Thin-Slicing Network: A Deep Structured Model for Pose Estimation in Videos

Jie Song[1] Limin Wang[2] Luc Van Gool[2] Otmar Hilliges[1]
[1]AIT Lab, ETH Zurich [2]Computer Vision Lab, ETH Zurich

Abstract

Deep ConvNets have been shown to be effective for the task of human pose estimation from single images. However, several challenging issues arise in the video-based case such as self-occlusion, motion blur, and uncommon poses with few or no examples in the training data. Temporal information can provide additional cues about the location of body joints and help to alleviate these issues. In this paper, we propose a deep structured model to estimate a sequence of human poses in unconstrained videos. This model can be efficiently trained in an end-to-end manner and is capable of representing the appearance of body joints and their spatio-temporal relationships simultaneously. Domain knowledge about the human body is explicitly incorporated into the network providing effective priors to regularize the skeletal structure and to enforce temporal consistency. The proposed end-to-end architecture is evaluated on two widely used benchmarks for video-based pose estimation (Penn Action and JHMDB datasets). Our approach outperforms several state-of-the-art methods. [1]

1. Introduction

Estimating human poses is one of the core problems in computer vision and has many applications in the life-sciences, computer animation and the growing fields of robotics, augmented and virtual reality. Accurate pose estimates can also drastically improve the performance of activity recognition and high-level analysis of videos (cf. [14, 34, 36]). Recent pose estimation methods have exploited deep convolutional networks (ConvNets) for body-part detection in single, fully unconstrained images [2, 17, 18, 22, 31, 32, 35]. While demonstrating the feasibility of detection-based pose estimation from images taken under general conditions, such methods still struggle with several challenging aspects including the diversity of human appearance and self-symmetries. Several methods [2, 37] have explicitly incorporated geometric constraints among body parts into such frameworks, ensuring spatial consistency and penalizing physically impossible solutions (cf. Figure 1, (c)).

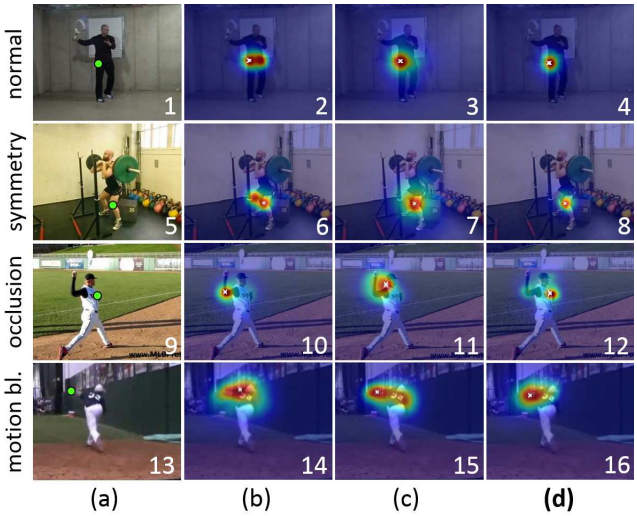

Figure 1. Our method incorporates spatio-temporal information into a single end-to-end trainable network architecture, aiming to deal with challenging problems such as (self-)occlusions, motion blur, and uncommon poses. Taking fully unconstraind images as input (a), we regress body-part locations with standard ConvNet layers (b). Spatial inference helps in overcoming confusion due to symmetric body parts (c). Our spatio-temporal inference layer (d) can deal with extreme cases where spatial information only fails (cf. 11 vs 12, 15 vs 16) *and* improves prediction accuracies for unary terms due to repeating measurements by temporal propagation of joint position estimates (3 vs 4).

In this paper we consider the comparatively less studied problem of human pose estimation from unconstrained *videos* [11, 20, 39, 42]. While inheriting many properties from image-based pose estimation, it also brings new challenges. In particular, unconstrained videos such as those found in online portals, contain many frames with occlusions, unusual poses, and motion blur (see Figure 1). These issues continue to limit the accuracy of joint detection even if taking priors about the spatial configuration of the human skeleton into consideration, and often result in visible jitter if such models are applied directly to video sequences.

To tackle these problems, we propose to incorporate spa-

[1]Code and models are available at https://github.com/JieSong89/thin-slicing-network.

tial and temporal modeling into deep learning architectures. The proposed model is based on a simple observation: human motion exhibits high temporal consistency, which may be captured by optical flow warping [20, 39, 42] and spatio-temporal inference [34, 36]. Specifically, we incorporate a spatio-temporal relational model into the ConvNet and develop a new deep structured architecture which we call *Thin-Slicing Network*. Our model allows for end-to-end training of body part regressors and spatio-temporal relational models in a *unified framework*. This improves generalization capabilities by regularizing the learning process both spatially and temporally. We deploy a fully ConvNet for initial part detection. A flow warping layer propagates joint prediction heat maps temporally and a novel inference layer, performing message passing on arbitrary loopy graphs along both spatial and temporal edges, is introduced.

In consequence, our approach can deal with many challenging situations arising in unconstrained video, and outperforms both pure joint-position estimation methods and those incorporating spatial priors only. Figure 1 illustrates how our approach can accurately predict joint positions in difficult situations of full occlusion (3^{rd} row, given visibility in adjacent frames) or severe motion blur (4^{th} row, by exploiting temporal consistency). Last but not least, the model also improves predictions in relatively simple cases (see Figure 1, 1^{st} and 2^{nd} row). This can be explained by optimizing of several correlated but different frames through the entire architecture jointly, which not only learns weights of the inference layers, but also refines the underlying ConvNet-based part regressors, resulting in more accurate joint detections.

In summary our main contributions are: (i) A structured model captures the inherent consistency of human poses in video sequences based on a loopy spatio-temporal graph. Our approach does not rely on explicit human motion priors but leverages dense optical flow to exploit image evidence from adjacent frames. (ii) An efficient and flexible inference layer performs message passing along the spatial and temporal graph edges and significantly reduces joint position uncertainty. (iii) The entire architecture integrates a ConvNet-based joint regressors and a high-level structured inference model in a unified framework which can be optimized in an end-to-end manner. (iv) Our method significantly improves the state-of-the-art performance on two widely used video based pose estimation benchmarks: the Penn Action dataset [40] and the JHMDB dataset [14].

2. Related work

Pose estimation from **single images** has benefitted tremendously from leveraging structural models such as tree-structured pictorial models [1] and part-based models [15, 21, 23, 38], encoding the relationships between articulated joints. While capturing kinematic correlations, such models are prone to errors such as double-counting part evidence. More expressive loopy graph models, allowing for cyclic joint dependencies have been proposed to better capture symmetry and long-range correlation [5, 25, 28, 30]. Since exact inference in cyclic graphs is generally speaking intractable, approximate inference methods like loopy belief propagation are typically used.

The above methods are based on hand-crafted features and are sensitive to (the limits of) their representative power. More recently, **convolutional deep learning** architectures have been deployed to learn richer and more expressive features directly from data [2, 18, 22, 31, 32], outperforming prior work. Toshev et al. [32] directly regress the joint coordinates from images. Follow-up work suggests that regressing full image confidence maps as intermediate representation can be more effective [2, 31]. While multi-stage convolutional operations can capture information in large receptive fields, they still lack the ability to fully model skeletal structure in their predictions.

Several approaches to refine confidence maps have been proposed. First, additional convolutional layers taking joint heat-maps as input can be added to learn implicit spatial dependencies without requiring explicit articulated human body priors [4, 31, 35]. Second, [2, 22] explicitly resort to graphical models to post-process regressed confidence maps. However, the parameters of part regression networks and spatial inference are learned independently [2, 22]. In [37] an end-to-end trainable framework, combining convolutional operations and spatial refinement is proposed. Our work not only incorporates spatial information but also models temporal dependencies.

Pose estimation in videos brings new challenges (illustrated in Figure 1) and requires the coupling of parts across frames to ensure accurate and temporally stable predictions. Early work initializes a temporal tracker from few predicted poses in the sequence's initial frames [27] but suffers from pose drift. Tracking-by-detection schemes have been used to more robustly estimate poses in videos [8, 19, 24]. Researchers have also attempted to design spatio-temporal graphs to capture motion in short video sequences [3, 7, 16, 26, 29, 33, 34, 36, 39]. However, modeling spatial and temporal dependencies explicitly results in highly interconnected models (i.e., loopy graphs with large tree-width) and exact inference becomes again intractable. One solution is to resort to approximate inference, for instance using sampling based approaches [29, 33] or loopy belief propagation [7, 16]. Alternatively, approximating the original large loopy model into one or multiple simplified tree-based models allows for efficient exact inference [3, 39].

Some recent deep learning methods aide predictions in the current frame with information from its neighbors [13]. Similar to our approach, [20] directly propagates joint position estimates from previous to the current frame via opti-

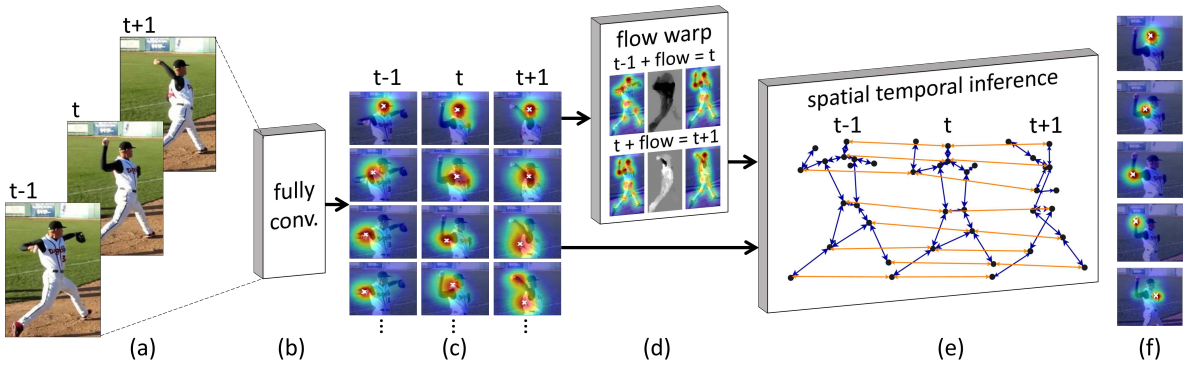

Figure 2. **Schematic overview of Thin-Slicing Network architecture**. Our model takes a small number of adjacent frames as input (a) and fully convolutional layers (b) regress initial body joint position estimates (c). We compute dense optical flow between neighboring frames to propagate joint position estimates through time. A flow based warping layer aligns joint heat-maps to the current frame (d). A spatio-temporal inference layer performs iterative message passing along both spatial and temporal edges of the loopy pose configuration graph (e) and computes final joint position estimates (f). For convenience of illustration, we only plot one target frame.

cal flow. Warped heatmaps from multiple nearby frames are combined as weighted average. Chain models [11] can capture longer temporal dependencies but makes assumptions about regular motion patterns. Our approach also incorporates spatio-temporal models into deep ConvNets but differs in that it (i) explicitly models the spatial configuration of human poses; (ii) regularizes temporal joint positions using dense optical flow via (iii) a novel inference layer, performing message passing on general loopy spatio-temporal graphs; (iv) and is end-to-end trainable.

3. Thin-Slicing Networks

Figure 2 shows an overview of our proposed network architecture, consisting of several interconnected layers. Given a thin-slice of a video sequence (i.e., a small number of adjacent frames), a spatial fully ConvNet first regresses joint confidence maps (heat-maps) of joint positions for each input frame (Figure 2 (c)). These heat-maps are sent into a *flow warping* layer and a *spatio-temporal inference* layer. The flow warping layer (Figure 2 (d)) warps the body part heat-maps via dense optical flow so that they align with its neighboring frame. Finally, both the warped and the current frame heat-maps pass through the *spatio-temporal inference* layer (Figure 2 (e)). This layer conducts inference between body parts spatially and temporally, producing the final joint position estimates (Figure 2 (f)).

3.1. Fully convolutional joint regression layer

Several recent works regress heat-maps of body joints via ConvNets [2, 18, 22, 31, 35, 17]. Such models usually consist entirely of convolutional operations combined with spatial pooling layers. We leverage such a ConvNet [35] as basis for our architecture. More specifically as joint detection layers shown in Figure 2 (b). Such models have already demonstrated the ability to capture local appearance properties and outperform hand-designed shallow features by large margins, but occlusions, (self-)symmetries and motion blur still pose significant challenges (cf. Figure 1). In order to alleviate these problems, a novel spatio-temporal message passing layer (Sec. 3.3) is proposed and incorporated into the network for end-to-end training.

3.2. Flow warping layer

While our goal is to improve temporal stability of joint predictions, we do not incorporate an explicit motion model (since human motion tends to be too unpredictable) but instead rely on dense optical flow to propagate information temporally. The joint detection heat-maps, produced by fully convolutional layers, is passed through the flow warping layer to align heat-maps from one frame to the targeted neighbor (Figure 2 (d)). Pixel-wise flow vectors are used to align confidence estimates in neighboring frames to the target frame by shifting confidence values along the track directions. Next, these warped heat-maps serve as input to the spatio-temporal inference layer.

3.3. Spatio-temporal inference layer

Incorporating domain specific knowledge into deep networks has been proven to be effective in many vision tasks such as object detection [10] and semantic segmentation [41]. In this work, we propose to explicitly incorporate spatio-temporal dependencies into an end-to-end trainable framework.

Modeling

Let $G = (V, E)$ be a graph as shown in Figure 2 (e), with vertices V and edges $E \subseteq V \times V$ denoting the spatio-temporal structure of a human pose. $K = |V|$ is the number

of body parts, and $i \in \{1, ..., K\}$ is the i^{th} part. Each vertex corresponds to one of the body parts (i.e., head, shoulders), and each edge represents a connection between two of these parts spatially (blue arrows in Figure 2 (e)) or between the same part but distributed temporally (yellow arrows in Figure 2 (e)). We denote these edges as E_s and E_f respectively. Given an image I, a pose p with respect to this graph G is defined as a set of 2D coordinates in the image space representing the positions of the different body parts: $p = \{p_i = (x_i, y_i) \in \mathbb{R}^2 : \forall i \in V\}$. The single-image pose estimation problem then can be formulated as the maximization of the following score $S(I, p)$ for a pose p given an image I:

$$S(I, p) = \sum_{i \in V} \phi_i(p_i|I) + \sum_{(i,j) \in E_s} \psi_{i,j}(p_i, p_j), \quad (1)$$

where $\phi_i(p_i|I)$ is the unary term for the body part i at the position p_i in image I and $\psi_{i,j}(p_i, p_j)$ is the pairwise term modeling the spatial compatibility of two neighboring parts i and j. The unary term provides confidence values of part i based on the local appearance and it is modeled by the fully ConvNet (Sec. 3.1). For pairwise term we use a spring energy model to measure the deformation cost, where $\psi_{i,j}(p_i, p_j)$ is defined as $w_{i,j} \cdot d(p_i - p_j)$. With standard quadratic deformation constraints $d(p_i - p_j) = [\Delta x \quad \Delta x^2 \quad \Delta y \quad \Delta y^2]^T$, where $\Delta x = x_i - x_j$ and $\Delta y = y_i - y_j$ are the relative positions of part i with respect to part j. The parameter $w_{i,j}$ encodes rest location and rigidity of each spring, which can be learned from data alongside the remaining network parameters.

Given a slice of a video sequence $\mathbb{I} = (I_1, I_2, ..., I_T)$ as shown in Figure 2 (a), the temporal links (yellow arrows in Figure 2 (e)) are introduced among neighboring frames in order to impose temporal consistency for estimating poses $\mathbb{P} = (p^1, p^2, ..., p^T)$. The objective score function of the entire slice with temporal constrains is then given by:

$$S(\mathbb{I}, \mathbb{P})_{slice} = \sum_{t=1}^{T} S(I^t, p^t) + \sum_{(i,i^*) \in E_f} \psi_{i,i^*}(p_i, p'_{i^*}). \quad (2)$$

Here $S(I^t, p^t)$ is the score function for each frame as defined in Eq. (1). The pairwise term $\psi_{i,i^*}(p_i, p'_{i^*})$ regularizes the temporal consistency of the part i in neighboring frames. Specifically, here $p'_{i^*} = p_{i^*} + f_{i^*,i}(p_{i^*})$ and $f_{i^*,i}(p_{i^*})$ is the optical flow evaluated at p_{i^*}. This term denotes the flow warping process in which pixel-wise flow tracks are applied to align confidence values in neighboring frames to the target frame. We use the same quadratic spring model as above to penalize the estimation drift between these neighboring frames.

Inference

Inference corresponds to maximizing S_{slice} defined in Eq. (2) over p for the image sequence slice. When the relational graph $G = (V, E)$ is a tree-structured graph, exact belief propagation can be applied efficiently by one pass of dynamic programming in polynomial time. However, for cases in which the factor graph is not tree-structured but contains cycles, the belief propagation algorithm is not applicable as no leaf-to-root order can be established. However, loopy belief propagation algorithms such as the Max-Sum algorithm make approximate inference possible in intractable loopy models [9]. Empirical performance has consistently been reported to be excellent across various problems [37, 28]. More precisely, in our case at each iteration a part i sends a message to its neighbors and also receives reciprocal messages along the edges in G:

$$score_i(p_i) \leftarrow \phi_i(p_i|I) + \sum_{k \in child(i)} m_{ki}(p_i), \quad (3)$$

where $child(i)$ is defined as the set of children of part i. The local $score_i(p_i)$ is the sum of the unary terms $\phi_i(p_i|I)$ and the messages collected from its all children. The messages $m_{ki}(p_i)$ sent from body part k to part i are given by:

$$m_{ki}(p_i) \leftarrow \max_{p_k} (score_k(p_k) + \psi_{k,i}(p_k, p_i)). \quad (4)$$

Eq. (4) computes for every location of part i the best scoring location of its child k, based on the score of part k and the spring model between i and k. This cost maximization process can be efficiently solved via the generalized distance transforms [6], reducing the computational complexity to be linear in the number of possible part locations, which is the size of the regressed heat-map from the fully ConvNet (Sec. 3.1). This inference process could be operated by several iterations till convergence.

In our implementation of the spatio-temporal message passing layer, for the first iteration, the local score for each part is initialized by its corresponding unary term obtained from the regressor layers (Figure 2 (c)). The inference process is illustrated in Figure 2 (e). The children of one node could be either adjacent parts in the same frame or the same part in the neighboring frames. For the first case, the heat-maps of other parts are directly taken as input to the generalized distance transform, while for the second case the $score_k(p_k)$ is the heat-map after flow warping (Figure 2 (d)). We implement message passing in a broadcasting style where messages are passed simultaneously across every edge in both directions.

Specifically, for each part i, Eq. (4) computes the best score from its child k. The forward of this maximization process is efficiently solved via the generalized distance transform. The resulting Max location p^* for each pixel is stored. Similar to the Max Pooling operation, the backprop-

agation of Eq. (4) is achieved through sub-gradient decent:

$$\frac{\partial m_{ki}(p_i)}{\partial score_k(p_k)} = \begin{cases} 1 & \text{if } p_k = p^*, \\ 0 & \text{otherwise.} \end{cases}$$

$$\frac{\partial m_{ki}(p_i)}{\partial \psi_{k,i}(p_k, p_i)} = \begin{cases} 1 & \text{if } p_k = p^*, \\ 0 & \text{otherwise.} \end{cases}$$

The gradient for the parameter of the spring model w_{ki} is calculated by $\frac{\partial m_{ki}(p_i)}{\partial w_{ki}} = \frac{\partial m_{ki}(p_i)}{\partial \psi_{k,i}(p_k,p_i)} d(p_k - p_i)$, where $d(p_k - p_i)$ is the quadratic displacement.

4. Learning

The learning of Thin-Slicing Network is decomposed into two stages: (1) Training fully convolutional layers and (2) Joint training with flow warping and inference layers.

Training fully convolutional layers As discussed in Sec. 3.1, we deploy fully convolutional layers as the basic regressor to produce the belief maps for all the body parts in the sequence. As shown in Figure 2 (c), every pixel position has a confidence value for each joint. The ground truth heat-map for a part i is written as $b^i_*(Y_i = p)$, which is created by placing a Gaussian peak at the center location of the part. In our implementation, we set peak values as 1 and the background as 0. We aim to minimize the l_2 distance between the predicted and ideal belief maps for each part, yielding the loss function:

$$f = \sum_{i=1}^{K} \sum_{p} \left\| b^i(p) - b^i_*(p) \right\|^2 . \tag{5}$$

We use the stochastic gradient descent algorithm to train these fully convolutional layers with dropouts.

Joint training with flow warping and inference layers For the second stage of training, the unified end-to-end model (Figure 2) is jointly trained by initializing the weights of the fully convolutional layers with the pre-trained parameters. In this training stage, instead of using l_2 distance loss, we use the hinge loss during optimization. The final loss is defined in Eq. (6), $I^i(p)$ is an indicator which is equal to 1 if the pixel lies within a circle of radius r centered on the ground truth joint position, otherwise it is equal to -1:

$$f = \sum_{i=1}^{K} \sum_{p} \max(0, 1 - b^i(p) \cdot I^i(p)). \tag{6}$$

The parameters in the inference layer are differentiable and hence can be trained end-to-end alongside the other weights in the network by stochastic gradient descent.

5. Experiments

In this section we present results from our experimental evaluation of the proposed architecture performed on standard datasets. First we introduce the datasets and the implementation details as used during our experiments. Furthermore, we compare performance of our method with two separate baselines: a fully convolutional network and a ConvNet with spatial inference only. Finally, we compare our results with other state-of-the-art approaches across datasets.

5.1. Datasets

We conduct experiments on the Penn Action [40] and JHMDB [14] datasets, both standard datasets to evaluate video-based pose estimation.

Penn Action dataset the Penn Action dataset [40] is one of the largest datasets with full annotations of human joints in videos, containing 2326 unconstrained videos depicting 15 different action categories and the annotations include 13 human joints for each image. An additional occlusion label for each joint is also provided. We follow the original paper [40] to split the data into training and testing subsets in a roughly half-half manner. In total there are around 90k images for training and 80k images for testing.

JHMDB dataset The JHMDB dataset [14] contains 928 videos and 21 action classes. The dataset provides three different splits of training and testing, and we report the average performance over these three splits for all evaluations on this dataset. We also conduct experiments on a subset of this dataset (sub-JHMDB dataset) to compare with other state-of-the-art methods. This subset contains 316 clips with 12 action categories. In this subset the whole human body is in the image and all joints are annotated with ground truth positions.

5.2. Implementation Details

Data augmentation to introduce more variation in the training data and thus reducing overfitting, we augment the data by rotating images between -90 to 90 degrees chosen randomly and by scaling by a random factor between 0.5 to 2. When pre-training the fully convolutional layers, the inputs to the network are the cropped image patch around the center of persons with random shifts. For end-to-end training with the flow warping and spatio-temporal message passing layer, the input patches for the sequence are controlled to have the same pre-processing.

Network parameter settings for the fully convolutional layers, we deploy the network structure based on [35]. This model has a multiple-stage structure which is designed to

alleviate the problem of vanishing gradients. We use an input size of 368 $\times$ 368 px in order to cover sufficient context. The batch size is set to 20 for pre-training the convolutional layers and 6 for jointly training the unified network respectively when the thin-slicing is 5 frames. The learning rates are initialized as 0.0005 for the first stage of training and dropped by a factor of 3 every 20k iterations. For end-to-end training, the learning rate is set to be lower (0.0001) and is dropped every 5k iterations also by a factor of 3. The dropout rate is set to 0.5 for the first stage and increased to 0.7 for the second stage with flow warping and message passing layers to reduce potential effects of overfitting. The fully ConvNet is trained for 10 epochs for initialization. The unified end-to-end model typically converges after 3-4 epochs. The flow warping layer takes resized optical flow images of the same size as the heat-maps as input with their values rescaled by the same scaling factor.

For the spatio-temporal message passing layer, we initialize the weight of the quadratic term to 0.01 and the first-order term to 0 for the generalized distance transform algorithm [6]. Please note that setting the normalization terms when collecting messages sent from children can help stabilize the training process. A similar observation is also reported in [37]. We find that three iterations of approximate inference already provides satisfactory results and if not specified otherwise message passing is stopped after three iterations in our experiments.

Edge connections in the graph The spatio-temporal loopy structure used in this implementation is visualized in Figure 2 (e). Spatially, the structured model has edges coinciding with body limbs and it additionally connects symmetric body parts (e.g., left wrist and right wrist, left knee and right knee) to alleviate image evidence double counting issues. Temporal edges connect the same body parts across two adjacent frames. However, our implementation of the inference layer is flexible and can perform approximate inference on arbitrary loopy graph configurations.

5.3. Evaluation Protocol

For consistent comparison with prior work on both the Penn Action dataset and the JHMDB dataset [11, 36, 19], we use a metric referred to as PCK, introduced in [38]. A candidate keypoint prediction is considered to be correct if it falls within $\alpha \cdot \max(h, w)$ pixels of the ground-truth keypoint, where h and w are the height and width of the bounding box of the instance in question, and α controls the relative threshold for considering correctness. We report results from different settings of α. We also report results that plot accuracy vs normalized distance from ground truth in pixels, where a joint is deemed correctly located if it is within a set distance of d pixels from a ground-truth joint center, where d is normalized by the size of the instance.

Method	Head	Shou	Elbo	Wris	Hip	Knee	Ankl	Mean
[19]	62.8	52.0	32.3	23.3	53.3	50.2	43.0	45.3
[36]	64.2	55.4	33.8	24.4	56.4	54.1	48.0	48.0
[12]	89.1	86.4	73.9	73.0	85.3	79.9	80.3	81.1
[11]	95.6	93.8	90.4	90.7	91.8	90.8	91.5	91.8
baseline	97.9	94.9	76.8	72.0	95.9	88.8	85.1	87.0
S-infer	98.0	90.3	85.2	86.7	93.7	93.5	93.6	91.4
ST-infer	98.0	97.3	95.1	94.7	97.1	97.1	96.9	96.5
ST-infer($\star$)	97.9	91.1	91.3	90.9	92.5	94.4	94.5	92.8
ST-infer($*$)	97.9	89.7	84.4	86.5	93.4	93.7	93.8	91.0
ST-infer(2)	97.6	96.8	95.2	95.1	97.0	96.8	96.9	96.4

Table 1. Comparison of PCK@0.2 on Penn Action dataset. We compare our proposed model with a baseline model, a baseline model with spatial inference and other state-of-the-art methods. We also investigate the performance of independent training ($\star$), the baseline ConvNet after end-to-end training ($*$) and temporal connection across 2 frames (2).

5.4. Analysis on Penn Action Dataset

Baseline comparison: Table 1 shows the relative performance on the Penn Action test set. For consistent comparison with previous work [36, 11, 19] the metric PCK@0.2 is used. This means a prediction is considered correct if it lies within $(\alpha = 0.2) \times \max(s_h, s_w)$. We first compare results from a baseline model, a spatial-only model and finally our spatio-temporal inference model. The baseline model corresponds to the pure fully ConvNet as described in Sec. 3.1 and is trained with loss Eq. (5). We also report the result after only applying spatial inference on top of the heat-maps obtained from the ConvNet, coresponding to only the blue arrows in Figure 2 (e). Please note that these two settings essentially treat video-based pose estimation as pure concatenation of single image predictions. Finally, we report the performance of our proposed end-to-end trainable network with full spatio-temporal inference.

Our baseline setting achieves 87.0% average accuracy for all 13 body parts. Spatial inference with geometric constraints among human body parts in individual images increases the overall result by 4.4%. By incorporating temporal consistency across frames, we observe an additional accuracy gain of 5.1% over spatial inference only.

Body parts like head and shoulders are usually visible and less flexible, so even with the baseline model very high detection accuracy can be achieved. However, parts such as elbows and wrists are the most flexible joints of our body. This flexibility can yield configurations with very large variation and these joints are also prone to be occluded by other parts of the body. This is shown by the low detection rates from the baseline model. With spatial message passing, the accuracy increases, and our proposed model boosts this again by roughly 10%. Note that predictions for shoulders can be negatively influenced by sending or receiving messages from elbows through spatial inference only. However, deploying temporal information helps in recovering from

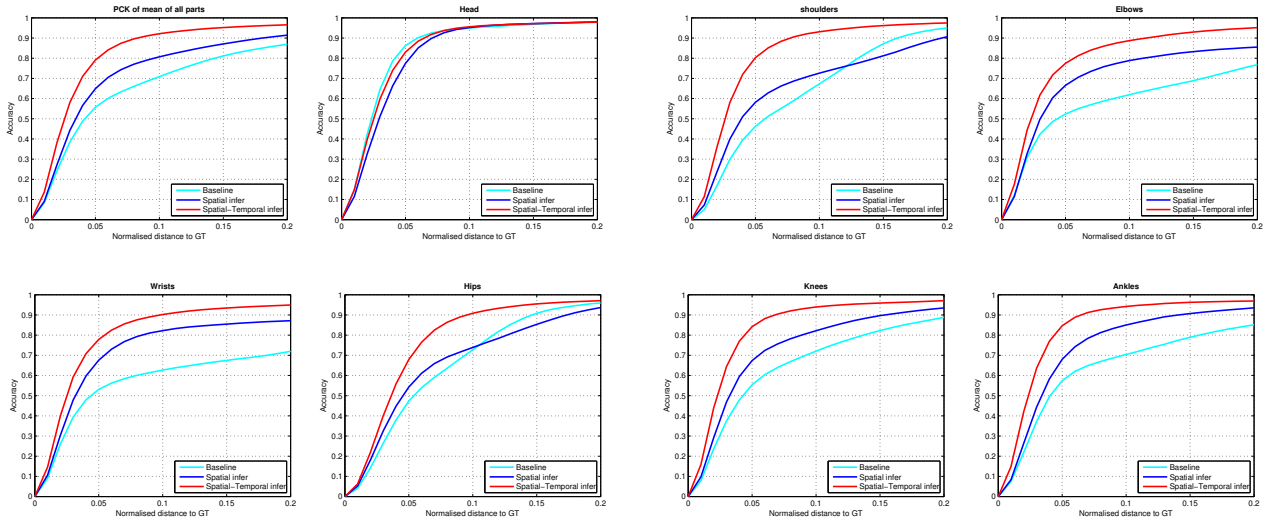

Figure 3. PCK curve for Penn Action dataset. We compare our proposed model with two baselines – ConvNet-only and spatial inference-only. Ours yields consistent accuracy improvements across the entire range of strictness.

such errors.

Analysis of normalized distance curves Figure 3 plots the normalized distance to the ground truth annotations. Generally, our proposed model outperforms the baseline model and the one with spatial inference over all levels of the evaluation and across all joints. Interestingly, even for stable (and hence easy to predict) joints like the head, we can still see improvements. In particular when the metric gets more strict (i.e., smaller d). In the cases of more flexible body parts such as elbows, wrists and knees, a constant improvement for both loose and strict metric can be observed. Especially over the 0.05 to 0.1 region, we can clearly observe more accurate predictions. This further suggests that back-propagating the error from several frames through our spatio-temporal network architecture benefits both unary and pairwise terms.

Further evaluations We also test the effectiveness of joint training of convolutional layers with message passing. Keeping the weights of convolutional layers fixed, we just train the parameters in the spatio-temporal inference layer. The overall performance is 92.8% (Table 1, row annotated by ($\star$)). It improves over the baseline model by 5.8% but could not reach the performance of joint training. The end-to-end training helps the fully convolutional layers to capture appearance features better. To validate this claim we conduct the same evaluation using the convolutional layers from the end-to-end trained model (removing the spatio-temporal inference layers) and compare the result with the baseline model (trained standalone). An overall 4% performance increase (Table 1, row annotated by ($*$)) can be observed. We also perform the experiment with temporal edges across not only 1 frame but 2 frames (Table 1, row annotated by (2)). However, here we do not observe a significant increase of mean accuracy.

Method	Head	Shou	Elbo	Wris	Hip	Knee	Ankl	Mean
baseline	93.2	72.4	57.3	61.9	88.4	63.6	48.6	70.9
S-infer	93.6	85.1	72.9	70.1	87.2	66.2	52.2	76.5
ST-infer	93.6	94.7	84.8	80.2	87.7	68.8	55.2	81.6
baseline($*$)	86.2	50.2	42.9	47.4	61.4	43.4	34.1	54.5
S-infer($*$)	86.1	62.8	55.2	51.9	68.3	48.1	36.7	60.2
ST-infer($*$)	85.4	77.6	69.4	62.6	76.9	57.4	42.9	68.7

Table 2. Results on full JHMDB dataset. The first three rows are based on PCK@0.2 while the results with ($*$) are with PCK@0.1.

Comparison with state-of-the-art Table 1 lists the comparison between the results of previous methods and ours. We first compare with shallow hand-crafted features based works [36, 19]. [19] is based on N-best algorithm and [36] employing different action specific models. We use the figures reported in [36] for comparison. We outperform them by a large margin for all body parts. [11] incorporates deep features with a recurrent structure to model long-term dependency between frames. While only propagating information over short periods of time (thin-slices of the sequence), we still attain an overall performance boost of 4.7% on this dataset. Please note that ours consistently localizes all joints better than prior work.

5.5. Analysis on JHMDB dataset

We also conduct a systematic evaluation on the JHMDB dataset [14]. The average result of three splits on this dataset is illustrated in Table 2. The first three rows summarize the performance under the PCK@0.2 metric. The same three models and settings as previously are evaluated and we observe results consistent with the experiments conducted on

Figure 4. Qualitative results on Penn Action dataset. We visualize connections among challenging limbs (arms and legs). Some failure cases are listed. Our method may miss limbs due to significant occlusions and heavy blur (last row).

the Penn Action dataset. The proposed end-to-end model boosts the overall performance by a relatively large margin. We also provide results for PCK@0.1 (Table 2, row marked with $*$). To consistently compare with other state-of-the-art

Method	Head	Shou	Elbo	Wris	Hip	Knee	Ankl	Mean
[19]	79.0	60.3	28.7	16.0	74.8	59.2	49.3	52.5
[36]	80.3	63.5	32.5	21.6	76.3	62.7	53.1	55.7
[12]	90.3	76.9	59.3	55.0	85.9	76.4	73.0	73.8
baseline	97.2	82.2	65.2	66.5	96.3	84.4	76.8	82.3
S-infer	97.0	87.3	74.9	71.1	97.5	89.4	86.0	86.9
ST-infer	97.1	95.7	87.5	81.6	98.0	92.7	89.8	92.1

Table 3. PCK@0.2 results on sub-JHMDB dataset. We compare with other previous methods and our own baselines.

results, we perform further experiments on a subset of the JHMDB dataset. These subsets remove sequences with incomplete bodies. The comparison is listed in Table 3. We outperform shallow feature based methods by a large margin [19, 36]. In [12], features are taken from the deep ConvNet and a graphical model based inference is conducted independently to refine the result. Our proposed method also provides better performance across all body parts.

5.6. Qualitative results

Figure 4 illustrates results from representative sequences taken from our experiments. Our method can capture articulated poses with strong pose changes across several frames. Cases with cluttered background, occlusion, and blur are included. Failure cases, shown in the bottom row of Figure 4, are often linked to extended periods of motion blur or occlusion across many frames. This hinders the ConvNet from capturing local appearance properties and impacts the estimation of dense optical flow. In these cases temporal inference over longer distances may be necessary.

6. Conclusion

We have proposed an end-to-end trainable network taking spatio-temporal consistency into consideration to estimate human poses in natural, unconstrained video sequence. We have experimentally shown that leveraging such a unified structured prediction approach outperforms multiple baselines and state-of-the art methods across datasets. Training regression layers jointly with the spatio-temporal inference layer benefits cases that display motion blur and occlusions but also improves predictions of unary terms due to the iterative back-propagation of errors. Interesting directions for future work include long-range temporal dependencies and handling of groups of people.

Acknowledgments

This work was partially supported by the ERC Starting Grant OptInt, the ERC Advanced Grant VarCity, and the Toyota Research Project TRACE-Zurich.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, pages 1014–1021, 2009.

[2] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, pages 1736–1744, 2014.

[3] A. Cherian, J. Mairal, K. Alahari, and C. Schmid. Mixing body-part sequences for human pose estimation. In *CVPR*, pages 2353–2360, 2014.

[4] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured feature learning for pose estimation. In *CVPR*, pages 4715–4723.

[5] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Human pose estimation using body parts dependent joint regressors. In *CVPR*, pages 3041–3048, 2013.

[6] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell University, 2004.

[7] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, pages 1–8, 2008.

[8] K. Fragkiadaki, H. Hu, and J. Shi. Pose from flow and flow from pose. In *CVPR*, pages 2059–2066, 2013.

[9] B. J. Frey and D. J. C. MacKay. A revolution: Belief propagation in graphs with cycles. In *NIPS*, pages 479–485, 1998.

[10] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *CVPR*, pages 437–446, 2015.

[11] G. Gkioxari, A. Toshev, and N. Jaitly. Chained predictions using convolutional neural networks. *arXiv preprint arXiv:1605.02346*, 2016.

[12] U. Iqbal, M. Garbade, and J. Gall. Pose for action-action for pose. *arXiv preprint arXiv:1603.04037*, 2016.

[13] A. Jain, J. Tompson, Y. LeCun, and C. Bregler. Modeep: A deep learning framework using motion features for human pose estimation. In *ACCV*, pages 302–315, 2014.

[14] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, pages 3192–3199, 2013.

[15] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, pages 1–11, 2010.

[16] M. W. Lee and R. Nevatia. Human pose tracking in monocular sequence using multilevel structured models. *IEEE Trans. Pattern Anal. Mach. Intell.*, (1):27–38, 2009.

[17] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499, 2016.

[18] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In *CVPR*, pages 2329–2336, 2014.

[19] D. Park and D. Ramanan. N-best maximal decoders for part models. In *ICCV*, pages 2627–2634, 2011.

[20] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *ICCV*, pages 1913–1921, 2015.

[21] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *CVPR*, pages 588–595, 2013.

[22] 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. *arXiv preprint arXiv:1511.06645*, 2015.

[23] D. Ramanan. Learning to parse images of articulated bodies. In *NIPS*, pages 1129–1136, 2006.

[24] D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *CVPR*, pages 271–278, 2005.

[25] X. Ren, A. C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *ICCV*, pages 824–831, 2005.

[26] B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *CVPR*, pages 1281–1288, 2011.

[27] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, pages 702–718, 2000.

[28] L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *CVPR*, pages 2041–2048, 2006.

[29] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *The International Journal of Robotics Research*, (6):371–391, 2003.

[30] M. Sun and S. Savarese. Articulated part-based model for joint object detection and pose estimation. In *ICCV*, pages 723–730, 2011.

[31] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, pages 1799–1807, 2014.

[32] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, pages 1653–1660, 2014.

[33] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, (2):283–298, 2008.

[34] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *ECCV*, pages 565–580, 2014.

[35] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, pages 4724–4732, 2016.

[36] B. Xiaohan Nie, C. Xiong, and S.-C. Zhu. Joint action recognition and pose estimation from video. In *CVPR*, pages 1293–1301, 2015.

[37] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, pages 3073–3082, 2016.

[38] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392, 2011.

[39] D. Zhang and M. Shah. Human pose estimation in videos. In *ICCV*, pages 2012–2020, 2015.

[40] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, pages 2248–2255, 2013.

[41] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, pages 1529–1537, 2015.

[42] S. Zuffi, J. Romero, C. Schmid, and M. J. Black. Estimating human pose with flowing puppets. In *CVPR*, pages 3312–3319, 2013.