Unsupervised Video Adaptation for Parsing Human Motion*

Haoquan Shen[1], Shoou-I Yu[2], Yi Yang[3], Deyu Meng[4], and Alexander Hauptmann[2]

[1] School of Computer Science, Zhejiang University, China
[2] School of Computer Science, Carnegie Mellon University, USA
[3] ITEE, The University of Queensland, Australia
[4] School of Mathematics and Statistics, Xi'an Jiaotong University, China
{shenhaoquan,yee.i.yang}@gmail.com, {iyu,alex}@cs.cmu.edu,
dymeng@mail.xjtu.edu.cn

Abstract. In this paper, we propose a method to parse human motion in unconstrained Internet videos without labeling any videos for training. We use the training samples from a public image pose dataset to avoid the tediousness of labeling video streams. There are two main problems confronted. First, the distribution of images and videos are different. Second, no temporal information is available in the training images. To smooth the inconsistency between the labeled images and unlabeled videos, our algorithm iteratively incorporates the pose knowledge harvested from the testing videos into the image pose detector via an adjust-and-refine method. During this process, continuity and tracking constraints are imposed to leverage the spatio-temporal information only available in videos. For our experiments, we have collected two datasets from YouTube and experiments show that our method achieves good performance for parsing human motions. Furthermore, we found that our method achieves better performance by using unlabeled video than adding more labeled pose images into the training set.

Keywords: Unsupervised Video Pose Estimation, Image to Video Adaptation, Unconstrained Internet Videos.

1 Introduction

[1]In this paper, we focus on articulated pose estimation in unconstrained Internet videos. While limited research efforts have been made to pose detection in videos [7,25,9,16], they only consider clean video data (*e.g.*, TV shows) rather than Internet videos which are much more noisy. Furthermore, the performance largely relies on the selection of training data and the accuracy may drop dramatically if the distributions of training and testing data are quite different. As such, the existing work have constrained the training and testing video to be similar. For example, in [7] and [25], researchers collected both the training and testing data from the TV shows "Friends" and "Lost". In that way, the scene, the person and apparel of both training and testing data are consistent. Pose detection in those videos is simplified.

* Electronic supplementary material - Supplementary material is available in the online version of this chapter at http://dx.doi.org/10.1007/978-3-319-10602-1_23. Videos can also be accessed at http://www.springerimages.com/videos/978-3-319-10601-4

[1] The code and datasets will be available upon request.

D. Fleet et al. (Eds.): ECCV 2014, Part V, LNCS 8693, pp. 347–360, 2014.

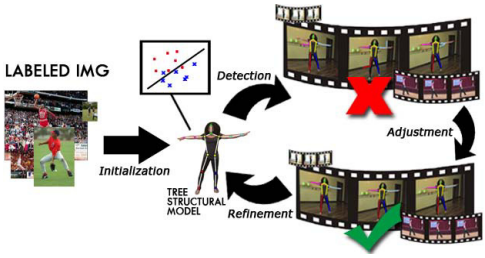


Fig. 1. The framework of our algorithm

The main unsolved challenge in current work is that Internet videos generally have huge apparel variations with different levels of occlusions and cluttered body parts. It is not reasonable to directly apply the model trained from clean TV shows to predict the poses in Internet videos, especially for articulated pose detection. To achieve reliable pose detection performance, it is necessary to have a large amount of training videos covering a variety of apparels, backgrounds (scenes) and poses. Yet it is very time consuming to label the poses and computationally intensive to train the models as a single video clip contains many frames in real cases. Compared to videos, it is much easier to label still images without the tediousness of reviewing the video streams. For example, the effort of labeling 100 images is much less than that of labeling 100 video clips. In addition, there are some image datasets, *e.g.*, PARSE [17], with labeled body parts that contain a variety of articulated poses. In this paper, we propose to leverage such free data to estimate poses in Internet quality videos. To the best of our knowledge, this is the first work on articulated pose detection without any labeled videos. The merit of our algorithms is that no human supervision is required.

As shown in Fig. 1, our algorithm starts with training an image pose estimator using an external pre-labeled image dataset. These pose estimators can be used as good initializations for Internet videos, since labeled images have relatively larger variations although less than Internet videos. We then propose a self-refining approach to adapt the pose knowledge from the testing videos, and incorporate the information into the next round of learning, during which both spatial and temporal constraints are utilized. More specifically, we first apply a self-adjustment approach to the results of image pose detection by tracking the trajectory of each body part across multiple frames with spatial smoothing constraints. Then, we use a scoring strategy to pick the frames with high confidence in the testing data, while preserving the diversity of selected key-frames and adding them as extra labeled poses to the training process.

Our contributions are summarized as follows:

1. We address the limitations of previous work, which are unable to deal with Internet videos with large variations and heavy clutters. We propose a self-refining approach to adapt the pose knowledge from static images to Internet videos.
2. We introduce a self-adjustment method to improve accuracy by tracking the trajectory of each body part across multiple frames with spatial smoothing constraints.
3. We collect a challenging pose detection dataset consisting of full-body and half-body dancing clips from Internet videos, which have large variations in terms of scene, person, apparel, *etc*.

2 Related Work

Pose detection is a very valuable but tough task in computer vision. Researchers have addressed the problem of pose detection in video dating back to the classic model-based approaches [15,8,21]. The difficulties are summarized as follows:

1. Huge variations of human poses on Internet videos, as depicted in Fig. 8 and Fig. 9: For example, human limbs are stretched and foreshortened. Left and right limbs reverse regularly due to rotation and self-occlusion. Appearances including skin color, clothing, body shape differ from one person to another. In some scenarios, multiple persons are seen simultaneously and occlude each other.
2. Poor quality of most Internet videos: Uploaded videos often have low resolution and serious motion blur.
3. Lack of generalizability. In fact, most of the existing methods are training-data-driven: When we detect poses in Internet videos which consist of more varied body shapes, apparel, backgrounds, *etc*, existing models generally cannot adapt well to the new domain.

Recent work has examined this problem for static images, assuming that techniques for static images will be needed in video-based articulated trackers. Other than the techniques exploring the tradeoff between generative and discriminative models from an overall perspective [11,28,23], multiple approaches advocate strong body models. The graph-based and tree-structured models are the two main approaches for this task. Loopy models [20,13,26,1,30] (graph-based models) have stricter constraints of different body parts and usually lead to good performance. But they are also harder to optimize and more time consuming. Other approaches are tree models, which allow for efficient inference, but are often plagued by the well-known phenomena of double counting [3,19,22]. In addition, researchers [24] also extend the single model to multiple model scenario and use model selection to improve performance. Recently, a novel tree structured framework [32,31] has received much attention. It extends the classic pictorial structure [3,6] and parameterizes body parts by both pixel locations and latent variable "orientations". This model realizes a good balance between performance and efficiency, which achieves state-of-the-art performance for static images and can be efficiently implemented when Structural SVM [5] and Dynamic Programming are applied.

There are also some research efforts to pose detection in video streams [7,25,9,16]. For example, a segmentation-based pose and flow framework is proposed in [7], which is similar to [9,10]. In [14], Ma et. al. proposed an algorithm to adapt the knowledge from clean lab-generated videos for action recognition in the real world videos, e.g., the YouTube videos. In [25], researchers approximate the full, intractable spatio-temporal loopy model of pose detection by decomposing it into an ensemble of tree models. Other pose detectors use labeled videos as training data and train pose detectors by applying both spatial and temporal constraints [2,16]. Several recent papers [27,12,18,4], enhance performance by using tracking methods. However, all of these use video data from TV shows or lab recorded videos which are cleaner than Internet video data. Furthermore, the performance largely relies on the selection of training data and the accuracy may drop dramatically if the distributions of training and testing data are quite

different. To handle Internet videos well, in this paper, we propose a self-refining approach to uncover the pose knowledge of Internet videos.

3 Framework

The framework of our method is shown in Fig. 1. Specifically, we first initialize our model with a small number of labeled images. Then, we apply a self-refining approach to adapt the pose knowledge to the testing videos. This approach can be summarized as:

1. Detect human pose on every frame of the test videos using [32,31], which is a state-of-the-art image pose detector (Section 3.1).
2. Adjust pose detection results by using continuity and tracking constraints for the testing videos (Section 3.2).
3. Gradually add high confidence frames automatically found in the testing videos to the training set for the next round of learning (Section 3.3). Repeat step 1.

In the following sections, we introduce our three main implementation procedures in detail.

3.1 Pose Detection

For each iteration in the self-refining process, we first generate an image pose detector. In the initialization stage, only labeled images are used as training data. After that, additional high-confidence frames in the testing videos are automatically selected for use in training. Here we follow the tree-structured model [32,31] and write the score function of a candidate pose as follows:

$$\begin{aligned} \max_{p,t} & \sum_{i \in vertex} b_i^{t_i} + \sum_{ij \in edge} b_{ij}^{t_i,t_j} \\ & + \sum_{i \in vertex} w_i^{t_i} \cdot \phi(f, p_i) \\ & + \sum_{ij \in edge} w_{ij}^{t_i,t_j} \cdot \psi(p_i - p_j) \end{aligned} \tag{1}$$

In Eq. (1), $vertex$ and $edge$ are the nodes and edges of the pose tree. p_i, t_i stand for the pixel location and orientation of part i. The parameter $b_i^{t_i}$ favors a particular type of assignment for part i, while the pairwise parameter $b_{ij}^{t_i,t_j}$ favors particular co-occurrences of part types. $\phi(f, p_i)$ is the feature vector extracted from p_i. The third term can be viewed as the loss when part i is placed at location p_i with the orientation t_i. w^{t_i} is a template learned from Structural SVM by taking orientation t_i as a latent variable [32,31]. The last term stands for the loss of a "switching" spring which is the dot product of spring parameter $w_{ij}^{t_i,t_j}$ and pixel difference of parts. Following [32,31], we solve Eq. (1) by using Dynamic Programming (DP).

3.2 Pose Adjustment

Continuity Constraint. One important property of pose detection on videos is that the positions of human joints in consecutive frames will not change dramatically. We call

this property Continuity Constraint. In this step, we adjust the pose detection results by using this continuity property. We denote V and f as video and frame. $next(f)$ is the frame after f. $vertex$ is the nodes of the tree models. p_i^f and $\widetilde{p}_i^f$ are the location for part i in frame f before and after the adjustment process. Our adjustment process can be converted into optimizing the following objective:

$$\min_{\widetilde{p}} \sum_{f \in V} \sum_{i \in vertex} \left(||\widetilde{p}_i^f - p_i^f||_2^2 + \alpha ||\widetilde{p}_i^{next(f)} - \widetilde{p}_i^f||_2^2 \right) \tag{2}$$

In Eq. (2), the first term restricts the adjusted results to be similar to the original ones. The second term is the temporal constraint that joints in adjacent frames won't change much. α parameterizes the weight of the continuity constraint. By doing this, our insight is that wrong results will cause a big discontinuity to adjacent frames with high confidence score, resulting in a big loss to the second temporal term, which can be reduced in the optimization step. Fig. 2 shows two examples of pose adjustment using the continuity constraint, from which we can see a visible improvement after the adjustment.

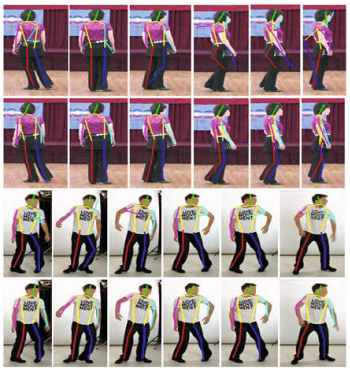


Fig. 2. Pose detection results before (rows 1 & 3) and after (rows 2 & 4) adjustment with continuity constraint

Tracking Constraint. A tracking rectification algorithm, which tracks each body part across multiple frames, is used to rectify incorrect body parts. Given the pose detection results for a source frame f, one could track each body part forward or backward in time and produce hypotheses of part locations for neighboring frames. Inversely, neighboring frames of f will also produce body part location hypotheses for frame f. As shown in Fig. 4, to rectify the pose detection results for frame f, we perform a weighted fusion of all the hypotheses provided by the neighboring frames. The weight of each hypothesis is determined by the pose detection score in the source frame. Our insight is that high-scoring poses have more accurate predictions of body part locations, thus making the hypotheses generated by these detections also more reliable.

Specifically, as shown in Fig. 3, suppose that we want to use the results of the frame f to generate the tracking results after 25 frames (1 second). Taking the right wrist as

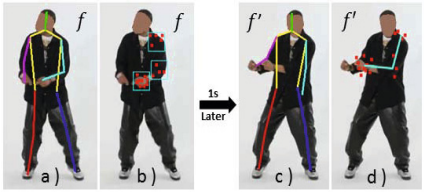


Fig. 3. Pose refinement using tracking cues. Frame f' is the frame 1 second after frame f. a) Pose detection results of frame f according to Eq. (1). b) Trajectory keypoints of the right arm in frame f. c) Wrong pose detection results of frame f' according to Eq. (1). d) Refined pose according to the arm trajectory key point of frame f'.

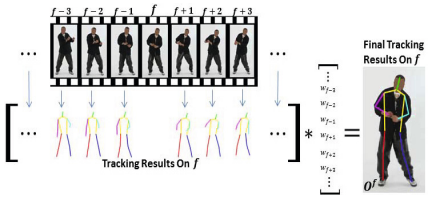


Fig. 4. Procedure to generate tracking results

an example, firstly, we can cover every joint with a box. Then, we detect all the dense trajectory keypoints in each box and track these trajectory keypoints by using [29]. Finally, the prediction results are generated by averaging the tracking points. Similarly, we can apply this method to other parts to generate the tracking results for the full human skeleton. In fact, the tracking results can be very good in practice as shown in Fig. 3.

We denote the fused tracking results of the i-th part in f as O_i^f and rewrite the Eq. (2) as the following:

$$\begin{aligned} \min_{\widetilde{p}} \sum_{f \in V} \sum_{i \in vertex} \Big(& ||\widetilde{p}_i^f - p_i^f||_2^2 \\ & + \alpha ||\widetilde{p}_i^{next(f)} - \widetilde{p}_i^f||_2^2 + \beta ||\widetilde{p}_i^f - O_i^f||_2^2 \Big) \end{aligned} \tag{3}$$

In Eq. (3), other than what we have discussed before, the last term restricts the adjusted results to be similar with the tracking results. β parameterizes the degree that we trust the tracking results. As shown in Fig. 3, there often exists a situation where the pose detection results are wrong but the tracking results are correct. This will cause a big tracking error, which can be optimized by balancing the tracking error with other constraints.

Since (3) is a convex optimization problem, we can calculate the derivative for every variable $\widetilde{p}_i$ and solve it using an iterative method by setting the derivative to be zero.

3.3 Pose Detector Refinement

In the refinement process, we automatically select the frames with top scores in the testing videos and use them as extra training data in the next round of learning. Denote $\mathcal{R}$

as the pose detection results. S_S and S_T are the pose detection and the pose adjustment scores. We define the score of results $\mathcal{R}$ on frame f as follows:

$$S(f, \mathcal{R}) = S_S(f, \mathcal{R}) + S_T(f, \mathcal{R}) \tag{4}$$

Where, similar to Eq. (1) and (3), we write the spatial and temporal scores as follows:

$$\begin{aligned} S_S(f, \mathcal{R}) = & \textstyle\sum_{i \in vertex} b_i^{t_i} + \sum_{ij \in edge} b_{ij}^{t_i, t_j} \\ & + \textstyle\sum_{i \in vertex} w_i^{t_i} \cdot \phi(f, \widetilde{p}_i) \\ & + \textstyle\sum_{ij \in edge} w_{ij}^{t_i, t_j} \cdot \psi(\widetilde{p}_i - \widetilde{p}_j) \end{aligned} \tag{5}$$

$$\begin{aligned} S_T(f, \mathcal{R}) = & -\gamma \textstyle\sum_{i \in vertex} \Big(||\widetilde{p}_i^{next(f)} - \widetilde{p}_i^f||_2^2 + \\ & ||\widetilde{p}_i^{prev(f)} - \widetilde{p}_i^f||_2^2 \Big) - \theta \textstyle\sum_{i \in vertex} ||\widetilde{p}_i^f - O_i^f||_2^2 \end{aligned} \tag{6}$$

In Eq. (5), the spatial score S_S reflects both the confidence of every body part and the matching rate of every adjacent body part. In Eq. (6), the temporal score S_T is the negative loss in the adjustment procedure, in which the first and second terms stand for the location differences of every body part to adjacent two frames, and the third term stands for the error compared to the tracking results. Here, γ and θ parameterize the degree of punishment on frame discontinuity and tracking mismatch.

Note that even though Eq. (5) and (6) have the similar forms as Eq. (1) and (3), their purposes are different. For Eq. (1) and (3), they are used during optimization, whereas Eq. (5) and (6) are only used to compute scores. No optimization is done using Eq. (5) ans (6).

In our refinement process, to keep both the quality and diversity of added testing key-frames, we only select the frames which have scores above 0.4 in Eq. (4) and we select at most 4 frames from each video per iteration.

4 Experiments

Datasets: We have constructed one full-body and one upper-body dataset for testing purposes from the dancing videos of Youtube, which we call Full-body Youtube Dancing Pose (FYDP) dataset and Upper-body Youtube Dancing Pose (UYDP) dataset, respectively. Each of the FYDP and UYDP dataset contains 20 video clips. Each video clip lasts around 4 seconds and consists of around 100 consecutive annotated video frames. Specifically, our FYDP dataset contains dancing videos with fast and slow movements, rotating and split-leg positions, stretched and forshortened limbs. In the UYDP dataset, more intricate upper-body motions are included. Some typical frames in FYDP and UYDP are depicted in Fig. 8 and Fig.9, respectively. In addition to FYDP and UYDP, we also used the VideoPose2 dataset collected in [25] to evaluate the performance of the proposed algorithm.

In our experiments, the labeled images for initialization are selected from the PARSE dataset [17] and the BUFFY dataset [4], respectively. PARSE dataset contains 305 pose-annotated images of highly-articulated full body images of human poses, and the BUFFY dataset contains 748 upper-body-annotated images extracted from 5 episodes

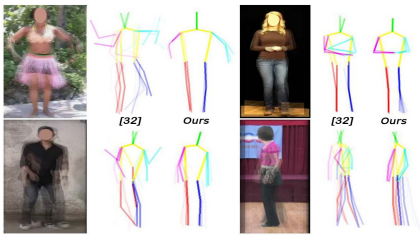


Fig. 5. A comparison between our model and that of [32]. This figure shows the aggregated frames sampled from dancing video clips of two seconds and the pose skeletons obtained by our model and [32], respectively.

of a TV show. Both datasets have specified the training and testing sets [32,31]. In each of our full-body and upper-body experiments, we respectively use three settings of PARSE and BUFFY images as initialization to test how sensitive our method is to the number of labeled data: half of the images from the training set, all the images from the training set, all images of the training and testing sets.

To train our image pose detector, we follow the experiment settings in [32,31] and use the negative training images from the INRIAPerson database [33]. These images tend to be outdoor scenes which do not contain people.

Evaluation Criteria. Following [32], in which researchers have discussed the limitations of PCP (Probability of a Correct Pose) [4], we use APK (Average Precision of Keypoints) and PCK (Percent of Correct Keypoints) [32] in our experiments with the threshold to be 0.1 for FYDP dataset and 0.2 for UYDP and VideoPose2 [25] datasets. When the bounding boxes of every person is given, PCK evaluates the percentage of correct keypoints. For comparison, APK is stricter, in that both missed-detections and false-positives are penalized.

Structure. We use 26 parts and 18 parts tree-structured models in our full-body and upper-body experiments, respectively, in which both joint positions and some midway points between limbs are included. For each part, we use 4(8) mixtures for full-body(upper-body) detector, which has shown to be a good tradeoff between performance and efficiency in [32].

Parameters. In our experiments, we iterate 3 times to adapt the domain knowledge of testing videos, which is demonstrated by our experiments to be a good balance between efficiency and performance. In our refining process, there are two parameters γ and θ. We empirically set $\gamma = 0.5, \theta = 1$, for which our method can consistently perform well. For α and β in Eq. (3), our experiments verify that the proposed framework is not sensitive to both parameters. We empirically set $\alpha = 5, \beta = 1$.

Table 1. Full-body pose detection results on FYDP dataset, when different number of training images from PARSE dataset are used

50% randomly sampled images from the training set of PARSE are used for training

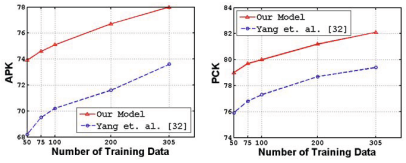
Criteria	Method	Head	Shou	Elbo	Wris	Hip	Knee	Ankle	Total
APK	Yang et. al. [32]	92.9	85.7	57.0	27.1	73.2	70.2	71.3	68.2
	Our Model	**95.0**	**88.5**	**65.2**	**32.3**	**78.1**	**78.2**	**79.8**	**73.9**
PCK	Yang et. al. [32]	94.4	89.3	69.9	48.7	80.8	75.0	73.4	75.9
	Our Model	**95.5**	**91.2**	**74.6**	**50.8**	**82.4**	**81.6**	**77.2**	**79.0**

All images from the training set of PARSE are used for training

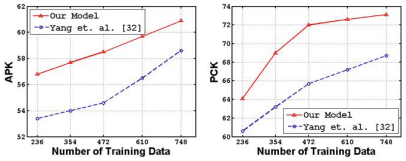
Criteria	Method	Head	Shou	Elbo	Wris	Hip	Knee	Ankle	Total
APK	Yang et. al. [32]	94.4	86.7	58.2	33.3	68.6	73.4	76.4	70.2
	Our Model	**95.9**	**88.8**	**67.3**	**35.4**	**79.7**	**79.3**	**79.3**	**75.1**
PCK	Yang et. al. [32]	95.2	89.9	69.0	53.1	78.1	78.2	77.7	77.3
	Our Model	**96.1**	**91.0**	**74.8**	**54.5**	**83.6**	**81.7**	**78.8**	**80.0**

All images of PARSE are used for training

Criteria	Method	Head	Shou	Elbo	Wris	Hip	Knee	Ankle	Total
APK	Yang et. al. [32]	95.3	86.9	66.2	41.1	73.8	75.2	76.8	73.6
	Our Model	**95.7**	**89.6**	**73.6**	**43.5**	**82.4**	**82.9**	**78.2**	**78.0**
PCK	Yang et. al. [32]	95.8	89.9	73.7	58.5	80.1	79.8	78.1	79.4
	Our Model	**96.2**	**91.7**	**78.4**	**60.3**	**85.4**	**83.8**	**79.2**	**82.1**



(a) Full-body pose detection results on FYDP dataset, when different number of training images from PARSE dataset are used



(b) Upper-body pose detection results on UYDP dataset, when different number of training images from BUFFY dataset are used

Fig. 6. Full-body and upper-body pose detection results

Compared Algorithms. In the experiments, we compare our method to [32,24], which are state-of-the-art pose detectors on static images. As the algorithm proposed in [24] is only able to detect three body parts, *i.e.* shoulder, elbow and wrist, we do not report the results of [24] on the full-body dataset FYDP. Note that we aim to parse human motion without labeling any videos for training. We are unable to compare our algorithm to [7,25] because both [7] and [25] require labeled *video clips* for training, which are unavailable in our experiments. Other than [32], which achieves both state-of-the-art results and high time efficiency, we could extend any image pose detector to the video scenario by simply replacing the pose detection process.

Table 2. Upper-body pose detection results on UYDP dataset, when different number of training images from BUFFY dataset are used

50% randomly sampled images from the training set of BUFFY are used for training

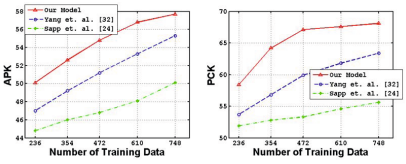
Criteria	Method	Head	Shou	Elbo	Wris	Hip	S. E. W. Avg	All Avg
APK	Sapp et. al. [24]	NA	67.1	**32.0**	35.3	NA	44.8	NA
	Yang et. al. [32]	84.2	74.2	22.5	44.3	41.9	47.0	53.4
	Our Model	**88.3**	**80.1**	22.6	**47.5**	**45.5**	**50.1**	**56.8**
PCK	Sapp et. al. [24]	NA	81.3	38.8	35.6	NA	51.9	NA
	Yang et. al. [32]	88.1	79.5	36.4	45.1	54.1	53.7	60.6
	Our Model	**89.8**	**87.4**	**38.9**	**48.8**	**55.7**	**58.4**	**64.1**

All images from the training set of BUFFY are used for training

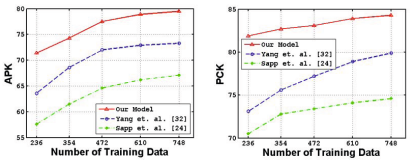
Criteria	Method	Head	Shou	Elbo	Wris	Hip	S. E. W. Avg	All Avg
APK	Sapp et. al. [24]	NA	69.5	33.7	37.3	NA	46.8	NA
	Yang et. al. [32]	85.0	78.2	29.2	46.2	34.4	51.2	54.6
	Our Model	**90.9**	**83.5**	**33.3**	**47.7**	**36.9**	**54.8**	**58.5**
PCK	Sapp et. al. [24]	NA	82.2	39.6	38.1	NA	53.3	NA
	Yang et. al. [32]	90.9	84.9	43.6	51.4	57.7	59.9	65.7
	Our Model	**97.5**	**95.6**	**49.0**	**56.6**	**61.5**	**67.1**	**72.0**

All images of BUFFY are used for training

Criteria	Method	Head	Shou	Elbo	Wris	Hip	S. E. W. Avg	All Avg
APK	Sapp et. al. [24]	NA	72.0	**38.6**	39.6	NA	50.1	NA
	Yang et. al. [32]	88.0	81.3	33.9	50.7	39.1	55.3	58.6
	Our Model	**91.6**	**84.8**	37.2	**51.0**	**39.8**	**57.7**	**60.9**
PCK	Sapp et. al. [24]	NA	83.5	42.4	40.8	NA	55.6	NA
	Yang et. al. [32]	92.5	86.5	49.4	54.2	60.8	63.4	68.7
	Our Model	**97.7**	**94.8**	**53.8**	**55.6**	**63.4**	**68.1**	**73.1**



(a) Pose detection results on UYDP dataset, when different number of training images from BUFFY dataset are used.



(b) Pose detection results on VideoPose2 dataset, when different number of training images from BUFFY dataset are used.

Fig. 7. Three body parts (shoulder, elbow, wrist) pose detection results

Experimental Settings. In this paper, we have done three experiments separately to demonstrate the effectiveness of our proposed framework. In order to see whether our method is sensitive to the number of labeled training data, for each experiment, we show the results under three different settings based on the number of labeled training data to do initialization: half images of the training set (for either BUFFY or PARSE) are used for training, all images of the training set are used for training, all images of the training and testing set are used for training. The three experiments are as follows:

Table 3. Upper-body pose detection results on VideoPose2 dataset, when different number of training images from BUFFY dataset are used

50% randomly sampled images from the training set of BUFFY are used for training

Criteria	Method	Head	Shou	Elbo	Wris	Hip	S. E. W. Avg	All Avg
APK	Sapp et. al. [24]	NA	81.0	42.4	49.3	NA	57.6	NA
	Yang et. al. [32]	**92.3**	90.1	40.1	60.5	46.5	63.6	65.9
	Our Model	**92.3**	**92.5**	**58.3**	**63.5**	**49.6**	**71.4**	**71.2**
PCK	Sapp et. al. [24]	NA	90.8	61.3	59.3	NA	70.5	NA
	Yang et. al. [32]	96.1	94.7	56.6	67.9	71.3	73.1	77.3
	Our Model	**96.6**	**97.2**	**78.6**	**69.8**	**74.5**	**81.9**	**83.3**

All images from the training set of BUFFY are used for training

Criteria	Method	Head	Shou	Elbo	Wris	Hip	S. E. W. Avg	All Avg
APK	Sapp et. al. [24]	NA	84.6	57.1	52.0	NA	64.6	NA
	Yang et. al. [32]	**97.0**	94.4	56.4	65.2	51.3	72.0	72.9
	Our Model	96.7	**95.8**	**68.2**	**68.4**	**60.7**	**77.5**	**78.0**
PCK	Sapp et. al. [24]	NA	92.5	66.2	61.5	NA	73.4	NA
	Yang et. al. [32]	**97.2**	94.6	67.0	70.1	74.0	77.2	80.6
	Our Model	96.9	**97.5**	**78.4**	**73.3**	**77.8**	**83.1**	**84.8**

All images of BUFFY are used for training

Criteria	Method	Head	Shou	Elbo	Wris	Hip	S. E. W. Avg	All Avg
APK	Sapp et. al. [24]	NA	85.0	64.2	52.2	NA	67.1	NA
	Yang et. al. [32]	95.3	95.2	64.1	60.5	52.3	73.3	73.5
	Our Model	**96.6**	**95.9**	**74.5**	**68.2**	**61.7**	**79.5**	**79.4**
PCK	Sapp et. al. [24]	NA	93.3	69.3	61.3	NA	74.6	NA
	Yang et. al. [32]	97.0	96.1	73.4	70.2	71.0	79.9	81.5
	Our Model	**97.3**	**97.1**	**82.4**	**73.4**	**77.5**	**84.3**	**85.5**

1. We compare our full-body model to Yang et. al. [32] on FYDP by utilizing PARSE [17] to do initialization as shown in Table 1 and Fig. 6 (a).
2. We compare our upper-body model to Yang et. al. [32] on UYDP by utilizing BUFFY [4] to do initialization as shown in Table 2, Fig. 6 (b) and Fig. 7 (a).
3. We compare our upper-body model to Yang et. al. [32] and Sapp et. al. [24] on VideoPose2 dataset [25] by utilizing BUFFY [4] to do initialization as shown in Table 3 and Fig. 7 (b). We do not compare to video based method [7,25] since video data are not available in the training process.

Experiment Results. From Table 1, Table 2 and Table 3, we can see that our method achieves a significant improvement compared to the image pose detector. In addition, if we look at the results in detail, in Table 1, when we use half of the training set (50 images) to do training, we can obtain 73.9% APK and 79.0% PCK. If we instead trained a state-of-the-art image pose detector [32] with 305 images, it only achieves 73.6% APK and 79.4% PCK. This shows that our method, with only $\frac{1}{6}$ training data, can still generate comparable results to the state-of-the-art image pose detector [32].

Furthermore, from Fig. 6, we observe that: 1) when the number of labeled training images increases, both the performance of our method and [32] are improved. 2) our

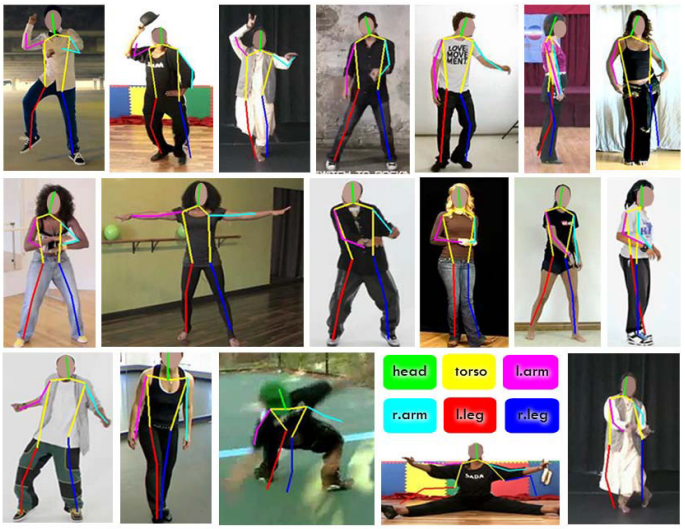


Fig. 8. Key frame results on FYDP dataset. We show different parts of full-body skeleton using 6 different colors. Other than the last three images of the last row, all images show successful examples. By examining the failure cases, we find our model is still confused by foreshortened limbs, horizontal people and the left/right limb.

Fig. 9. Key frame results on UYDP dataset. We show different parts of half-body skeleton using 4 different colors. Other than the last three images of the last row, all the images show successful examples. We see our model still has difficulty to hidden parts, foreshortened limbs and self-occlusion.

model is always better than [32] when the number of training data varies. 3) our model is not very sensitive to the number of training data, and can generally get pretty good results when only 50 images are used for training.

To vividly compare our pose detection results to a state-of-the-art image-based pose detector [32], we visualize the human motion parsing results of a two-second video clip, which are shown in Fig. 5. The comparison shows that our model is more robust to noise and can clearly represent the movements. In addition, we also show some successful and failed examples of our full-body and upper-body results in Fig. 8 and Fig. 9.

5 Conclusion

We propose an unsupervised framework to adapt pose detector from images to unconstrained Internet videos. A novel adjustment strategy is proposed to iteratively exploit the domain specific information in unconstrained videos, where no labeled videos are available. Temporal smoothness and body part consistency are simultaneously satisfied to refine the pose detection model, which is initialized only by labeled images. The merit of our work is that the pre-trained model does not have to fit the testing data, which are unseen during initialization. Therefore, no human supervision is required when we adapt the image model to videos. Our framework is a general one, which can be readily extended to any other image pose detector for Internet videos. We demonstrate the effectiveness and robustness of our framework through the full-body and upper-body pose experiments based on a real world Internet video set. One limitation of the proposed algorithm is that if the video resolution is low, the tracking results may not be robust enough. In these cases, the improvement from tracking part will decrease. In the future, we will improve the tracking method.

Acknowledgments. This paper was partially supported by the US Department of Defense the U. S. Army Research Office (W911NF-13-1-0277), partially supported by the National Science Foundation under Grant Number IIS-12511827, partially supported by the ARC DECRA project (DE130101311), the UQ ECR project (2013002401) and the NSFC projects with No.61373114. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DoI/NBC, ARO, NSF, or the U.S. Government.

References

1. Bergtholdt, M., Kappes, J.: A study of parts-based object class detection using complete graphs. In: IJCV (2009)
2. Fablet, R., Black, M.J.: Automatic detection and tracking of human motion with a view-based representation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 476–491. Springer, Heidelberg (2002)
3. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. IJCV 61(1), 55–79 (2005)
4. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Progressive search space reduction for human pose estimation. In: CVPR (2008)
5. Finley, T., Joachims, T.: Training structural svms when exact inference is intractable. In: ICML (2008)
6. Fischler, M., Elschlager, R.: The representation and matching of pictorial structures, vol. 100, pp. 67–92 (1973)
7. Fragkiadaki, K., Hu, H., Shi, J.: Pose from flow and flow from pose. In: CVPR (2013)
8. Hogg, D.: Model-based vision: a program to see a walking person. Image and Vision computing 1(1), 5–20 (1983)
9. Jiang, H.: Human pose estimation using consistent maxcovering. In: ICCV (2009)
10. Ju, S.X., Black, M.J., Yacoob, Y.: Cardboard people: A parameterized model of articulated image motion. In: FG (1996)

11. Kumar, M., Zisserman, A., Torr, P.: Efficient discriminative learning of parts-based models. In: CVPR (2010)
12. Lan, X., Huttenlocher, D.: Beyond trees: Common-factor models for 2d human pose recovery. In: ICCV (2005)
13. Lee, M., Cohen, I.: Proposal maps driven mcmc for estimating human body pose in static images. In: CVPR (2004)
14. Ma, Z., Yang, Y., Nie, F., Sebe, N., Yan, S., Hauptmann, A.: Harnessing lab knowledge for real-world action recognition. International Journal of Computer Vision 109(1-2), 60–73 (2014)
15. O'Rourke, J., Badler, N.: Model-based image analysis of human motion using constraint propagation. PAMI 2(6), 522–536 (1980)
16. O'Rourke, J., Badler, N.: 2d human pose estimation in tv shows. Statistical and Geometrical Approaches to Visual Motion Analysis 1, 128–147 (2009)
17. Ramanan, D.: Learning to parse images of articulated bodies. In: NIPS (2007)
18. Ramanan, D., Forsyth, D., Zisserman, A.: Strike a pose: Tracking people by finding stylized poses. In: CVPR (2005)
19. Ramanan, D., Sminchisescu, C.: Training deformable models for localization. In: CVPR (2006)
20. Ren, X., Berg, A.C., Malik, J.: Recovering human body configurations using pairwise constraints between parts. In: ICCV (2005)
21. Rohr, K.: Towards model-based recognition of human movements in image sequences. CVGIP-Image Understanding 59(1), 94–115 (1994)
22. Ronfard, R., Schmid, C., Triggs, B.: Learning to parse pictures of people. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part IV. LNCS, vol. 2353, pp. 700–714. Springer, Heidelberg (2002)
23. Sapp, B., Jordan, C., Taskar, B.: Adaptive pose priors for pictorial structures. In: CVPR (2010)
24. Sapp, B., Taskar, B.: Modec: Multimodal decomposable models for human pose estimation. In: CVPR (2013)
25. Sapp, B., Weiss, D., Taskar, B.: Parsing human motion with stretchable models. In: CVPR (2011)
26. Sigal, L., Black, M.: Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In: CVPR (2006)
27. Sigal, L., Isard, M., Sigelman, B.H., Black, M.J.: Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In: NIPS (2003)
28. Singh, V.K., Nevatia, R., Huang, C.: Efficient inference with multiple heterogeneous part detectors for human pose estimation. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 314–327. Springer, Heidelberg (2010)
29. Wang, H., Kläser, A., Schmid, C., Liu, C.: Action Recognition by Dense Trajectories. In: IEEE Conference on Computer Vision & Pattern Recognition, Colorado Springs, United States, pp. 3169–3176 (June 2011), http://hal.inria.fr/inria-00583818/en
30. Wang, Y., Mori, G.: Multiple tree models for occlusion and spatial constraints in human pose estimation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 710–724. Springer, Heidelberg (2008)
31. Yang, Y., Ramanan, D.: Articulated pose estimation using flexible mixtures of parts. In: CVPR (2011)
32. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. PAMI 61(1), 55–79 (2013)
33. Yuille, A., Rangarajan, A.: The concave-convex procedure. Neural Computation 15(4), 915–936 (2003)