Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation

Bugra Tekin Pablo Márquez-Neila Mathieu Salzmann Pascal Fua
EPFL, Switzerland
{bugra.tekin,pablo.marquezneila,mathieu.salzmann,pascal.fua}@epfl.ch

Abstract

Most recent approaches to monocular 3D human pose estimation rely on Deep Learning. They typically involve regressing from an image to either 3D joint coordinates directly or 2D joint locations from which 3D coordinates are inferred. Both approaches have their strengths and weaknesses and we therefore propose a novel architecture designed to deliver the best of both worlds by performing both simultaneously and fusing the information along the way. At the heart of our framework is a trainable fusion scheme that learns how to fuse the information optimally instead of being hand-designed. This yields significant improvements upon the state-of-the-art on standard 3D human pose estimation benchmarks.

1. Introduction

Monocular 3D human pose estimation is a longstanding problem of Computer Vision. Over the years, two main classes of approaches have been proposed: Discriminative ones that directly regress 3D pose from image data [1, 8, 34, 46, 56, 67] and generative ones that search the pose space for a plausible body configuration that aligns with the image data [21, 60, 68]. With the advent of ever larger datasets [30], models have evolved towards deep architectures, but the story remains largely unchanged. The state-of-the-art approaches can be roughly grouped into those that directly regress 3D pose from images [30, 38, 64, 65] and those that first predict a 2D pose in the form of joint location confidence maps and fit a 3D model to this 2D prediction [9, 76].

Since detecting the 2D image location of joints in easier than directly inferring the 3D pose, it can be done more reliably. However, inferring a 3D pose from these 2D locations is fraught with ambiguities and the above-mentioned methods usually rely on a database of 3D models to resolve them, at the cost of a potentially expensive run-time fitting procedure. By contrast, the methods that regress directly to 3D avoid this extra step but also do not benefit of the well-posedness of the 2D joint detection location problem.

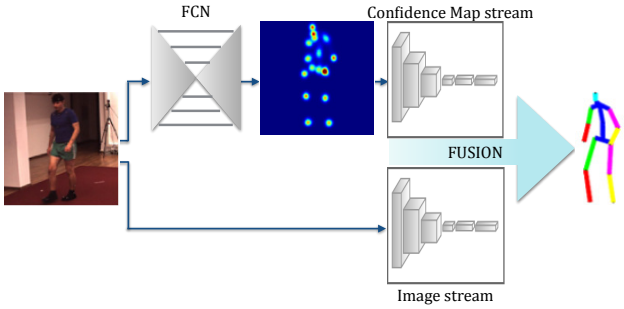


Figure 1: **Overview of our approach.** One stream of our network accounts for the 2D joint locations and the corresponding uncertainties. The second one leverages all 3D image cues by directly acting on the image. The outputs of these two streams are then fused to obtain the final 3D human pose estimate.

In this paper, we propose the novel architecture depicted by Fig. 1 designed to deliver the best of both worlds. The first stream, which we will refer to as the *Confidence Map Stream*, first computes a heatmap of 2D joint locations and then infer the 3D poses from it. The second stream, which we will dub the *Image Stream*, is designed to produce features that complement those computed by the first stream and can be used in conjunction with them to compute the 3D pose, that is, guide the regression process given the 2D locations.

However, for this approach to be beneficial, effective fusion of the two streams is crucial. In theory, it could happen at any stage of the two streams, ranging from early to late fusion, with no principled way to choose one against the other. We therefore also developed a *trainable fusion* scheme that learns how to fuse the two streams.

Ultimately, our approach allows the network to still exploit image cues while inferring 3D poses from 2D joint locations. As we demonstrate in our experiments, the features computed by both streams are decorrelated and therefore truly encode complementary information. Our contributions can be summarized as follows:

- We introduce a discriminative fusion framework to

simultaneously exploit 2D joint location confidence maps and 3D image cues for 3D human pose estimation.

- We introduce a novel trainable fusion scheme, which automatically learns where and how to fuse these two sources of information.

We show that our approach significantly outperforms the state-of-the-art results on standard benchmarks and yields accurate pose estimates from images acquired in unconstrained outdoors environments.

2. Related Work

The existing 3D human pose estimation approaches can be roughly categorized into discriminative and generative ones. In what follows, we review both types of approaches.

Discriminative methods aim at predicting 3D pose directly from the input data, may it be single images [28, 29, 37, 38, 39, 46, 52, 55, 64, 73], depth images [23, 50, 59], or short image sequences [65]. Early approaches falling into this category typically worked by extracting handcrafted features and learning a mapping from these features to 3D poses [1, 8, 28, 29, 37, 56, 67]. Unsurprisingly, the more recent methods tend to rely on Deep Networks [38, 64, 65, 75]. In particular, [38, 65] rely on 2D poses to pretrain the network, thus exploiting the commonalities between 2D and 3D pose estimation. In fact, [38] even proposes to jointly predict 2D and 3D poses. However, in such approaches, the two predictions are not coupled. By contrast, [45] introduces a network that uses 2D information for 3D pose estimation. This method, however, does not exploit pixelwise joint location uncertainty, and only makes use of the 2D evidence late in the pose estimation process. While these methods exploit the available 3D image cues, they fail to explicitly model 2D joint location uncertainty, which matters when addressing a problem as ambiguous as monocular 3D pose estimation.

Since pose estimation is much better-posed in 2D than in 3D, a popular way to infer joint positions is to use a generative model to find a 3D pose whose projection aligns with the 2D image data. In the past, this usually involved inferring a 3D human pose by optimizing an energy function derived from image information, such as silhouettes [6, 14, 21, 22, 25, 31, 44, 49, 60], trajectories [74], feature descriptors [58, 62, 63] and 2D joint locations [2, 3, 5, 20, 36, 51, 57, 68, 69]. Another class of approaches retrieve the pose from a dictionary of 3D poses based on similarity with the 2D image evidence [18, 26, 39, 41, 42]. With the growing availability of large datasets and the advent of Deep Learning, the emphasis has shifted towards using discriminative 2D pose regressors [11, 13, 15, 16, 24, 27, 32, 43, 47, 48, 66, 70, 71] to extract the 2D pose and infer a 3D one from it [9, 19, 72, 76].

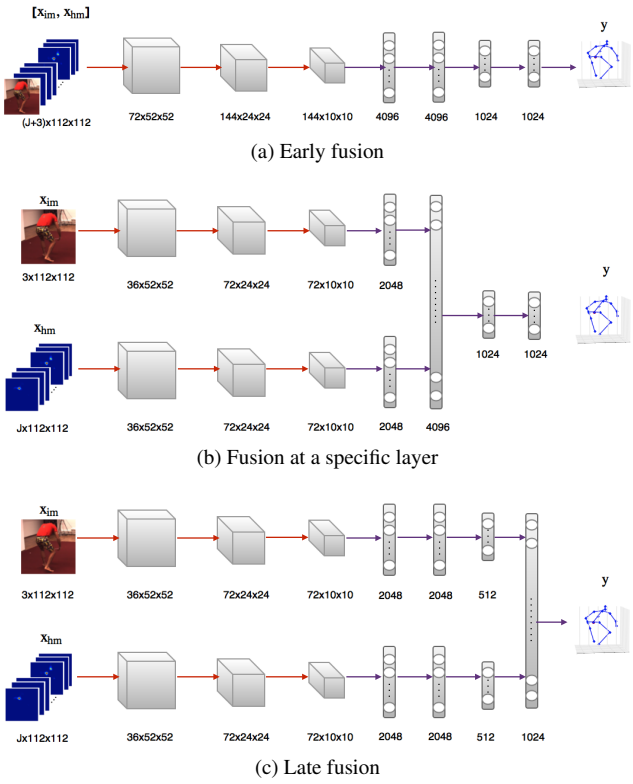


Figure 2: **Three different instances of hard-coded fusion.** The fusion strategies combine 2D joint location confidence maps with 3D cues directly extracted from the input image.

The 2D joint locations are represented by heatmaps that encode the confidence of observing a particular joint at any given image location. A human body representation, such as a skeleton [76], or a more detailed model [9] can then be fitted to these predictions. While this takes 2D joint positions into account, it ignores image information during the fitting process. It therefore discards potentially important 3D cues that could help resolve ambiguities.

3. Approach

Our goal is to increase the robustness and accuracy of monocular 3D pose estimation by exploiting image cues to the full while also taking advantage of the fact that 2D joint locations can be reliably detected by modern CNN architectures. To this end, we designed the two stream architecture depicted by Fig. 1. The Confidence Map Stream shown at the top first computes a heatmap of 2D joint locations from which feature maps can be computed. The Image Stream shown at the bottom extracts additional features directly from the image and all these features are fused to produce a final 3D pose vector.

As shown in Fig. 2, there is a whole range of ways to

perform the fusion of these two data streams, ranging from early to late fusion with no obvious way to choose the best, which might well be problem-dependent anyway. To solve this conundrum, we rely on the fusion architecture depicted by Fig. 3, which involves indroducing a third *fusion stream* that combines the feature maps produced by the two data streams in a trainable way. Each layer of the fusion stream acts on a linear combination of the previous fusion layer with the concatenation of the two data stream outputs. In effect, different weight values for these linear combinations correspond to different fusion strategies.

In the remainder of this section, we formalize this generic architecture and study different ways to set these weights, including learning them along with the weights of the data streams, which is the approach we advocate.

3.1. Fusion Network

Let $\{\mathbf{I}_l\}_{l=0}^{L}$ be the feature maps of the *image stream* and $\{\mathbf{X}_l\}_{l=0}^{L}$ be the feature maps of the *confidence map stream*. As special cases, $\mathbf{I}_0 : [1,3] \times [1,H] \times [1,W] \rightarrow [0,1]$ is the input RGB image, and $\mathbf{X}_0 : [1,J] \times [1,H] \times [1,W] \rightarrow \mathbb{R}_+$ are the confidence maps encoding the probability of observing each one of J body joints at any given image location. The feature maps $\mathbf{I}_l$ and $\mathbf{X}_l$ at each layer l must coincide in width and height but can have different number of channels. In the following, we denote each feature map at level l as both the output of layer l and the input to layer $l+1$.

Let $\{\mathbf{Z}_l\}_{l=0}^{L+1}$ be the feature maps of the *fusion stream*. The feature map $\mathbf{Z}_l$ is the output of layer l, but, unlike in the data streams, the input to layer $l+1$ is a linear combination of $\mathbf{Z}_l$ with $\mathbf{I}_l$ and $\mathbf{X}_l$ given by

$$(1 - w_l) \cdot \text{concat}(\mathbf{I}_l, \mathbf{X}_l) + w_l \cdot \mathbf{Z}_l, \quad 1 \leq l \leq L, \qquad (1)$$

where $\text{concat}(\cdot,\cdot)$ is the concatenation of the given feature maps along the channel axis, and w_l is the l-th element of the fusion weights $\mathbf{w} \in [0,1]^L$ controlling the mixture. For this mixture to be possible, $\mathbf{Z}_l$ must have the same size as $\mathbf{I}_l$ and $\mathbf{X}_l$ and a number of channels equal to the sum of the number of channels of $\mathbf{I}_l$ and $\mathbf{X}_l$. As special cases, $\mathbf{Z}_0 = \text{concat}(\mathbf{I}_0, \mathbf{X}_0)$, and $\mathbf{Z}_{L+1} \in \mathbb{R}^{3J}$ is the output of the network, that is, the J predicted 3D joint locations.

In essence, the fusion weights $\mathbf{w}$ control where and how the fusion of the data streams occurs. Different settings of these weights lead to different fusion strategies. We illustrate this with two special cases below, and then introduce an approach to automatically learn these weights together with the other network parameters.

Early fusion. If the fusion weights are all set to one, $\mathbf{w} = \mathbf{1}$, the two data streams are ignored, and only the fusion one is considered to compute the output. Since the fusion stream takes the concatenation of the image $\mathbf{I}_0$ and the confidence maps $\mathbf{X}_0$ as input, this is equivalent to the early fusion architecture of Fig. 2(a).

Fusion at a specific layer. Instead of fusing the streams in the very first layer, one might want to postpone the fusion point to a later layer $\beta \in \{0, \cdots, L\}$. In our formalism, this can be achieved by setting the fusion weights to $w_l = \mathbb{I}[l > \beta]$, where $\mathbb{I}$ is the indicator function. For example, when $\beta = 4$, our network becomes equivalent to the one depicted by Fig. 2(b). The early and late fusion architectures of Fig. 2(a, c) can also be represented in this manner by setting $\beta = 0$ and $\beta = L$, respectively.

Ultimately, the complete fusion network encodes a function $f(\mathbf{i}, \mathbf{x}; \theta, \mathbf{w}) = \mathbf{Z}_{L+1}|_{\mathbf{I}_0=\mathbf{i}, \mathbf{X}_0=\mathbf{x}}$ mapping from an image $\mathbf{i}$ and confidence maps $\mathbf{x}$ to the 3D joint locations, parametrized by layer weights θ and fusion weights $\mathbf{w}$. With manually-defined fusion weights, given a set of N training pairs $(\mathbf{i}_n, \mathbf{x}_n)$ with corresponding ground-truth joint positions $\mathbf{y}_n$, the parameters θ can be learnt by minimizing the square loss expressed as

$$L(\theta) = \sum_{n=1}^{N} \|f(\mathbf{i}_n, \mathbf{X}_n; \theta, \mathbf{w}) - \mathbf{y}_n\|_2^2 \ . \qquad (2)$$

Trainable fusion. Setting the weights manually, which in our formalism boils down to choosing β, is not obvious; the best value for β will typically depend on the network architecture, the problem and the nature of the input data. A straightforward approach would consist of training networks for all possible values of β to validate the best one, but this quickly becomes impractical. To address this issue, we introduce a trainable fusion approach, which aims to learn β from data jointly with the network parameters. To this end, however, we cannot directly use the indicator function, which has zero derivatives almost everywhere, thus making it inapplicable to gradient-based optimization. Instead, we propose to approximate the indicator function by a sigmoid function

$$w_l = \frac{1}{1 + e^{-\alpha \cdot (l - \beta)}}, \qquad (3)$$

parameterized by α and β. As above, β determines the stage at which fusion occurs and α controls how sharp the transition between weights with value 0 and with value 1 is. When $\alpha \rightarrow \infty$, the function in Eq. 3 becomes equivalent to the indicator function[1], while, when $\alpha = 0$, the network mixes the data and fusion streams in equal proportions at every layer.

In practice, mixing the data and fusion streams at every layer is not desirable. First, by contrast to having binary weights $\mathbf{w}$, which deactivate some of the layers of each stream, it corresponds to a model with a very large number of active parameters, and thus prone to overfitting. Furthermore, after training, a model with binary weights can be

[1] Except at $l = \beta$.

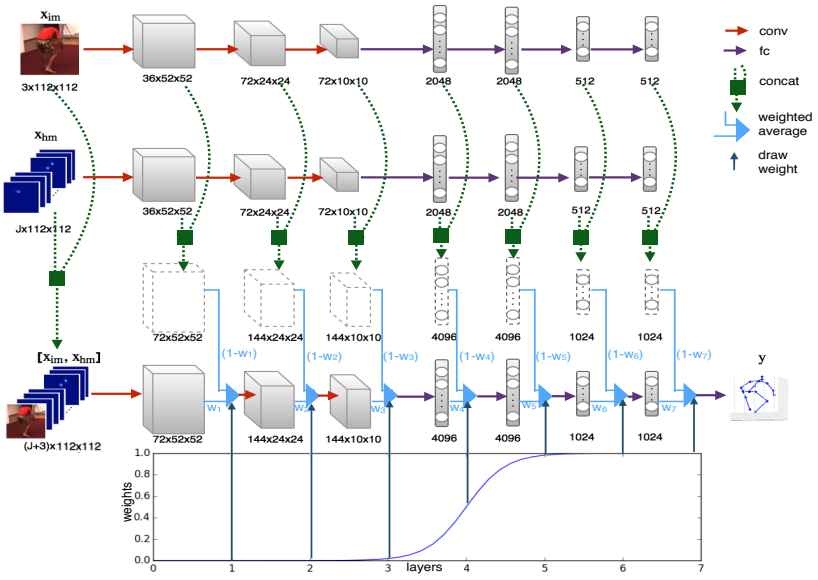


Figure 3: **Trainable fusion architecture.** The first two streams take as input the image and 2D joint location confidence maps, respectively. The combined feature maps of the image and confidence map stream are fed into the fusion stream and linearly combined with the outputs of the previous fusion layer. The linear combination of the streams is controlled by a weight vector shown at the bottom part of the figure. The numbers below each layer represent the corresponding size of the feature maps for convolutional layers and the number of neurons for fully connected ones.

pruned, by removing the inactive layers in each stream, that is all layers l from the fusion stream where $w_l \approx 0$, and all layers l from the data streams where $w_l \approx 1$. This yields a more compact, and thus more efficient network for test-time prediction.

To account for this while learning where to fuse the information sources, we modify the loss function of Eq. (2) by incorporating a term that penalizes small values of α and favors sharp fusions. This yields a loss of the form

$$L(\theta, \alpha, \beta) = \sum_{n=1}^{N} \|f(\mathbf{i}_n, \mathbf{X}_n; \theta, \alpha, \beta) - \mathbf{y}_n\|_2^2 + \frac{\lambda}{\alpha^2} \ , \quad (4)$$

with α and β as trainable parameters, in addition to θ, and a hyperparameter λ weighing the penalty term. Altogether, this loss lets us simultaneously find the most suitable fusion layer β for the given data and the corresponding network parameters θ, while encouraging a sharp fusion function to mimic the behavior of the indicator function.

In practice, we initialize α with a small value of 0.1 and β to the middle layer of the complete network. We use the ADAM [35] gradient update method with a learning rate of 10^{-3} to guide the optimization. We set the regularization parameter to $5 \cdot 10^3$, which renders the magnitude of both the regularization term and the main cost comparable. We use dropout and data augmentation to prevent overfitting.

3.2. 2D Joint Location Confidence Map Prediction

Our approach depends on generating heatmaps of the 2D joint locations that we can feed as input to the confidence map stream. To do so, we rely on a fully-convolutional network with skip connections [43]. Given an RGB image as input, it performs a series of convolutions and pooling operations to reduce its spatial resolution, followed by upconvolutions to produce pixel-wise confidence values for each pixel. We employed the stacked hourglass network design of [43], which carries out repeated bottom-up, top-down processing to capture spatial relationships in the image. We perform heatmap regression to assign high confidence values to the most likely joint positions. In our experiments, we fine-tuned the hourglass network initially trained on the MPII dataset [4] using the training data specific to each experiment as a preliminary step to training our fusion network. In practice, we have observed that using the more accurate 2D joint locations predicted by the stacked network architecture improves the overall 3D prediction accuracy over using those predicted by a single-stage fully-convolutional network, such as [54]. Ultimately, these predictions provide reliable intermediate features for the 3D pose estimation task.

4. Results

In this section, we first describe the datasets we tested our approach on and the corresponding evaluation proto-

cols. We then compare our approach against the state-of-the-art methods and provide a detailed analysis of our general framework.

4.1. Datasets

We evaluate our approach on the Human3.6m [30], HumanEva-I [61], KTH Multiview Football II [10] and Leeds Sports Pose (LSP) [33] datasets described below.

Human3.6m is a large and diverse motion capture dataset including 3.6 million images with their corresponding 2D and 3D poses. The poses are viewed from 4 different camera angles. The subjects carry out complex motions corresponding to daily human activities. We use the standard 17 joint skeleton from Human3.6m as our pose representation.

HumanEva-I comprises synchronized images and motion capture data and is a standard benchmark for 3D human pose estimation. The output pose is a vector of 15 3D joint coordinates.

KTH Multiview Football II provides a benchmark to evaluate the performance of pose estimation algorithms in unconstrained outdoor settings. The camera follows a soccer player moving around the pitch. The videos are captured from 3 different camera viewpoints. The output pose is a vector of 14 3D joint coordinates.

LSP is a standard benchmark for 2D human pose estimation and does not contain any ground-truth 3D pose data. The images are captured in unconstrained outdoor settings. 2D pose is represented in terms of a vector of 14 joint coordinates. We report qualitative 3D pose estimation results on this dataset.

4.2. Evaluation Protocol

On Human3.6m, we used the same data partition as in earlier work [38, 39, 40, 65, 76] for a fair comparison. The data from 5 subjects (S1, S5, S6, S7, S8) was used for training and the data from 2 different subjects (S9, S11) was used for testing. We evaluate the accuracy of 3D human pose estimation in terms of average Euclidean distance between the predicted and ground-truth 3D joint positions, as in [38, 39, 40, 65, 76]. Training and testing were carried out monocularly in all camera views.

In [9], [46][2] , and [58][3] the estimated skeleton was first aligned to the ground-truth one by Procrustes transformation before measuring the joint distances. This is therefore what we also do when comparing against [9, 46, 58].

[2]While [46] also reports results without Procrustes analysis, the authors confirmed to us by email that their evaluation assumes the ground-truth depth of the root joint to be known to go from their volumetric representation to 3D pose in metric space. Since this also sets the scale of the skeleton, we believe that a comparison using the full Procrustes transformation for both their approach and ours is the right one to perform here.

[3]This it is not explicitly stated in [58], but the authors confirmed this to us by email.

On HumanEva-I, following the standard evaluation protocol [9, 62, 65, 72, 76], we trained our model on the training sequences of subjects S1, S2 and S3 and evaluated on the validation sequences of all subjects. We pretrained our network on Human3.6m and used only the first camera view for further training and validation.

On the KTH Multiview Football II dataset, we evaluate our method on the sequence containing Player 2, as in [7, 10, 46, 65]. Following [7, 10, 46, 65], the first half of the sequence from camera 1 is used for training and the second half for testing. To compare our results to those of [7, 10, 46, 65], we report accuracy using the percentage of correctly estimated parts (PCP) score. Since the training set is quite small, we propose to pretrain our network on the recent synthetic dataset [12], which contains images of sports players with their corresponding 3D poses. We then fine-tuned it using the training data from KTH Multiview Football II. We report results with and without this pretraining.

4.3. Comparison to the State-of-the-Art

We first compare our approach with state-of-the-art baselines on the *Human3.6m [30], HumanEva [61] and KTH Multiview Football [10]* datasets.

Human3.6m. In Table 1, we compare the results of our trainable fusion approach with those of the following state-of-the-art single image-based methods: KDE regression from HOG features to 3D poses [30], jointly training a 2D body part detector and a 3D pose regressor [38, 45], the maximum-margin structured learning framework of [39, 40], the deep structured prediction approach of [64], pose regression with kinematic constraints [75], and 3D pose estimation with mocap guided data augmentation [53]. For completeness, we also compare our approach to the following methods that rely on either multiple consecutive images or impose temporal consistency: regression from short image sequences to 3D poses [65], fitting a sparse 3D pose model to 2D confidence map predictions across frames [76], and fitting a 3D pose sequence to the 2D joints predicted by images and height-maps that encode the height of each pixel in the image with respect to a reference plane [17].

As can be seen from the results in Table 1, our approach outperforms all the methods on all the action categories by a large margin. In particular, we outperform the image-based regression methods of [30, 38, 39, 40, 64, 45, 75], as well as the model-fitting strategy of [39, 40]. This, we believe, clearly evidences the benefits of fusing 2D joint location confidence maps with 3D image cues, as done by our approach. Furthermore, we also achieve lower error than the method of [53], despite the fact that it relies on additional training data. Even though our algorithm uses only individual images, it also outperforms the methods that rely on sequences [17, 65, 76].

Input	Method	Directions	Discussion	Eating	Greeting	Phone Talk	Posing	Buying	Sitting	Sitting Down
Single-Image	Ionescu et al. [30]	132.71	183.55	132.37	164.39	162.12	150.61	171.31	151.57	243.03
	Li & Chan [38]	-	148.79	104.01	127.17	-	-	-	-	-
	Li et al. [39]	-	134.13	97.37	122.33	-	-	-	-	-
	Li et al. [40]	-	133.51	97.60	120.41	-	-	-	-	-
	Zhou et al. [76]	-	-	-	-	-	-	-	-	-
	Rogez & Schmid [53]	-	-	-	-	-	-	-	-	-
	Tekin et al. [64]	-	129.06	91.43	121.68	-	-	-	-	-
	Park et al. [45]	100.34	116.19	89.96	116.49	115.34	117.57	106.94	137.21	190.82
	Zhou et al. [75]	91.83	102.41	96.95	98.75	113.35	90.04	93.84	132.16	158.97
Video	Tekin et al. [65]	102.41	147.72	88.83	125.28	118.02	112.3	129.17	138.89	224.90
	Zhou et al. [76]	87.36	109.31	87.05	103.16	116.18	106.88	99.78	124.52	199.23
	Du et al. [17]	85.07	112.68	104.90	122.05	139.08	105.93	166.16	117.49	226.94
Single-Image	Ours	**54.23**	**61.41**	**60.17**	**61.23**	**79.41**	**63.14**	**81.63**	**70.14**	**107.31**

Input	Method:	Smoking	Taking Photo	Waiting	Walking	Walking Dog	Walking Pair	Avg. (All)	Avg. (6 Actions)
Single-Image	Ionescu et al. [30]	162.14	205.94	170.69	96.60	177.13	127.88	162.14	159.99
	Li & Chan [38]	-	189.08	-	77.60	146.59	-	-	132.20
	Li et al. [39]	-	166.15	-	68.51	132.51	-	-	120.17
	Li et al. [40]	-	163.33	-	73.66	135.15	-	-	121.55
	Zhou et al. [76]	-	-	-	-	-	-	120.99	-
	Rogez & Schmid [53]	-	-	-	-	-	-	121.20	-
	Tekin et al. [64]	-	162.17	-	65.75	130.53	-	-	116.77
	Park et al. [45]	105.78	149.55	125.12	62.64	131.90	96.18	117.34	111.12
	Zhou et al. [75]	106.91	125.22	94.41	79.02	126.04	98.96	107.26	104.73
Video	Tekin et al. [65]	118.42	182.73	138.75	55.07	126.29	65.76	124.97	120.99
	Zhou et al. [76]	107.42	143.32	118.09	79.39	114.23	97.70	113.01	106.07
	Du et al. [17]	120.02	135.91	117.65	99.26	137.36	106.54	126.47	118.69
Single-Image	Ours	**69.29**	**78.31**	**70.27**	**51.79**	**74.28**	**63.24**	**69.73**	**64.53**

Table 1: **Comparison of our approach with state-of-the-art algorithms on *Human3.6m*.** We report 3D joint position errors in mm, computed as the average Euclidean distance between the ground-truth and predicted joint positions. '-' indicates that the results were not reported for the respective action class in the original paper. Note that our method consistently outperforms the baselines.

Method:	3D Pose Error
Sanzari et al. [58]	93.15
Bogo et al. [9]	82.3
Pavlakos et al. [46]	53.2
Ours	**50.12**

Table 2: Comparison of our approach to the state-of-the-art methods that use Procrustes transformation on Human3.6m. We report 3D joint position errors (in mm).

Since results are reported in [9, 58, 46] for the average accuracy over all actions using the Procrustes transformation, as explained in Section 4.2, we do the same when comparing against these methods. Table 2 shows that we also outperform these baselines.

HumanEva. In Table 3, we present the performance of our fusion approach on the HumanEva-I dataset [61]. We adopted the evaluation protocol described in [9, 62, 72, 76] for a fair comparison. As in [9, 62, 72, 76], we measure 3D pose error as the average joint-to-joint distance after alignment by a rigid transformation. Our approach also significantly outperforms the state-of-the-art on this dataset.

Method	S1	S2	S3	Average
Simo-Serra et al. [62]	65.1	48.6	73.5	62.4
Bogo et al. [9]	73.3	59.0	99.4	77.2
Zhou et al. [76]	34.2	30.9	49.1	38.07
Yasin et al. [72]	35.8	32.4	41.6	36.6
Tekin et al. [65]	37.5	25.1	49.2	37.3
Ours	**27.24**	**14.26**	**31.74**	**24.41**

Table 3: Quantitative results of our fusion approach on the Walking sequences of the HumanEva-I dataset [61]. S1, S2 and S3 correspond to Subject 1, 2, and 3, respectively. The accuracy is reported in terms of average Euclidean distance (in mm) between the predicted and ground-truth 3D joint positions.

KTH Multiview Football. In Table 4, we compare our approach to [7, 10, 46, 65] on the KTH Multiview Football II dataset. Note that [7] and [10] rely on multiple views, and [65] makes use of video data. As discussed in Section 4.2, we report the results of two instances of our model: one trained on the standard KTH training data, and one pre-trained on the synthetic 3D human pose dataset of [12] and fine-tuned on the KTH dataset. Note that, while working with a single input image, both instances outperform all the

Method:	[10]	[10]	[7]	[65]	[46]	Ours-NoPretraining	Ours-Pretraining
Input:	Image	Image	Image	Video	Image	Image	Image
Num. of cameras:	1	2	2	1	1	1	1
Pelvis	97	97	-	99	-	66	**100**
Torso	87	90	-	**100**	-	**100**	**100**
Upper arms	14	53	64	74	94	74	**100**
Lower arms	06	28	50	49	80	**100**	88
Upper legs	63	88	75	98	96	**100**	**100**
Lower legs	41	82	66	77	84	77	**88**
All parts	43	69	-	79	-	83.2	**95.2**

Table 4: On KTH Multiview Football II, we compare our method that uses a single image to those of [10, 46, 65] that use either one or two images, the one of [7] that uses two, and the one of [65] that operates on a sequence. As in [7, 10, 46, 65], we measure performance as the percentage of correctly estimated parts (PCP) score. A higher PCP score corresponds to better 3D pose estimation accuracy.

Method:	3D Pose Error
Image-Only	124.13
CM-Only	79.28
Early Fusion	76.41
Late Fusion	74.12
Trainable Fusion	**69.73**

Table 5: Comparison of different fusion strategies and single-stream baselines on Human3.6m. We report the 3D joint position errors (in mm). The fusion networks perform better than those that use only the image or only the confidence map as input. Our trainable fusion achieves the best accuracy overall.

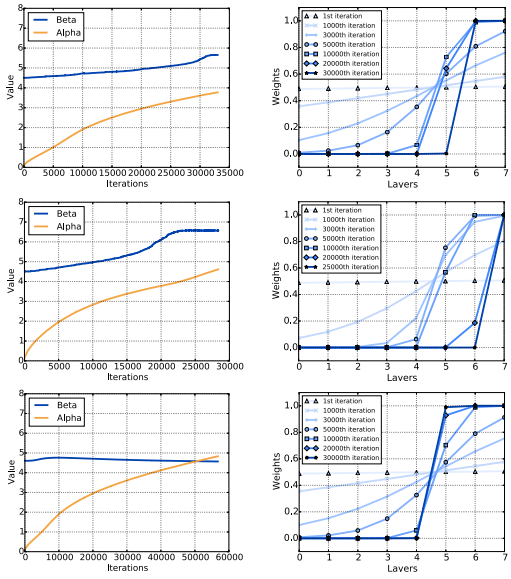
baselines. Note also that pretraining on synthetic data yields the highest accuracy. We believe that this further demonstrates the generalization ability of our method.

In Fig. 5, we provide representative poses predicted by our approach on the Human3.6m, HumanEva and KTH Multiview Football datasets.

4.4. Detailed Analysis

We now analyze two different aspects of our approach. First, we compare our trainable fusion approach to early fusion, depicted in Fig. 2(a), and late fusion, depicted in Fig. 2(c). Then, we analyze the benefits of leveraging both 2D joint locations with their corresponding uncertainty and additional image cues. To this end, we make use of two additional baselines. The first one consists of a single stream CNN regressor operating on the image only. We refer to this baseline as *Image-Only*. The second is a CNN trained to predict 3D pose from only the 2D confidence map (CM) stream. We refer to this baseline as *CM-Only*.

In Table 5, we report the average pose estimation errors on Human3.6m for all these methods. Our trainable fusion strategy yields the best results. Note also that, in general, all fusion strategies outperform the state-of-the-art methods in Table 1. Importantly, the *Image-Only* and *CM-Only* baselines perform worse than our approach, and all fusion-based



(a) (b)

Figure 4: Evolution of **(a)** α and β, and **(b)** the fusion weights in Human3.6m as training progresses. Top row: Directions; Middle row: Discussion; Bottom row: Sitting Down.

methods. This demonstrates the importance of fusing 2D joint location confidence maps along with 3D cues in the image for monocular pose estimation.

In Fig. 4, we depict the evolution throughout the training iterations of **(a)** the parameters α and β that define the weight vector in our trainable fusion framework as given by Eq. 3, and **(b)** the weight vector itself. An increasing value of α, expected due to our regularizer, indicates that fusion becomes sharper throughout the training. An increasing β, which is the typical behavior, corresponds to fusion occurring in the later stages of the network. We conjecture that this is due to the fact that features learned by the image and confidence map streams at later layers become less correlated, and thus yield more discriminative power.

To analyze this further, we show in Fig. 7 the squared Pearson correlation coefficients between all pairs of features of the confidence map stream and of the image stream at the last convolutional layer of our trainable fusion network. As can be seen in the figure, the image and confidence map streams produce decorrelated features that are complementary to each other allowing to effectively account for different input modalities.

4.5. Qualitative Results

In Fig. 6, we present qualitative pose estimation results on the Leeds Sports Pose dataset. We trained our network on the synthetic dataset of [12] and tested on images acquired outdoors in unconstrained settings. The accurate 3D predictions of the challenging poses demonstrate the generalization ability and robustness of our method.

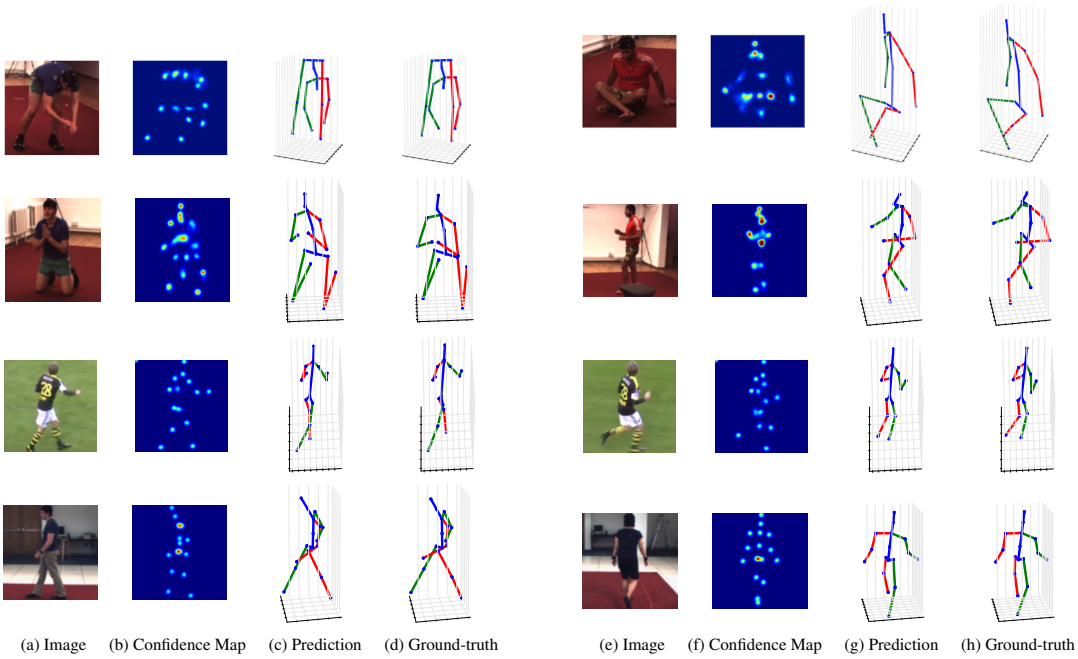


Figure 5: **Pose estimation results on Human3.6m, HumanEva and KTH Multiview Football. (a, e)** Input images. **(b, f)** 2D joint location confidence maps. **(c, g)** Recovered pose. **(d, h)** Ground truth. Note that our method can recover the 3D pose in these challenging scenarios, which involve significant amounts of self occlusion and orientation ambiguity. Best viewed in color.

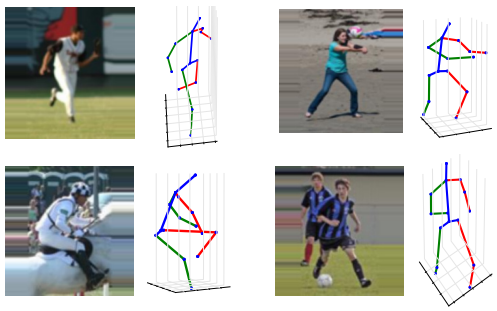


Figure 6: **Pose estimation results on the Leeds Sports Pose dataset.** We show the input image and the predicted 3D pose for four images. Best viewed in color.

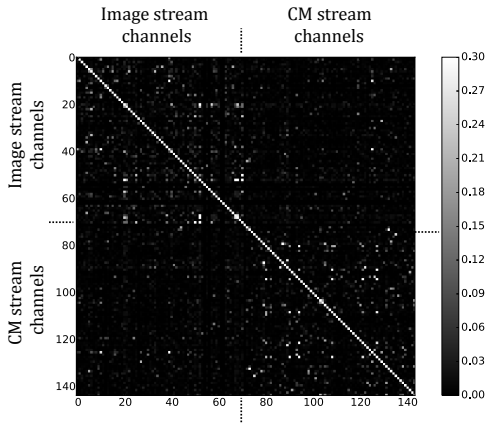


Figure 7: Squared Pearson correlation coefficients (R^2) between each pair of the features learned at the last convolutional layer of our trainable fusion network computed from 128 randomly selected images in Human3.6m. As can be seen in the lower left and upper right submatrices, the feature maps of the image and the confidence map streams are decorrelated.

5. Conclusion

In this paper, we have proposed to fuse 2D and 3D image cues for monocular 3D human pose estimation. To this end, we have introduced an approach that relies on two CNN streams to jointly infer 3D pose from 2D joint locations and from the image directly. We have also introduced an approach to fusing the two streams in a trainable way.

We have demonstrated that the resulting CNN pipeline significantly outperforms state-of-the-art methods on standard 3D human pose estimation benchmarks. Our framework is general and can easily be extended to incorporate other modalities, such as optical flow or body part segmentation. Furthermore, our trainable fusion strategy could be applied to other fusion problems, which is what we intend to do in future work.

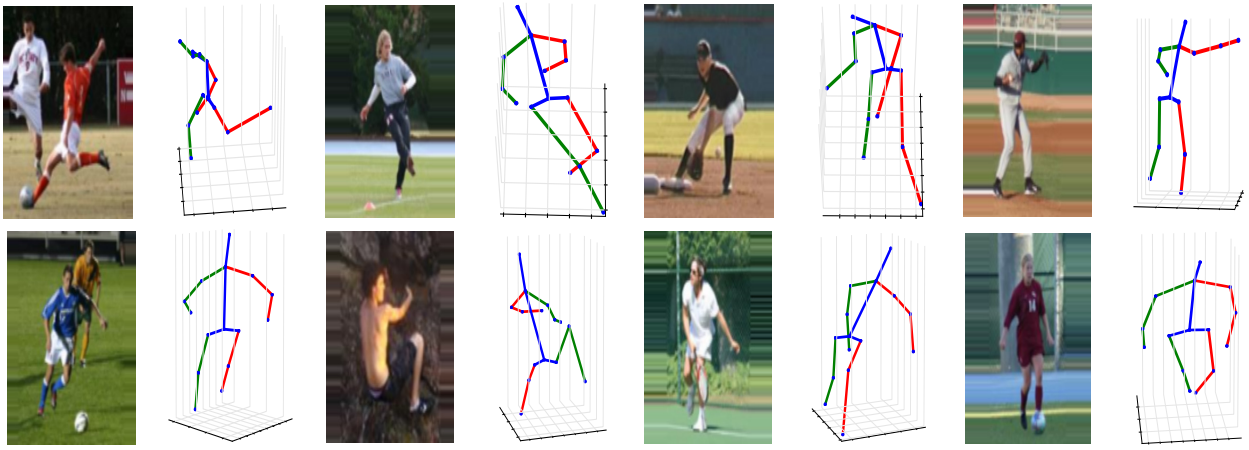


Figure 8: Pose estimation results on LSP. We trained our network on the recently released synthetic dataset of [12] and tested it on the LSP dataset. The quality of the 3D pose predictions demonstrates the generalization of our method. Best viewed in color.

A. Appendix

In this appendix, we analyze the influence of our regularization term encouraging sharp fusion in Eq. 4, provide running time for our algorithm, and show additional qualitative results on the Leeds Sports Pose [33], HumanEva-I [61], Human3.6m [30] and KTH Multiview Football II [10] datasets.

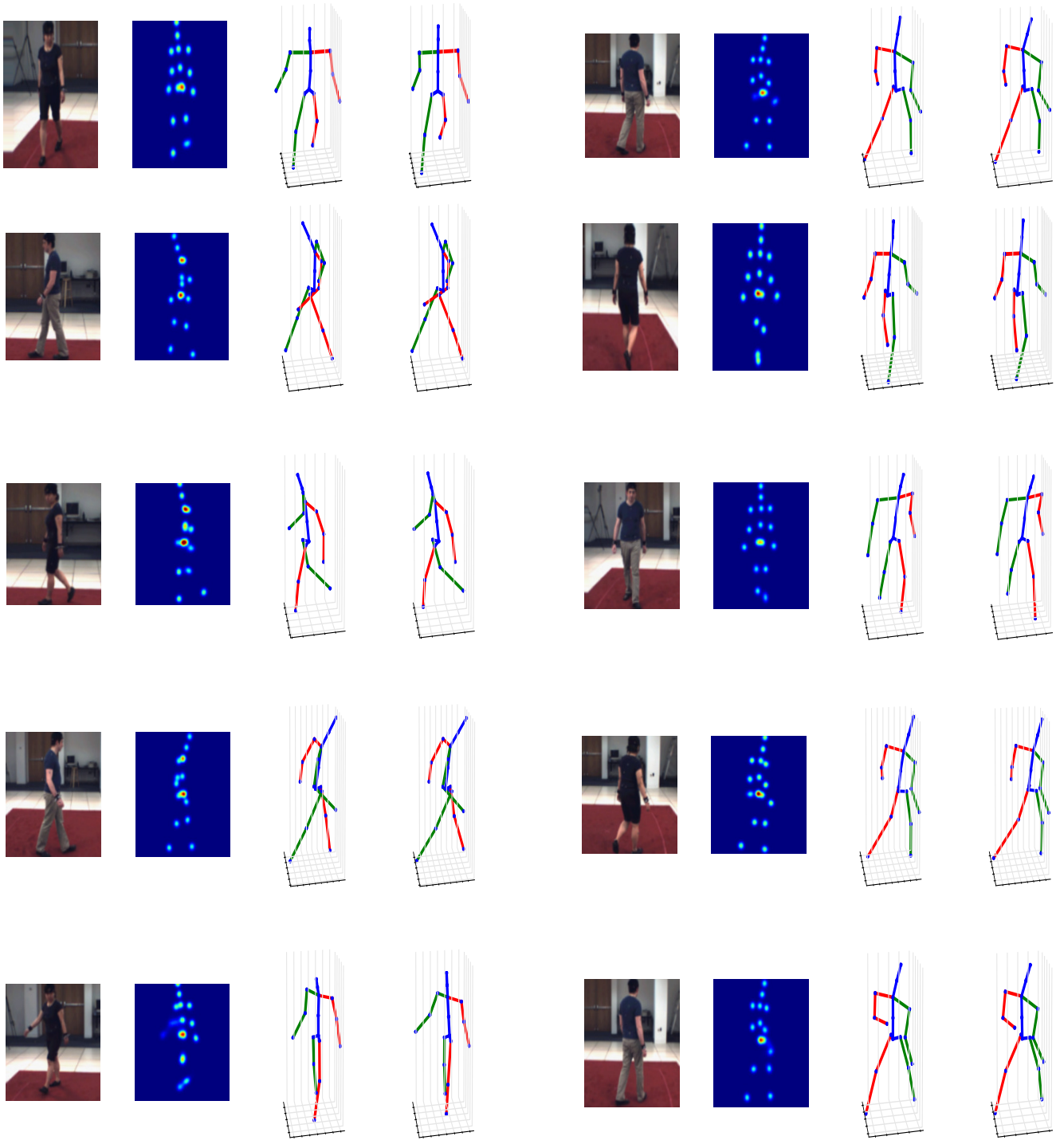
Effect of the regularization. Below, we analyze the effect of the regularization term that encourages sharp fusion in Eq. 4. In the absence of the regularization term, the network mixes the data and fusion streams without necessarily fusing them at a specific layer. As discussed in the main paper, this corresponds to a model with many active parameters. Therefore it is prone to overfitting and computationally less efficient at test-time. In Table 6, we compare the results of our approach with and without this regularization term. For the latter, we do not parametrize the weights of the network with a sigmoid function and do not constrain the network to have a sharp fusion. The results confirm that encouraging sharp fusion yields both better accuracy and faster prediction.

Running time. We carried out our experiments on a machine equipped with an Intel Xeon CPU E5-2680 and an NVIDIA TITAN X Pascal GPU. It takes 90 ms to compute 2D joint location confidence maps and 6 ms to predict 3D pose with our fusion network. Therefore, the total runtime of our method is 0.096 sec/frame (over 10 fps), which compares favorably with the recent model-based methods ranging from 0.04 fps to 1 fps [72, 58, 76].

Method	3D Pose Error	Runtime
Without regularization	68.30	0.013
With regularization	60.17	0.006

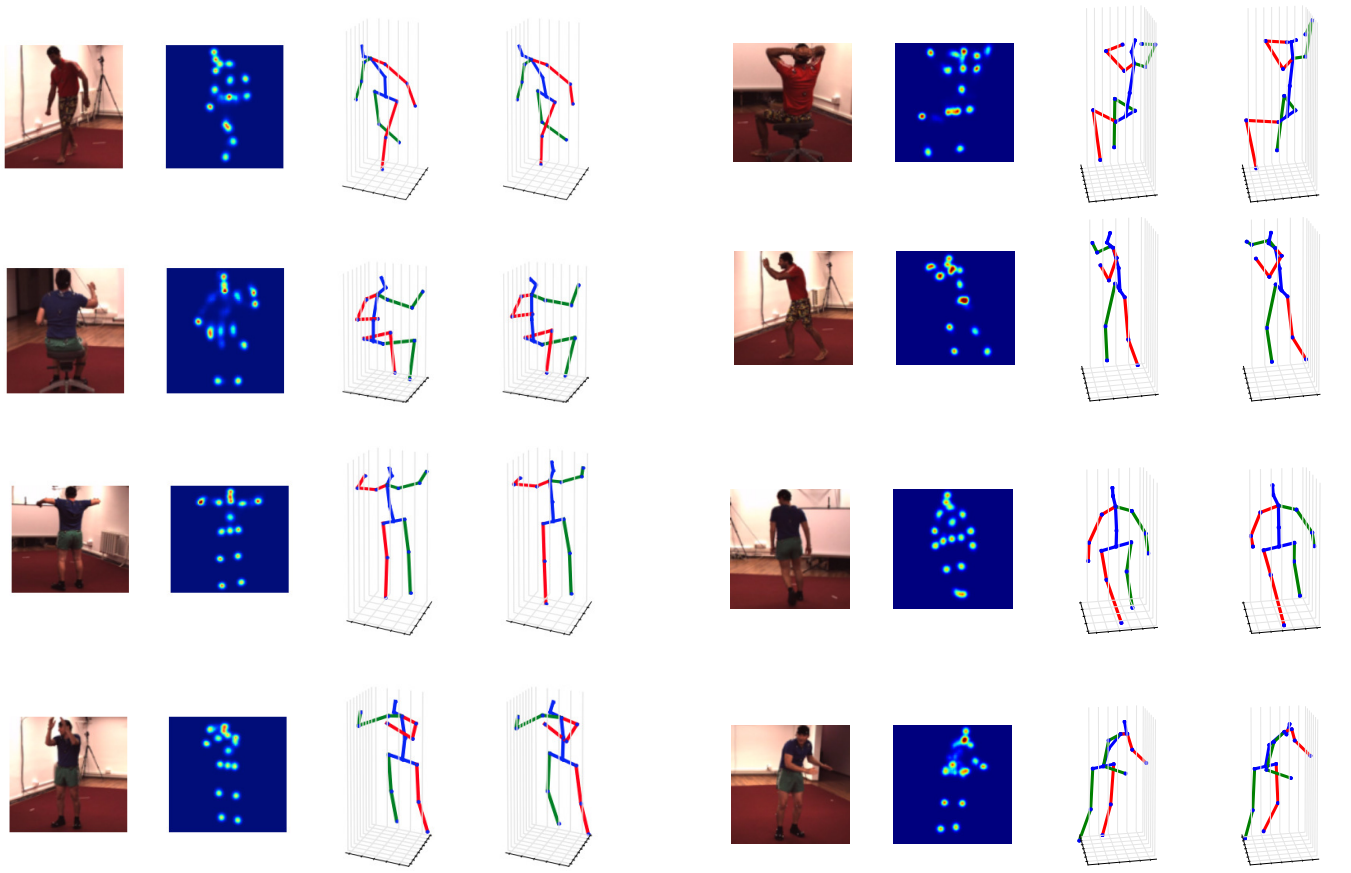
Table 6: Quantitative results of our fusion approach with and without the regularization term encouraging sharp fusion. These experiments were carried out on the *Eating* action class of Human3.6m. 3D pose error is computed as the average Euclidean distance (in milimeters) between the predicted and ground-truth 3D joint positions. Runtime denotes the computational time spent, in sec/frame, during testing for the fusion network with and without the regularization term. With the regularization term, inactive layers are pruned after training, which yields a more efficient network for test-time prediction.

Additional qualitative results. We provide additional qualitative results for the HumanEva [61], Human3.6m [30], and KTH Multiview Football II [10] datasets in Figs. 9 10, and 11, respectively. We further demonstrate that our regressor trained on the recently released synthetic dataset of [12] generalizes well to real images obtained from the Leeds Sports Pose dataset [33] in Fig. 8. Additional qualitative results can be found in the accompanying videos.



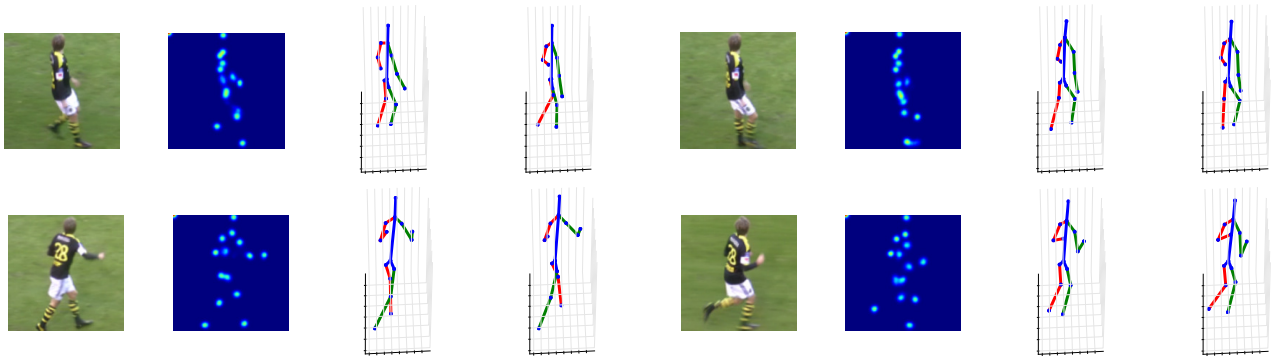
(a) Image (b) Confidence Map (c) Prediction (d) Ground-truth (e) Image (f) Confidence Map (g) Prediction (h) Ground-truth

Figure 9: Pose estimation results on HumanEva-I. **(a, e)** Input images. **(b, f)** 2D joint location confidence maps. **(c, g)** Recovered pose. **(d, h)** Ground truth. Best viewed in color.



(a) Image (b) Confidence Map (c) Prediction (d) Ground-truth (e) Image (f) Confidence Map (g) Prediction (h) Ground-truth

Figure 10: Pose estimation results on Human3.6m. **(a, e)** Input images. **(b, f)** 2D joint location confidence maps. **(c, g)** Recovered pose. **(d, h)** Ground truth. Note that our method can recover the 3D pose in these challenging scenarios, which involve significant amounts of self occlusion and orientation ambiguity. Best viewed in color.



(a) Image (b) Confidence Map (c) Prediction (d) Ground-truth (e) Image (f) Confidence Map (g) Prediction (h) Ground-truth

Figure 11: Pose estimation results on KTH Multiview Football II. **(a, e)** Input images. **(b, f)** 2D joint location confidence maps. **(c, g)** Recovered pose. **(d, h)** Ground truth. Best viewed in color.

References

[1] A. Agarwal and B. Triggs. 3D Human Pose from Silhouettes by Relevance Vector Regression. In *CVPR*, 2004. 1, 2

[2] I. Akhter and M. J. Black. Pose-Conditioned Joint Angle Limits for 3D Human Pose Reconstruction. In *CVPR*, 2015. 2

[3] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele. Multi-View Pictorial Structures for 3D Human Pose Estimation. In *BMVC*, 2013. 2

[4] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *CVPR*, 2014. 4

[5] M. Andriluka, S. Roth, and B. Schiele. Monocular 3D Pose Estimation and Tracking by Detection. In *CVPR*, 2010. 2

[6] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker. Detailed Human Shape and Pose from Images. In *CVPR*, 2007. 2

[7] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3D Pictorial Structures for Multiple Human Pose Estimation. In *CVPR*, 2014. 5, 6, 7

[8] L. Bo and C. Sminchisescu. Twin Gaussian Processes for Structured Prediction. *IJCV*, 2010. 1, 2

[9] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In *ECCV*, 2016. 1, 2, 5, 6

[10] M. Burenius, J. Sullivan, and S. Carlsson. 3D Pictorial Structures for Multiple View Articulated Pose Estimation. In *CVPR*, 2013. 5, 6, 7, 9

[11] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human Pose Estimation with Iterative Error Feedback. In *CVPR*, 2016. 2

[12] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-or, and B. Chen. Synthesizing Training Images for Boosting Human 3D Pose Estimation. In *3DV*, 2016. 5, 6, 7, 9

[13] X. Chen and A. L. Yuille. Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations. In *NIPS*, 2014. 2

[14] Y. Chen, T. Kim, and R. Cipolla. Inferring 3D Shapes and Deformations from Single Views. In *ECCV*, 2010. 2

[15] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured Feature Learning for Pose Estimation. In *CVPR*, 2016. 2

[16] M. Du and R. Chellappa. Face Association Across Unconstrained Video Frames Using Conditional Random Fields. In *ECCV*, 2012. 2

[17] Y. Du, Y. Wong, Y. Liu, F. Han, Y. Gui, Z. Wang, M. Kankanhalli, and W. Geng. Marker-Less 3D Human Motion Capture with Monocular Image Sequence and Height-Maps. In *ECCV*, 2016. 5, 6

[18] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing Action at a Distance. In *ICCV*, pages 726–733, October 2003. 2

[19] A. Elhayek, E. Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt. Efficient Convnet-Based Marker-Less Motion Capture in General Scenes with a Low Number of Cameras. In *CVPR*, 2015. 2

[20] X. Fan, K. Zheng, Y. Zhou, and S. Wang. Pose Locality Constrained Representation for 3D Human Pose Reconstruction. In *ECCV*, 2014. 2

[21] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Optimization and Filtering for Human Motion Capture. *IJCV*, 2010. 1, 2

[22] S. Gammeter, A. Ess, T. Jaeggli, K. Schindler, B. Leibe, and L. Van Gool. Articulated Multi-Body Tracking Under Egomotion. In *ECCV*, 2008. 2

[23] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient Regression of General-Activity Human Poses from Depth Images. In *ICCV*, 2011. 2

[24] G. Gkioxari, A. Toshev, and N. Jaitly. Chained Predictions Using Convolutional Neural Networks. In *ECCV*, 2016. 2

[25] P. Guan, A. Weiss, A. Balan, and M. Black. Estimating Human Shape and Pose from a Single Image. In *ICCV*, 2009. 2

[26] N. R. Howe. A Recognition-Based Motion Capture Baseline on the Humaneva II Test Data. *MVA*, 2011. 2

[27] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model. In *ECCV*, 2016. 2

[28] C. Ionescu, J. Carreira, and C. Sminchisescu. Iterated Second-Order Label Sensitive Pooling for 3D Human Pose Estimation. In *CVPR*, 2014. 2

[29] C. Ionescu, F. Li, and C. Sminchisescu. Latent Structured Models for Human Pose Estimation. In *ICCV*, 2011. 2

[30] C. Ionescu, I. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *PAMI*, 2014. 1, 5, 6, 9

[31] A. Jain, T. Thormahlen, H. Seidel, and C. Theobalt. Moviereshape: Tracking and Reshaping of Humans in Videos. In *SIGGRAPH*, 2010. 2

[32] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler. Learning Human Pose Estimation Features with Convolutional Networks. In *ICLR*, 2014. 2

[33] S. Johnson and M. Everingham. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *BMVC*, 2010. 5, 9

[34] A. Kanaujia, C. Sminchisescu, and D. N. Metaxas. Semi-Supervised Hierarchical Models for 3D Human Pose Reconstruction. In *CVPR*, 2007. 1

[35] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimisation. In *ICLR*, 2015. 4

[36] A. G. Kirk and J. F. O. D. A. Forsyth. Skeletal Parameter Estimation from Optical Motion Capture Data. In *CVPR*, 2005. 2

[37] I. Kostrikov and J. Gall. Depth Sweep Regression Forests for Estimating 3D Human Pose from Images. In *BMVC*, 2014. 2

[38] S. Li and A. Chan. 3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network. In *ACCV*, 2014. 1, 2, 5, 6

[39] S. Li, W. Zhang, and A. B. Chan. Maximum-Margin Structured Learning with Deep Networks for 3D Human Pose Estimation. In *ICCV*, 2015. 2, 5, 6

[40] S. Li, W. Zhang, and A. B. Chan. Maximum-Margin Structured Learning with Deep Networks for 3D Human Pose Estimation. In *IJCV*, 2016. 5, 6

[41] G. Mori and J. Malik. Estimating Human Body Configurations Using Shape Context Matching. In *ECCV*, 2002. 2

[42] G. Mori and J. Malik. Recovering 3D Human Body Configurations Using Shape Contexts. *PAMI*, 2006. 2

[43] A. Newell, K. Yang, and J. Deng. Stacked Hourglass Networks for Human Pose Estimation. In *ECCV*, 2016. 2, 4

[44] D. Ormoneit, H. Sidenbladh, M. Black, T. Hastie, and D. Fleet. Learning and Tracking Human Motion Using Functional Analysis. In *IEEE Workshop on Human Modeling, Analysis and Synthesis*, 2000. 2

[45] S. Park, J. Hwang, and N. Kwak. 3D Human Pose Estimation Using Convolutional Neural Networks with 2D Pose Information. In *ECCV Workshops*, 2016. 2, 5, 6

[46] G. Pavlakos, X. Zhou, K. Derpanis, and K. Daniilidis. Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose. In *arXiv preprint arXiv:1611.07828*, 2016. 1, 2, 5, 6, 7

[47] T. Pfister, J. Charles, and A. Zisserman. Flowing Convnets for Human Pose Estimation in Videos. In *ICCV*, 2015. 2

[48] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele. Deepcut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. In *CVPR*, 2016. 2

[49] G. Pons-Moll, A. Baak, J. Gall, L. Leal-Taixe, M. Muller, H. Seidel, and B. Rosenhahn. Outdoor Human Motion Capture Using Inverse Kinematics and Von Mises-Fisher Sampling. In *ICCV*, 2011. 2

[50] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon. Metric Regression Forests for Correspondence Estimation. *IJCV*, 2015. 2

[51] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3D Human Pose from 2D Image Landmarks. In *ECCV*, 2012. 2

[52] G. Rogez, J. Rihan, C. Orrite, and P. Torr. Fast Human Pose Detection Using Randomized Hierarchical Cascades of Rejectors. *IJCV*, 2012. 2

[53] G. Rogez and C. Schmid. Mocap Guided Data Augmentation for 3D Pose Estimation in the Wild. In *NIPS*, 2016. 5, 6

[54] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015. 4

[55] R. Rosales and S. Sclaroff. Infering Body Pose Without Tracking Body Parts. In *CVPR*, June 2000. 2

[56] R. Rosales and S. Sclaroff. Learning Body Pose via Specialized Maps. In *NIPS*, 2002. 1, 2

[57] M. Salzmann and R. Urtasun. Combining Discriminative and Generative Methods for 3D Deformable Surface and Articulated Pose Reconstruction. In *CVPR*, June 2010. 2

[58] M. Sanzari, V. Ntouskos, and F. Pirri. Bayesian Image Based 3D Pose Estimation. In *ECCV*, 2016. 2, 5, 6, 9

[59] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient Human Pose Estimation from Single Depth Images. *PAMI*, 35(12):2821–2840, 2013. 2

[60] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic Tracking of 3D Human Figures Using 2D Image Motion. In *ECCV*, 2000. 1, 2

[61] L. Sigal and M. Black. Humaneva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion. Technical report, Department of Computer Science, Brown University, 2006. 5, 6, 9

[62] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer. A Joint Model for 2D and 3D Pose Estimation from a Single Image. In *CVPR*, 2013. 2, 5, 6

[63] E. Simo-Serra, A. Ramisa, G. Alenya, C. Torras, and F. Moreno-Noguer. Single Image 3D Human Pose Estimation from Noisy Observations. In *CVPR*, 2012. 2

[64] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured Prediction of 3D Human Pose with Deep Neural Networks. In *BMVC*, 2016. 1, 2, 5, 6

[65] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua. Direct Prediction of 3D Body Poses from Motion Compensated Sequences. In *CVPR*, pages 991–1000, 2016. 1, 2, 5, 6, 7

[66] A. Toshev and C. Szegedy. Deeppose: Human Pose Estimation via Deep Neural Networks. In *CVPR*, 2014. 2

[67] R. Urtasun and T. Darrell. Sparse Probabilistic Regression for Activity-Independent Human Pose Inference. In *CVPR*, 2008. 1, 2

[68] R. Urtasun, D. Fleet, and P. Fua. 3D People Tracking with Gaussian Process Dynamical Models. In *CVPR*, 2006. 1, 2

[69] J. Valmadre and S. Lucey. Deterministic 3D Human Pose Estimation Using Rigid Structure. In *ECCV*, 2010. 2

[70] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional Pose Machines. In *CVPR*, 2016. 2

[71] Y. Yang and D. Ramanan. Articulated Pose Estimation with Flexible Mixtures-Of-Parts. In *CVPR*, 2011. 2

[72] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall. A Dual-Source Approach for 3D Pose Estimation from a Single Image. In *CVPR*, 2016. 2, 5, 6, 9

[73] T.-H. Yu, T.-K. Kim, and R. Cipolla. Unconstrained Monocular 3D Human Pose Estimation by Action Detection and Cross Modality Regression Forest. In *CVPR*, 2013. 2

[74] F. Zhou and F. de la Torre. Spatio-Temporal Matching for Human Detection in Video. In *ECCV*, 2014. 2

[75] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei. Deep Kinematic Pose Regression. In *ECCV Workshops*, 2016. 2, 5, 6

[76] X. Zhou, M. Zhu, S. Leonardos, K. Derpanis, and K. Daniilidis. Sparseness Meets Deepness: 3D Human Pose Estimation from Monocular Video. In *CVPR*, 2016. 1, 2, 5, 6, 9