H36M dataset with different numbers of feedback iterations, and feedback disabled (Dis.). In this experiment parameters were shared between all iterations.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>60.67</td>
<td>70.37</td>
<td>66.70</td>
<td>89.75</td>
<td>74.11</td>
<td>62.56</td>
<td>62.72</td>
<td>121.49</td>
</tr>
<tr>
<td>1</td>
<td>61.76</td>
<td>70.28</td>
<td>64.94</td>
<td>89.48</td>
<td>72.88</td>
<td>61.13</td>
<td>62.62</td>
<td>119.77</td>
</tr>
<tr>
<td>2</td>
<td>59.59</td>
<td>66.48</td>
<td>64.15</td>
<td>88.56</td>
<td>69.76</td>
<td>59.39</td>
<td>60.83</td>
<td>113.59</td>
</tr>
</tbody>
</table>

The results in Table I show that using SE-connections gives a small improvement in pose estimation error, compared to when these connections are disabled. This shows that re-weighting the convolutional filters, i.e. using the attention mechanism, can give an improvement in performance compared to using a standard feed-forward network with no attention. We will now study the effect of introducing additional feedback information into the attention mechanism.

2) Iterative Feedback: Increasing the number of feedback iterations increases the number of times the network can observe the input image with updated pose information for conditional attention. In this experiment we vary the number of feedback iterations using during both training and testing. We also test the case where the SE-connections are removed altogether and no feedback iterations are used (SE-Dis). When no feedback is used the image is simply passed through the network once, in a feed-forward fashion, and the predicted pose recorded. When using feedback, the image is first passed through the network to obtain a pose estimate. The image is then passed through the network again, along with the pose estimate from the previous iteration for the attention mechanism. This process is repeated for each subsequent iteration, each time using the pose estimate from the previous iteration. This results in a linear increase in computational cost for both training and testing with a given number of feedback cycles. However, as we will show, only a small number of iterations are needed in practice to achieve good performance. This experiment was carried out using the CSE-ResNet-34 architecture. Results are shown in Table II.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>62.66</td>
<td>70.74</td>
<td>67.57</td>
<td>89.45</td>
<td>72.8</td>
<td>64.36</td>
<td>62.35</td>
<td>119.63</td>
</tr>
<tr>
<td>1</td>
<td>57.81</td>
<td>67.89</td>
<td>64.94</td>
<td>85.69</td>
<td>70.29</td>
<td>59.45</td>
<td>60.92</td>
<td>112.6</td>
</tr>
<tr>
<td>2</td>
<td>58.77</td>
<td>65.65</td>
<td>62.83</td>
<td>83.19</td>
<td>69.82</td>
<td>59.6</td>
<td>57.79</td>
<td>113.43</td>
</tr>
</tbody>
</table>

TABLE III: Pose estimation error (mm) on H36M with different numbers of feedback iterations and non-shared parameters i.e. independent parameters at each iteration

used at each feedback iteration are learned independently. This linearly increases the number of parameters as a function of the number of feedback iterations. However, the computational cost is still the same as an equivalent network with parameter sharing and an equal number of feedback iterations. This experiment was carried out using the CSE-ResNet-34 architecture. Therefore, the number of parameters to be learned when using 2 feedback cycles parameters was approx. 60 M. Results are shown in Table III.

When comparing the results in Table II with those in Table III, we can see that using trainable parameters at each feedback iteration can give an improvement in performance, compared to the network with shared parameters. This could be down to each layer better specializing in correcting the errors from the previous iteration. This improvement in performance comes at the cost of increasing the number of parameters by three times. In practice, the network with shared parameters gives similar performance while being far more efficient.

B. Network Architecture

The core idea of a ResNet-style architecture can be used to construct a family of networks with different numbers of residual blocks, and hence different numbers of trainable parameters. Increasing the number of parameters increases the learning capacity of the network, at the cost of increasing the computational cost. We test two CSE-ResNet variants, CSE-ResNet-18 and CSE-ResNet-34, with approximately 11 M and 21 M parameters respectively. Even larger networks were not tested due to the increasing computational cost, which makes these experiments infeasible with our current resources. Results are shown in Table IV for the H36M dataset and Table V for the HumanEva-I dataset. The network architectures and hyperparameters used were the same in both datasets to avoid over-fitting these parameters to a particular dataset.

Comparing the results produced by the CSE-ResNet-18 and CSE-ResNet-34 architectures in both datasets shows that using a larger network gives improved pose estimation error. It can also be seen that increasing the number of feedback iterations improves performance for both networks, although the increase in performance diminishes with each additional iteration showing the asymptotic effect of feedback.

In the HumanEva dataset the pose estimation results are much better than those obtained on the H36M dataset, likely
TABLE IV: Pose estimation error (mm) Human 3.6M dataset comparing the performance of the CSE-Resnet-18 (CSER18) and CSE-Resnet-34 (CSER34) architectures for increasing feedback iterations.

due to the significantly lower variability of subjects and activities. Moving to the larger CSE-ResNet-34 architecture on this dataset has only a small effect on accuracy, as perhaps this dataset is too small to take full advantage of the increased number of parameters, and hence greater learning capacity, of the larger network.

The wall-clock time to complete one epoch of training with no feedback iterations on the CSE-ResNet-34 architecture was around 2 hours, with CSE-ResNet-18 taking around half this time. Using feedback linearly increases these numbers in proportion to the number of iterations. Therefore we decided not to test larger architectures are these were computationally infeasible given our current resources. Pose inference is relatively fast with the CSE-ResNet-34 architecture able to process around 40 images per second with two feedback iterations, and the CSE-ResNet-18 architecture able to process around double this.

C. Multi-Task Training

In this experiment, we examine how the network’s pose estimation error is affected by the use of various tasks during training. Our 3D pose estimation network is trained to perform both 3D and 2D pose estimation using data from the H36M and MPII datasets. We would like to know if the additional constraints introduced by training the network to perform both 3D and 2D pose estimation helps to improve its pose estimation performance. We would also like to know if using additional data from a more diverse dataset, even if this is related to the other task, helps to create a more robust internal pose estimation representation and, as a consequence, improve the overall pose estimation performance. Thus, we aim to improve the 3D pose estimation performance on the controlled H36M dataset using additional data from the appearance-diverse and realistic MPII dataset. To answer these questions, we trained separate networks using different combinations of tasks. For each combination, we recorded the average 3D pose estimation error on the H36M validation-set, and we also record the 2D pose estimation error on the H36M and MPII validation-sets. The results are shown in Table VI for the effect on estimating 3D pose, and Table VII, for the effect on estimating 2D pose. The last line of tables VI and VII, where only the 3D task is used, is equivalent to the previous ablation experiments where no multi-task training is used.

Comparing the results in Table VI it is clear that the best 3D pose estimation performance occurs when the network is trained simultaneously on all tasks. When we remove the task of predicting 2D pose on the H36M dataset, performance drops significantly. This result shows the improvement of multi-task learning for pose estimation, and how introducing additional constraints, by forcing the network to predict 3D and 2D pose for all images the H36M dataset, helps to improve 3D pose estimation performance and shows the utility of having a full correspondence between images, 2D and 3D poses. Alternatively, when the task of predicting 2D pose on the MPII dataset is removed performance also drops but by a lesser amount. This task does not introduce constraints, as these images have no 3D pose information. Instead, we expect that it helps improve the robustness of low-level feature extraction, which is less important on the controlled H36M dataset. Finally, when the network is only trained on the 3D pose estimation task its performance is worse than when it was trained on all the other combinations of tasks. This shows that using both diverse data and adding additional constraints are individually important, and have a large effect when used together.

A similar pattern can be seen in Table VII for 2D pose prediction accuracy, measured using Probability of a Correct Pose (PCP) [9]. The PCP score considers a body part correctly annotated if its end-points are within a distance of 50% of the part-length from their ground-truth locations. Across both datasets, the best overall performance occurs when all the tasks are used during training. When the H36M 2D pose prediction task is excluded, performance at 2D pose prediction on the H36M dataset is similar to that of the MPII dataset. Since the performance on H36M is also the best reported one, this seems to suggest that the network has over-fitted to the H36M activities, maybe due to the unbalanced training-dataset with a larger number of images from H36M. When the MPII 2D pose task is excluded, the performance in H36M almost matches the top result, but it is unable to tackle the complex appearance variations of the realistic MPII dataset, giving the poorest PCP value in-the-wild of all combinations. This suggests that use of multi-task learning using in-the-wild 2D dataset discourages the network from over-specialising on the H36M dataset. As expected, if all 2D prediction tasks are excluded from training, performance at 2D pose estimation is uniformly poor and the system fails to generalise well to real-world data. These results justify the need to use diverse tasks, and hence diverse datasets, to train a network that performs well on in-the-wild data.

D. Data Augmentation

For this experiment, the network was trained while de-activating one of the augmentation techniques. Observing how much performance drops when a single augmentation technique is deactivated gives an indication of the relative
TABLE V: Pose Estimation error (mm) HumanEva-I dataset, comparing the performance of the CSE-Resnet-18 and CSE-Resnet-34 architectures

<table>
<thead>
<tr>
<th>Net</th>
<th>Walking (Cam1)</th>
<th>Jogging (Cam1)</th>
<th>Boxing (Cam1)</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s1</td>
<td>s2</td>
<td>s3</td>
<td>Avg.</td>
</tr>
<tr>
<td>CSER18</td>
<td>Iter</td>
<td>28.06</td>
<td>24.97</td>
<td>34.18</td>
</tr>
<tr>
<td>1</td>
<td>23.54</td>
<td>22.03</td>
<td>31.13</td>
<td>35.57</td>
</tr>
<tr>
<td>2</td>
<td>22.10</td>
<td>19.80</td>
<td>28.86</td>
<td>25.39</td>
</tr>
<tr>
<td>CSER34</td>
<td>0</td>
<td>26.44</td>
<td>24.16</td>
<td>36.95</td>
</tr>
<tr>
<td>1</td>
<td>22.17</td>
<td>19.88</td>
<td>31.63</td>
<td>24.36</td>
</tr>
<tr>
<td>2</td>
<td>21.47</td>
<td>19.75</td>
<td>30.94</td>
<td>24.05</td>
</tr>
</tbody>
</table>
Fig. 6: **Top Two Rows:** Example output of our system on the MPII dataset when trained with multi-task loss. These examples show that the system is capable of accurate 3D and 2D pose estimation in real-world conditions. For each image the figure shows the output from the 2D branch superimposed and the output of the 3D branch to the right of the corresponding figure. The final three images show incorrect outputs caused by factors such as self-occlusion by long clothing, difficult lighting, body position, and occlusion. **Bottom Two Rows:** Example output when training on only the H36M dataset i.e. multi-task loss disabled. Comparison with the top two rows shows the importance of multi-task training for real-world generalisation.

However our framework is not intended to minimise this error; we only use the 2D pose predictions to help with the 3D pose estimation task.

2) **Oxford Dataset:** To further demonstrate the effectiveness of our system in-the-wild, we analyse its performance on the Oxford Town Centre dataset. This dataset was not seen during training of the pose estimation system, hence it acts as an unseen test-set validating our system’s ability to perform under in-the-wild conditions. Two experiments were carried out: one using the ground-truth pedestrian bounding boxes, and the other using the HoG detections provided with the dataset. To predict the pose for an individual person, we extract the region of the image centred on each person’s bounding box, predict their pose using our network, then re-project the network’s pose prediction onto the full image.

We can see from Fig. 7 that the system is able to accurately estimate pose in either scenario. While its performance is better given ground truth bounding boxes, it is generally robust to noisy bounding box locations, such as the HoG detections. We note there are some errors in the pose predictions, even assuming ground truth detections, such as when a person is wearing clothing that obscures their limbs. In these cases, the system can have difficulty, especially when the person is also facing away from the camera. The difficulty of estimating pose in these circumstances is probably due to a lack of similar images in the training-set. In general, the system performs well on the Town Centre dataset with most people showing a natural gait, despite the fact that pose is estimated independently at every frame.

**F. Systematic Errors**

As seen in Section IV-E2, when a person detector is used to extract person images, the bounding box may not be centred on the person’s hips due to inaccurate pedestrian detection. This can have an effect of pose estimation accuracy as our system assumes the hips are centred in the image. Therefore, we investigate the sensitivity of our pose estimation system to detector inaccuracy by comparing two variants of our network: the network trained using standard hyper-parameters and training procedure, and the network where cropping data-augmentation was disabled during training. Both variants were trained on the H36M and MPII datasets.

To compare the robustness of the networks, we perform pose estimation on modified images from the H36M validation-set, where the person’s hips have been systematically offset from the image centre i.e., the person’s hips are no longer in the image centre. For each displacement distance, we record the average pose estimation error across all modified validation-set images. The results can be seen in Fig. 8, where displacement is recorded in terms of fractions of average body width.

These results show that the standard network, which was exposed to images with small displacements during training, performs well for small displacements - giving only a small loss of accuracy. For larger displacements, its performance falls off gracefully and only becomes completely unreliable once the centre of the image is focused on the background, rather than inside the body-region of the person of interest. This contrasts with the network trained without cropping augmentation where performance quickly degrades for even small displacements of the person from the image centre. This experiment demonstrates that our pose estimation method can cope with detector inaccuracy, and this conclusion is further reinforced up by the results in Section IV-E2 using a realistic detector on realistic data.
Fig. 7: Results from running our pose estimation system on frames from the Oxford Dataset using two different methods for extracting person bounding boxes. **Left:** Using HoG Person Detections **Right** Ground Truth Person Detections.

Fig. 8: Comparing the robustness of networks trained with and without cropping data augmentation. Results show the average pose error (mm) on the H36M validation-set as the person’s hips are displaced from the centre of the image, simulating person detector inaccuracy. Displacement from the centre is expressed as a fraction of average body width.

**G. Real-Time Pose Estimation**

In addition to the above, we show that our model is efficient enough to run in real-time on commodity hardware. We take the trained network and run it on a standard consumer laptop using either CPU only, or the CPU and GPU\(^1\). The laptop webcam was used to gather real-time images for input to the network. Person bounding boxes were extracted from each frame using the default OpenCV pedestrian detector [5]. All the bounding boxes for each frame were passed as a batch to the network in order to estimate the pose for each person. Using only the CPU, the complete system was able to perform multi-person pose estimation at around 10 Hz. When running the pose estimation component on an Nvidia GTX980 GPU, pose estimation was performed at 157 Hz for one person, 56 Hz for eight persons, and 3 Hz for 128 persons. The OpenCV person detector component runs at around 15 Hz by itself. Therefore when using GPU acceleration for pose estimation, and with a reasonable number of persons on screen, the limiting factor is the pedestrian detector, rather than the pose estimation network. Use of the detector was necessary due to the fact that our network was trained to assume the person is centred in the image frame, therefore individual persons must be cropped from the larger frame before their image is passed as input to the network. Better detectors using CNNs could be used in future versions to overcome this limitation. Example output, showing a screen-shot from the system being run in real-time can be seen in Fig. 9.

Fig. 9: Screen-capture of our system running on a consumer laptop in an uncontrolled environment. The system is performing multi-person pose estimation in real-time using the CPU.

**H. Comparison with State of the Art**

We now compare our pose estimation on the H36M and HumanEva-I datasets with the literature. The results for the H36M dataset are shown in Table X and for the HumanEva-I dataset in Table IX. Qualitative results are shown in Figure 10. Note that we use the same hyper-parameters for training on both datasets, showing that these parameters are not over-fitted to each dataset. As far as we know we are the first paper to systematically report results on both these datasets. Examining the results it can be seen that our system achieves competitive performance in both these datasets. Unlike several other methods that make use of additional training data from visually diverse 2D pose datasets [40] [17] [31], we achieve comparable performance using only training data from the original 3D pose datasets. Furthermore, our method is conceptually simpler than many others e.g. [32], while obtaining only

\(^1\)Code available at https://github.com/niallmcl/Human-Pose-Estimation
marginally lower accuracy. In addition, our proposed approach has the potential to be combined with some of the better performing approaches, such as using additional training 2D training data [40], or altering the pose representation [31], to further improve its performance.

I. Conclusion

In this paper, we have proposed a novel system for 3D human pose estimation. We introduce a novel conditional attention mechanism that is combined with iterative feedback for iterative pose estimation. This novel conditional attention mechanism reweights the convolutional filters to focus on image evidence relevant to predicting the current pose. We also introduce an effective 2D/3D multi-task training paradigm that allows for improving the individual performance of each task while making use of the incomplete but complementary information available in diverse datasets. We show that our conceptually simple network can achieve competitive results on the H36M and HumanEva-I datasets. More importantly, we prove that our architecture can provide robust pose estimates in realistic scenarios including commodity sensors such as webcams or CCTV cameras with multiple persons present.

REFERENCES


Table IX: Pose estimation error (mm) on the HumanEva-I dataset compared with the literature. Ours is with CSE-Resnet-34, iterations=2, Ours+MultiTask uses additional 2D training data from the MPII dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Directions</th>
<th>Discussion</th>
<th>Eating</th>
<th>Greet</th>
<th>Phone</th>
<th>Posing</th>
<th>Purchases</th>
<th>Sitting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sanzari16 [27]</td>
<td>48.82</td>
<td>56.31</td>
<td>95.98</td>
<td>84.78</td>
<td>96.47</td>
<td>66.3</td>
<td>107.41</td>
<td>116.89</td>
</tr>
<tr>
<td>Moreno17 [19]</td>
<td>69.54</td>
<td>80.15</td>
<td>78.2</td>
<td>87.01</td>
<td>100.75</td>
<td>76.01</td>
<td>69.65</td>
<td>101.71</td>
</tr>
<tr>
<td>Xue19 [39]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sarandi18 [28]</td>
<td>51.3</td>
<td>57.8</td>
<td>52.5</td>
<td>53.8</td>
<td>55.9</td>
<td>50.9</td>
<td>52.8</td>
<td>66.7</td>
</tr>
<tr>
<td>Kolotouros19 [15]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Zhou17 [40]</td>
<td>84.22</td>
<td>87.88</td>
<td>79.71</td>
<td>82.6</td>
<td>87.5</td>
<td>58.4</td>
<td>70.62</td>
<td>70.04</td>
</tr>
<tr>
<td>Zhaoi17 [17]</td>
<td>51.8</td>
<td>56.2</td>
<td>58.1</td>
<td>59</td>
<td>69.5</td>
<td>55.2</td>
<td>56.1</td>
<td>74</td>
</tr>
<tr>
<td>Sun17 [31]</td>
<td>42.1</td>
<td>44.3</td>
<td>45</td>
<td>45.4</td>
<td>51.5</td>
<td>43.2</td>
<td>41.3</td>
<td>59.3</td>
</tr>
<tr>
<td>Pavlakos17 [22]</td>
<td>67.38</td>
<td>71.95</td>
<td>66.7</td>
<td>69.07</td>
<td>71.95</td>
<td>65.03</td>
<td>68.3</td>
<td>83.66</td>
</tr>
<tr>
<td>Tekin17 [32]</td>
<td>54.23</td>
<td>61.41</td>
<td>60.17</td>
<td>61.23</td>
<td>79.41</td>
<td>63.14</td>
<td>81.63</td>
<td>70.14</td>
</tr>
<tr>
<td>Ours</td>
<td>59.59</td>
<td>66.48</td>
<td>64.15</td>
<td>84.56</td>
<td>69.76</td>
<td>59.39</td>
<td>60.83</td>
<td>79.23</td>
</tr>
<tr>
<td>Ours+MT</td>
<td>60.01</td>
<td>65.50</td>
<td>63.68</td>
<td>71.97</td>
<td>67.25</td>
<td>59.65</td>
<td>60.76</td>
<td>75.17*</td>
</tr>
</tbody>
</table>

Table X: Pose estimation error (mm) on the Human 3.6M dataset using Protocol 1 compared with the literature. Methods marked with * use additional training data outside the H36M dataset. Ours is with CSE-Resnet-34, iterations=2, Ours+MultiTask also uses additional 2D training data from the MPII dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>SitDown</th>
<th>Smoking</th>
<th>Photo</th>
<th>Wait</th>
<th>Walk</th>
<th>WalkD.</th>
<th>WalkL.</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sanzari16 [27]</td>
<td>129.63</td>
<td>97.84</td>
<td>105.8</td>
<td>65.94</td>
<td>92.58</td>
<td>130.46</td>
<td>102.21</td>
<td>93.15</td>
</tr>
<tr>
<td>Moreno17 [19]</td>
<td>113.91</td>
<td>89.68</td>
<td>102.71</td>
<td>98.49</td>
<td>79.18</td>
<td>82.4</td>
<td>77.17</td>
<td>87.3</td>
</tr>
<tr>
<td>Xue19 [39]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sarandi18 [28]</td>
<td>57.1</td>
<td>56.6</td>
<td>58.6</td>
<td>51.6</td>
<td>56.6</td>
<td>51.7</td>
<td>54.8</td>
<td>60.3</td>
</tr>
<tr>
<td>Kolotouros19 [15]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Zhou17 [40]</td>
<td>131.75</td>
<td>79.61</td>
<td>77.38</td>
<td>85.1</td>
<td>76.95</td>
<td>67.49</td>
<td>71.99</td>
<td>80.98</td>
</tr>
<tr>
<td>Zhaoi17 [17]</td>
<td>111.59</td>
<td>64.15</td>
<td>65.53</td>
<td>66.05</td>
<td>63.22</td>
<td>51.43</td>
<td>55.33</td>
<td>64.9*</td>
</tr>
<tr>
<td>Sun17 [31]</td>
<td>73.3</td>
<td>51</td>
<td>53</td>
<td>44</td>
<td>38.3</td>
<td>48</td>
<td>44.8</td>
<td>48.3*</td>
</tr>
<tr>
<td>Pavlakos17 [22]</td>
<td>96.51</td>
<td>71.74</td>
<td>76.97</td>
<td>65.83</td>
<td>59.11</td>
<td>74.89</td>
<td>63.24</td>
<td>71.9</td>
</tr>
<tr>
<td>Tekin17 [32]</td>
<td>107.31</td>
<td>69.29</td>
<td>78.31</td>
<td>70.27</td>
<td>51.79</td>
<td>74.28</td>
<td>63.24</td>
<td>69.73</td>
</tr>
<tr>
<td>Xue19 [39]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sarandi18 [28]</td>
<td>77.1</td>
<td>56.6</td>
<td>58.6</td>
<td>51.6</td>
<td>56.6</td>
<td>47.6</td>
<td>42.8</td>
<td>56.1</td>
</tr>
<tr>
<td>Kolotouros19 [15]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Tome18 [35]</td>
<td>95.2</td>
<td>50.2</td>
<td>64.3</td>
<td>52.2</td>
<td>43.9</td>
<td>51.1</td>
<td>45.3</td>
<td>52.8</td>
</tr>
<tr>
<td>Ours</td>
<td>113.59</td>
<td>69.91</td>
<td>85.63</td>
<td>80.99</td>
<td>48.68</td>
<td>70.54</td>
<td>61.61</td>
<td>71.9</td>
</tr>
<tr>
<td>Ours+MultiTask</td>
<td>103.12</td>
<td>67.45</td>
<td>84.79</td>
<td>68.66</td>
<td>49.75</td>
<td>71.29</td>
<td>61.23</td>
<td>68.8*</td>
</tr>
</tbody>
</table>

Fig. 10: Example 3D human pose output from the system on the H36M dataset.
Niall McLaughlin

Niall McLaughlin received his PhD from Queens University Belfast in 2012. Following this, he worked as a Research Fellow at the Centre for Secure Information Technologies (CSIT) for six years, focusing on the application of deep learning to video and image analysis. He was appointed as a lecturer in computer science at the school of EEECS at Queen’s University Belfast in August 2018. His research interests include face recognition, person re-identification from images and video, multi-person tracking and real-time 3D human pose estimation. In addition, he has recently worked on applying deep learning to the problem of malware detection and analysis for cybersecurity applications.

Jesus Martinez-del-Rincon

Jesus Martinez del Rincon is presently a Senior Lecturer in the School of Electronics, Electrical Engineering and Computer Science at the Queens University of Belfast. He received a BSc in Telecommunication Engineering in 2003 and was awarded a PhD in Computer Vision in 2008 from the University of Zaragoza for his work into the development of tracking algorithms for video surveillance and human motion analysis. More recently, his main research interest has been in Deep learning applied to video surveillance and cybersecurity.

Paul Miller

Paul C. Miller was born in Belfast, Northern Ireland, in 1964. He received the B.Sc. (Hons) and Ph.D degrees in pure and applied physics from Queens University Belfast in 1985 and 1989 respectively. From 1989 to 1991, he was a Research Fellow with the School of Electrical and Electronic Engineering at Queens. From 1991 to 1999, he was a Research Scientist, and then Senior Research Scientist, with the Defence, Science and Technology Organization in Adelaide, Australia. In 1999 he returned to Queens as a Lecturer with the School of Electronics, Electrical Engineering and Computer Science. He is currently a Reader and is the author of more than 100 articles. Since returning to academia he has continued to work in video analytics for both defence and civilian CCTV applications, and also bio-medical image analysis. His research interests include image restoration, segmentation, multi-target tracking, person re-identification and gender/age profiling of subjects in video. More recently, his main research interest has been in the convergence of cyber and physical security. Dr Miller was a recipient of the IPRCS International Machine Vision and Image Processing Conference Best Paper Award in 2008.