Joint Action Recognition and Pose Estimation From Video

Bruce Xiaohan Nie, Caiming Xiong and Song-Chun Zhu
Center for Vision, Cognition, Learning and Art
University of California, Los Angeles, USA
{niexh,caimingxiong}@ucla.edu, sczhu@stat.ucla.edu

Abstract

Action recognition and pose estimation from video are closely related tasks for understanding human motion, most methods, however, learn separate models and combine them sequentially. In this paper, we propose a framework to integrate training and testing of the two tasks. A spatial-temporal And-Or graph model is introduced to represent action at three scales. Specifically the action is decomposed into poses which are further divided to mid-level ST-parts and then parts. The hierarchical structure of our model captures the geometric and appearance variations of pose at each frame and lateral connections between ST-parts at adjacent frames capture the action-specific motion information. The model parameters for three scales are learned discriminatively, and action labels and poses are efficiently inferred by dynamic programming. Experiments demonstrate that our approach achieves state-of-art accuracy in action recognition while also improving pose estimation.

1. Introduction

1.1. Motivation and Objective

Action recognition and pose estimation are both important tasks for vision-based human motion understanding. They are widely used in applications, such as, intelligent surveillance systems and human-computer interaction systems. Despite their different goals, the two tasks are highly coupled and it is desirable to study them in a common framework. However, existing methods train models for the two tasks separately and combine the inference sequentially: taking pose estimation as input for action recognition[15, 34, 25, 8, 27, 24]. For certain actions defined by specific geometric configuration of body parts, pose estimation from a single image may be sufficient for action recognition [15, 28, 5, 32].

The main drawback of such methods is that the accuracy of action recognition highly relies on the obtained pose estimations. Due to the large pose variation and complex background in action datasets, the most discriminative parts (such as 'arms', 'hands', 'legs' and 'feet') are often missed in pose estimation, thereby deteriorating subsequent action recognition. However, those human parts have large motion in actions and can be recovered by motion information. For example, Fig. 1 shows that the arms and legs mis-detected by a pose estimation method [29] are successfully detected by our method. Besides the motion information on arms and legs, action recognition also provides strong priors on the pose sequences. Furthermore, if actions are limited to predefined categorizes, the actions provide strong constraints on the plaussible poses in space and time [7].

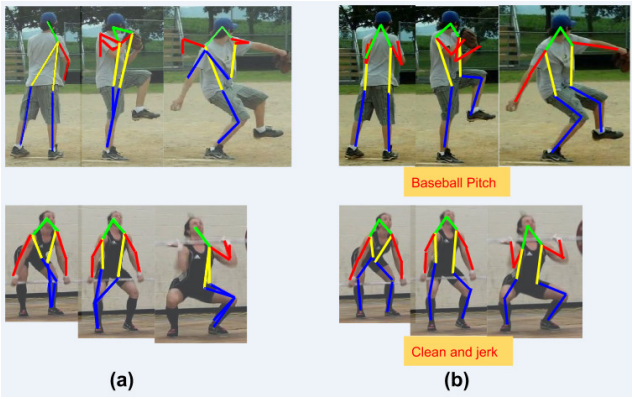

Figure 1. (a) Single frame human poses estimated by [29]. (b) Action recognition and poses estimation by our approach.

Many methods for action recognition bypass body poses and achieve promising results by using coarse/mid-level features for action classification on some datasets[6, 10, 26, 12, 18, 2, 33, 30]. In this paper, we will jointly train coarse/mid-level features with pose estimation so that these features are better aligned with body parts and improve the results.

The prevailing methods for pose estimation from still images adopt probabilistic and compositional graphical models where nodes represent part appearance and edges represent geometrical deformation[29, 19, 17, 20]. Errors mainly arise from small parts, like forearms and wrists due

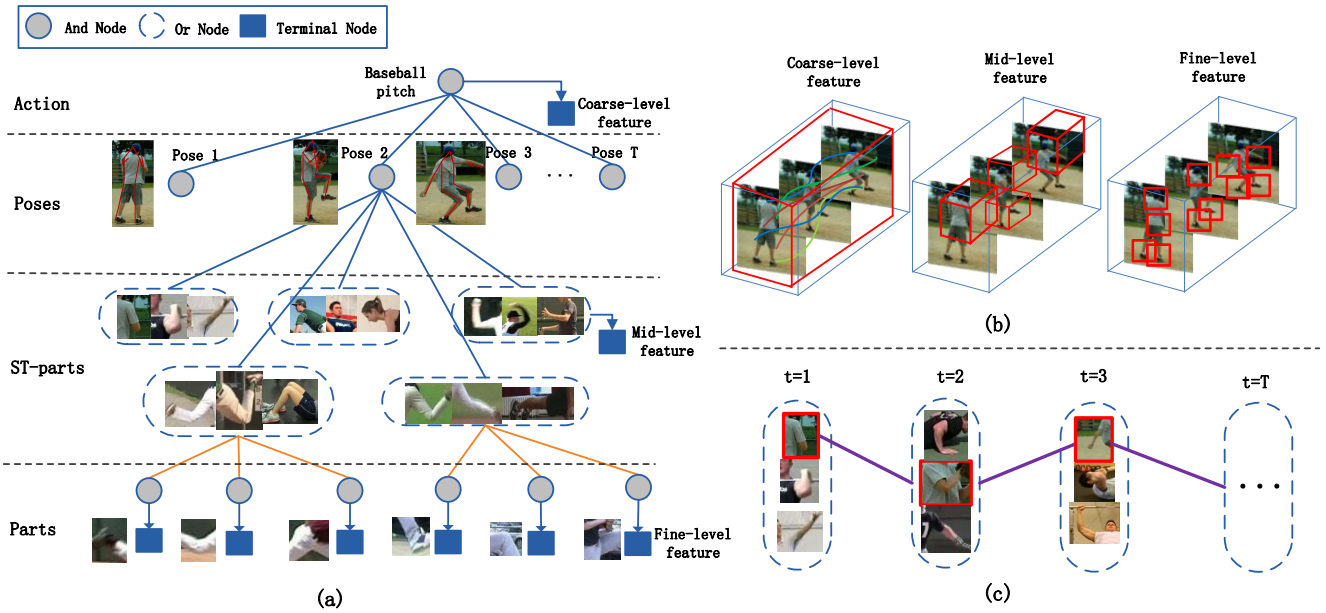

Figure 2. (a) Our spatial-temporal AOG model for action "Baseball pitch". The action is decomposed into poses, ST-parts, and parts. Each ST-part is an Or-Node that represents the mixture components. For simplicity we only draw all nodes at the second frame. The orange edges represent geometric deformations between ST-parts and parts. (b) The three feature levels. Action nodes, ST-part nodes and part nodes connect to terminal nodes that represent coarse-level, mid-level and fine-level features respectively. (c) An example of temporal relation on ST-part 'left arm'. The purple edges connecting five ST-parts at adjacent frames capture the temporal co-occurrence and deformation relations. During inference we select the best component (red rectangle) for each ST-part.

to large variation and blending with background features. Video pose estimation methods capture motion information by adding many pairwise terms among parts at subsequent frames to the graphical model[3, 36, 21], however, these models are loopy and require approximate inference. The smoothness features on pairwise terms are restricted to videos with slow motion and small appearance variation, but such prior assumptions break in action datasets. Human motion can become much larger and the changing of viewpoint makes appearance inconsistent at adjacent frames. As illustrated in Fig. 1, we improve the estimation of part locations by using action specific information.

1.2. Method Overview

This paper integrates the training and testing of action recognition and video pose estimation. During training, information from both tasks is utilized to optimize model parameters and in testing the action labels and part locations are inferred jointly.

We start by building a spatial-temporal And-Or graph model[37][23][14][13] to represent actions and poses jointly. Hierarchical structure of our model can represent top-down part geometric configurations in a single frame and lateral temporal pose relation in subsequent frames. On the top layer, the low-resolution action information is captured by coarse-level features and the action is decomposed into poses at each frame. Each pose is decomposed into five independent mid-level 'ST-parts' (ST means Spatio-Temporal) that cover a large portion of human bodies and are robust to image variations. All fine-level parts are conditioned on their ST-part parents. Each ST-part is discretized into several components by clustering. The ST-parts with the same component can be seen as a poselet[1] that has small variation of appearance and deformation and each component is represented by mid-level features and fine-level part features from single image pose estimation.

In order to capture the specific motion information of each action, ST-parts at adjacent frames are connected to represent temporal co-occurrence and deformation. The model parameters at three levels are trained separately by S-SVM and combined by a mixture of experts method. Due to the independence between ST-parts of each pose, we can infer both action label and poses efficiently by DP.

2. Related Works and Our Contributions

Action recognition and pose estimation are both popular topics in computer vision and there are numerous literature. This section refers to some recent work on both topics. Action recognition methods are grouped into two categories: coarse/mid-level feature based and pose feature based. Pose estimation methods are reviewed with two aspects: single image pose estimation and video pose estimation.

Coarse/mid-level feature based methods. The most successful framework is built on spatial-temporal interest points such as cubiods[6] and 3D Harris corner[10]. This framework extends object detection using 2D spatial interest points. After the interest points are detected, appearance and motion features like HOG[4] and HOF[11] are extracted and the bag-of-words representation is used for classification. Instead of using interest points, Wang et al. [26] extracts dense trajectories by optical flow and builds a bag-of-words representation on trajectory aligned features. While it has achieved good performance on many action datasets, it highly relies on the quality of optical flow. Although the coarse/mid-level features based methods succeed on some datasets, they offer no intuition about the relation between pose and action. The learning and inference with these methods is simple and fast, and they can work on the low-resolution videos.

Pose feature based methods. Recently, due to the great progress made in pose estimation, many action recognition methods try to borrow strength from high-level pose information. Yao et al. [34] represents action as several key-poses with an AOG model. Each key pose corresponds to a latent variable in a HMM model. Wang et al. [25] first runs pose estimation on video frames and builds pose features directly on the estimated poses, classifying with a bag-of-words framework. Jiang Wang et al. [27] develops MST-AOG model for cross-view action recognition. The 3D skeleton training data is applied to help mine the discriminative parts.

Single image pose estimation methods. The most popular framework used in single image pose estimation is to build a part graphical model based on human joints. Yang and Ramanan [29] build a tree-structure spring model to capture both spatial and co-occurrence relations between parts. Brandon et al. [19] uses a compositional AOG model to represent large appearance and geometry variation and image segmentation is employed to help distinguish the parts from cluttered background. Pishchulin et al. [17] builds a more flexible graphical model with strong local appearance representations and the mid-level semi-global poselets are combined with fine part appearance model.

Video pose estimation methods. Cherian et al. [3] extends the graphical model with temporal edges between parts at adjacent frames. The geometric and appearance comparability between parts is captured by temporal edges and approximate inference is performed on the highly loopy graphical model. Instead of using a graphical model Shen et al. [22] formulates the video pose estimation as a matching problem that tries to match the dense trajectories from 2D video with the projection of the 3D trajectories of human motion under different viewpoints in a 3D database.

To the best of our knowledge Yao et al. [31] is the only paper that tries to couple action recognition and pose estimation. It formulates the pose estimation as an optimization over a set of action specific manifold and conducts the two tasks iteratively. In training it requires that each video is from multiple views however we can work on datasets in which each sample is from only one view.

This paper combines action recognition and video pose estimation in a unified framework with a spatial-temporal And-Or Graph model. This paper makes three contributions to both action recognition and video pose estimation problems:

i) It proposes a spatial-temporal AOG model to integrate action recognition and video pose estimation. The two tasks are mutually benefit from each other in training and testing.

ii) It represents actions at three scales. Coarse and middle level features are trained jointly with pose features.

iii) It outperforms state-of-art action recognition and pose estimation methods on two action datasets: Penn Action dataset and sub-JHMDB dataset.

3. Representation and Modeling

3.1. Spatial-Temporal And-Or Graph Model

Fig.2 shows our spatial-temporal AOG model for representing action and poses. There are three kinds of nodes: And nodes, Or nodes and Terminal nodes. The And node captures the decomposition of a large entity. In our case the action and poses are represented by And nodes because they are decomposed into several children. The Or node represents structural variations. Here each ST-part is an Or node because it has different components. The Terminal node is observable and directly associates with image evidence. We have three kinds of terminal nodes to represent actions at different scales. The terminal nodes associated with action and ST-parts represent coarse and mid-level features and the terminal nodes at bottom level represent fine part features.

To unify action recognition and video pose estimation each action example A is represented by the poses p_t at each frame:

$$A = \{p_1, p_2, ..., p_T\} \tag{1}$$

T is the number of frames. Each pose p_t is represented by an And node and decomposed into several ST-parts l_i (Fig. 2(a)):

$$p_t = \{l_1, l_2, ..., l_M\} \tag{2}$$

M is the number of ST-parts. Each ST-part l_i is a mixture components model with several parts o_j:

$$l_i = \{o_0, ..., o_{N_i-1}, c_i\} \tag{3}$$

$o_j = \{x_j, y_j\}$ denotes the position of part j which should be one of the human joints, $o_j \in \Omega_{\text{part}}, \Omega_{\text{part}} = \{'head','torso','leftarm','rightarm', ...\}$, N_i is the number of parts that belong to parent i, o_0 is the root part for this ST-part. c_i is the component id and $c_i \in \{1, 2, ..., z_i\}$,

z_i is the number of components of ST-part i. The ST-parts with the same component have small appearance and geometrical variations and represent a motion status of the action. The learning of ST-parts will be discussed in the next section.

We divide the feature vector of ST-part into two categories: **classification feature** for action classification and **detection feature** for regularization.

Classification feature includes two terms: $\psi(l_i)$ and $\psi(c_i)$. $\psi(l_i) = [d_1\, d_2\, ...\, d_{z_i}]^T$ where $d_j = (o_0 - u_j)$ is the normalized Euclidean distance between the root part and the component center. $\psi(c_i) = [0, 0, \mathbf{1}(c_i), ..., 0, 0]$ is a z_i dimension indicator where the entry corresponding to component c_i is one and the others are zero.

Detection feature contains two portions: the part score $\sum_{j=0}^{N_i} S(o_j)$ and the deformation score $\sum_{j=1}^{N_i} S(o_j, o_0)$. The two scores are directly obtained from a single image pose estimation[29] and used to regularize action classification.

There are two kinds of edges in our model: orange edges represent the geometric deformation in a single frame and purple edges represent the smoothness and temporal co-occurrence of ST-parts at adjacent frames.

Deformation feature is a four-dimension vector which models deformation between ST-part and part as a 2D gaussian distribution: $\psi(E_d) = [dx, dy, dx^2, dy^2]^T, E_d \in \Omega_{\mathrm{D}}$.

Temporal co-occurrence feature at ST-part i of frame t is a $z_i \times z_i$ dimension indicator: $\psi(E_o) = [0, 0, \mathbf{1}(c^t)\mathbf{1}(c^{t+1}), ..., 0], E_o \in \Omega_{\mathrm{O}}$ which means that only the entry corresponding to components c^t and c^{t+1} is one and the others are zero.

Smoothness feature $d(l_i^t, l_i^{t+1})$ is the negative Euclidean distance between the root parts of l_i^t and l_i^{t+1}.

Although the action is represented by a sequence of poses, it is insufficient to only use pose features for action recognition because low resolutions makes part detection unreliable. Here we borrow the strength from coarse-level and mid-level features for compensation. For the coarse feature ψ_L, we follow the framework of [26] to extract the bag-of-words feature on the dense trajectories. For the mid-level feature ψ_M, we use the method from [27] to train HOG/HOF templates for each selected ST-part component, using the filter responses as features.

3.2. Score Functions

In this section, we introduce the score functions of our model in a bottom-up fashion. For simplicity we drop the action label in all formulations in this section.

The terminal nodes in the bottom layer ground all parts to image data. Instead of training part templates with action, we train them independently by single image pose estimation [29]. The part scores and part deformation scores are obtained directly from [29].

The score of ST-part i is defined by:

$$S(l_i) = S_d(l_i) + S_h(l_i) + \lambda \sum_{j=0}^{N_i} S(o_j) + \lambda \sum_{j=1}^{N_i} S(o_j, o_0) \tag{4}$$

There are four terms contributing to the ST-part score. The first two terms are classification scores and the last two terms are detection scores. $S_d(l_i) =< \omega_d^{l_i}, \psi(l_i) >$ measures the compatibility of component c_i. $S_h(l_i) =< \omega_h^{l_i}, \psi(c_i) >$ is the histogram score of component c_i. $S(o_j)$ is the score of part j and $S(o_j) = P(o_j)$ where $P(o_j)$ is the part marginal score from pose estimation. $S(o_j, o_0) =< \omega^{ij}, \psi(E_d^{ij}) >$ is the deformation score of part j related to the root part. Parameter λ is the weight for detection score. The inference algorithm will search all possible ST-parts in the feature pyramid and output a top candidate list for each frame.

Each pose is composed of M ST-parts thus the score is written as a summation of their scores.

$$S(p_t) = \sum_{i=1}^{M} S(l_i^t) \tag{5}$$

The relation between ST-parts in a single image is ignored so they are independent of each other, avoiding the loopy graph structure that is a common case in video pose estimation. The details will be discussed in section 5.

In our model, each action example is a sequence of poses following the transitions between poses at adjacent frames. Thus, the fine-level score of an action can be formulated as:

$$S_H(A) = \sum_{t=1}^{T} S(p_t) + \sum_{t=1}^{T-1} S(p_{t+1}|p_t) \tag{6}$$

$S(p_t)$ is defined in Eq. (5) and $S(p_{t+1}|p_t)$ measures the transition score between two poses. The transition relation of two poses is captured by transitions between their ST-parts and it is thus written as a summation of transition scores of ST-parts.

$$S(p_{t+1}|p_t) = \sum_{i=1}^{M} S(l_i^{t+1}|l_i^t) \tag{7}$$

The transition score between two ST-parts is defined as:

$$S(l_i^{t+1}|l_i^t) = S(c_i^t, c_i^{t+1}) + \beta d(l_i^t, l_i^{t+1}) \tag{8}$$

It includes two components: the co-occurrence score $S(c_i^t, c_i^{t+1}) =< \omega_o^{l_i}, \psi(E_o^{c_i^t, c_i^{t+1}}) >$ and smoothness score $\beta d(l_i^t, l_i^{t+1})$, where β is the weight for the smoothness.

The fine-level score of one image sequence is rewritten as follows, combining eqns. (5), (6) and (7).

$$S_H(A) = \sum_{i=1}^{M} (\sum_{t=1}^{T} S(l_i^t) + \sum_{t=1}^{T-1} S(l_i^{t+1}|l_i^t)) \tag{9}$$

In this form, the fine-level score is only related to the ST-parts. The inference algorithm will search for the positions and components of ST-parts that maximize this score.

With coarse-level and mid-level scores, the action score can be written as,

$$S(A) = \pi_L(A)S_L(A) + \pi_M(A)S_M(A) + \pi_H(A)S_H(A) \quad (10)$$

$S_L(A) =< \omega_L, \psi_L(A) >$ is coarse-level score and $S_M(A) =< \omega_M, \psi_M(A) >$ is mid-level score. The weights $\pi_L(A) =< \omega'_L, \phi'_L(A) >$, $\pi_M(A) =< \omega'_M, \phi'_M(A) >$ and $\pi_H(A) =< \omega'_H, \phi'_H(A) >$ are linear functions on features of action example A.

4. Inference

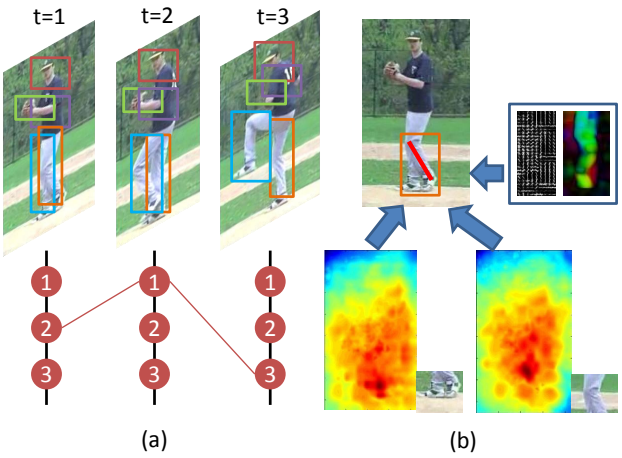

Figure 3. An example of our inference method. (a) For each frame we generate several ST-part candidates and obtain the best path for each ST-part by DP. (b) The ST-part is represented by the mid-level features (HOG and HOF template) and fine-level features (scores of knee and ankle).

The objective of our inference is to find the action label and part locations. The coarse-level score $S_L(A)$ and mid-level score $S_M(A)$ are computed directly by linear-SVM. As illustrated in Fig.3 (a), The fine-level action score $S_H(A)$ is divided into M independent terms each of which corresponds to the summation of unary scores and binary transition scores for one ST-part, thus dynamic programming can be used to find the best ST-part path:

$$[l_i^1, l_i^2, ..., l_i^T] = \arg\max \sum_{t=1}^{T} S(l_i^t) + \sum_{t=1}^{T-1} S(l_i^{t+1}|l_i^t) \quad (11)$$

This procedure is repeated M times to find the total M best paths for each action label. Finally the action label with maximum score is obtained in Eq. (10).

With the best action label, we trace back to the best ST-part paths for the action and obtain all joint locations. Notice that the joints 'left shoulder' and 'right shoulder' are shared by two ST-parts and we pick them from the ST-part 'head shoulder' because it is more robust than 'left arm' and 'right arm'.

To speed up computation, we first run [16] for each frame and compute response maps for all ST-parts. After non-maximum suppression we pick the ST-part candidates that have a score above τ. We connect all candidates on consecutive frames and compute their unary scores and binary transition scores. To determine the optimal threshold τ, we compute ST-part scores on the ground truth of all training images and pick the highest value for the threshold that does not prune the optimal one on training examples.

5. Learning

Our learning process includes two main stages: The first stage is to learn ST-parts. The second stage is to learn the model parameters for three levels including weights for unary ST-part score , temporal score between ST-parts in adjacent frames and classification weights for each action.

5.1. Learning ST-parts

As a mid-level representation of human pose, ST-parts are much more robust to image variations than fine parts, especially on action datasets containing large appearance, geometric and motion variations that make fine parts hard to detect. With pose annotations we can learn ST-parts from training data.

5.1.1 ST-part Representation

We use 13 joints to represent the human subject. The 13 joints are divided into 5 ST-parts: 'head-shoulder','left arm','right arm','left leg','right leg' each of which includes 3 joints Fig.5(a). In order to compute deformation we define 5 joints as root parts for ST-parts:head, left elbow, right elbow, left knee, right knee. Each ST-part is encoded by a feature vector:

$$f(l_i^t) = [\Delta p_1, \Delta p_2, \Delta p_0^t, \Delta p_1^t, \Delta p_2^t] \quad (12)$$

$\Delta p_1 = p_1 - p_0$ and $\Delta p_2 = p_2 - p_0$ are the offsets of parts relative to the root part. $\Delta p_0^t = [p_0^{t-1} - p_0^t, p_0^{t+1} - p_0^t]$ is the temporal offset of root part relative to the same joint in previous frame and the next frame. Δp_1^t and Δp_2^t are defined in the same way. Using the temporal offset as a feature is important because some ST-parts have the same joint configuration and can be only distinguished by motion. To make the feature invariant to scale, we estimate the pose scale s_t at each frame by head length, and then the feature is normalized by the scale factor: $f(l_i^t) = f(l_i^t)/s_t$.

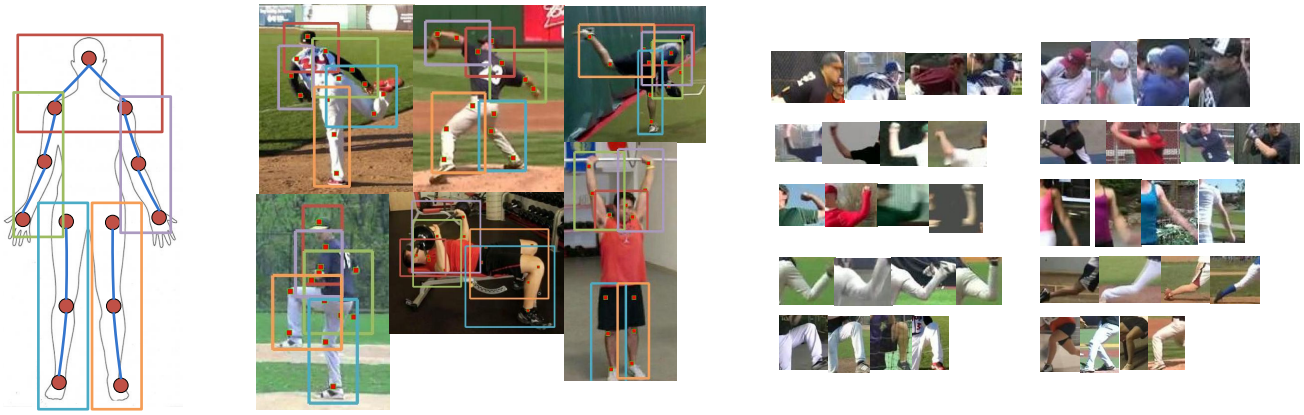

Figure 4. (a) The 13 joints used in our model. They are divided into five ST-parts each of which contains 3 joints. (b) Some examples of pose annotations in training data and their generated ST-parts. (c) Some examples of two components for each ST-part.

5.1.2 ST-part Clustering

To capture image variations from different viewpoints and actions each ST-part is represented as a mixture of components model and the components are obtained by doing k-means on the features $f(l_i^t)$. In order to make the ST-part component compact in appearance and motion, we first run k-means on the training examples with same action label and view label to get many small clusters each of which has small variation. Clusters that have few examples and belong to only one video are removed as annotation errors. Finally we combine these clusters according to their distance to let them to be shared by different actions and viewpoints. See some examples in Fig.5 (c).

5.2. Learning Model Parameters

5.2.1 Learning Coarse-level and Mid-level Templates

The coarse-level template ω_L is learned by linear-SVM on the dense trajectory features[26]. These features don't need any pose information and they capture the appearance and short-term motion on the moving blocks. The mid-level information is captured by HOG/HOF templates of ST-parts. Following[27], we train HOG/HOF templates on our ST-part components with SVM and convolute them with training images. The feature vector is constructed by performing spatial-temporal max-pooling on response maps, and the template ω_M is learned by linear-SVM.

5.2.2 Learning Fine-level Parameters

The parameters we need to learn for the fine-level score include $\omega_d^{l_i}$ and $\omega_h^{l_i}$ for the compatibility score and histogram score of each ST-part, $\omega_o^{l_i}$ for the ST-part co-occurrence score. We adapt latent Structure-SVM for learning those parameters with regularization. Although all training data have part annotations and we have ground truth for part locations and ST-part components, only using ground truth may hurt performance because there is a large difference between pose estimation results and ground truth in such challenging action datasets. Thus we allow the parts to move between the top N detected parts that are within a certain distance of the ground truth part locations. Learning iterates between two steps until convergence:

i) To train parameters $w = [\omega_d^{l_1}\ \omega_h^{l_1}\ \omega_o^{l_1} ... \omega_d^{l_M}\ \omega_h^{l_M}\ \omega_o^{l_M}]$, we discard the detection scores $\lambda \sum_{j=0}^{N_i} S(o_j)$ and $\lambda \sum_{j=1}^{N_i} S(o_j, o_0)$ and the smoothness score $\beta d(l_i^t, l_i^{t+1})$ and train the parameters with detected poses h_i. For the first iteration, h_0 is set to ground truth poses. This is formulated as a supervised multi-class classification problem:

$$\min_{\omega_t} \tfrac{1}{2}\|\omega_t\|_2 + \tfrac{C}{n}\sum_{i=1}^{n}\xi_i, \qquad (13)$$
$$s.t. \quad \max_{\hat{y}\in\mathcal{Y}} \omega_t^T(\phi(x_i, y_i^t) - \phi(x_i, \hat{y}_i^t)) \geq \Delta(y_i, \hat{y}_i) - \xi_i,$$

Here $y_i^t = (a_i, h_i^t)$ where a_i is action label. $\Delta(y_i, \hat{y}_i)$ is 1 if $a_i = \hat{a}_i$and 0 otherwise. t indexes the iteration.

ii) After computing parameters at iteration t, we add the detection score and the smoothness score back into the fine-level score function and infer the poses for each training example. λ and β are determined by experiments. Similar to inference in testing, we first generate the top N ST-parts candidates within a certain distance around the ground truth and find the best ST-part paths among those candidates under the ground truth action label by Eq. (11). Then we get the poses h_i^{t+1} from the poselets and go back to step 1.

After learning the parameters for the three levels, we obtain the scores for the three levels separately. Finally we learn the weights $\pi_L(A)$, $\pi_M(A)$ and $\pi_H(A)$ to combine them for the final action score. We formulate this combination in the mixture of experts framework[9] where each

expert corresponds to a classifier in each level. The weights are computed on each action example, so different weights indicate which expert the example prefers to use. Here we concatenate scores of different categories at each level as features to learn the weights.

6. Experiments

We test our method on two public action datasets: the Penn Action dataset[35] and the sub-JHMDB dataset[8]. Both datasets are proposed for the purpose of action recognition but they also provide annotations of human joints which are required by our training approach. The performance of both action recognition and pose estimation are evaluated on each dataset.

6.1. Evaluation on Penn Action Dataset

The Penn Action Dataset contains 15 action categories and the annotations include action labels, rough view labels and 13 human joints for each image. The occlusion label of each joint is also provided. We follow [35] to split the data into 50/50 for training/testing. The action 'playing guitar' and several other videos are removed because less than one third of a person is visible in those data. We find that there exist some un-annotated joints that always remain at the left-top corner of image. To correct those errors we train a regression model to predict positions of un-annotated joints by using the visible neighbor joints from videos with the same action and view label. In order to get diverse poses to train [29] we first cluster the training data based on whole pose features to get 500 clusters. Then we uniformly select total 5000 images from those clusters as training images. We use the code provided by [29], and we set part mixture number to 8: 6 for visible joints and 2 for occluded joints.

The number of mixture components of 5 S-T parts are 43, 37, 31, 56, 58. We find that more components does not improve performance but greatly increase training burden. The parameters $\lambda = 10$ and $\beta = 0.01$ for detection score and smooth score are determined by cross-validation on the training data. Training converges in only 3 iterations. The coarse-level and mid-level action templates are trained by the code from [26] and [27]. The number of candidates of the ST-part 'head-shoulder' is around 200 and of other ST-parts is around 1000 because the parts 'head' and 'shoulder' only have high scores on a few locations whereas other parts have much larger variations on the score map.

Table.2 compares the action recognition accuracy between previous methods and ours. We use the numbers of STIP, Dense, Action Bank and Actemes from [35]. Ours(fine) is trained by only fine-level features and Ours(all) is trained with all feature levels. With only fine-level features, the performance is not very good, but when coarse/mid-level features are added in the performance is improved due to the low resolution and heavy occlusion that make part detection unreliable and not good enough to classify actions.

Method	Accuracy
STIP[35]	82.9%
Dense[26]	73.4%
MST[27]	74.0%
Action Bank[35]	83.9%
Actemes[35]	79.4%
Ours(fine)	73.4%
Ours(all)	**85.5%**

Table 2. Recognition accuracy on Penn Action dataset. Action Bank is not directly comparable since it uses other training dataset.

The confusion matrix of Ours(all) is shown in Figure. 6. Our approach performs well on the actions such as 'bowl', 'pull up', 'push up' and 'squat', however we achieve low accuracy on actions with fast movement such as 'tennis forehand' because the motion blur makes the positions of critical parts like wrists always wrong.

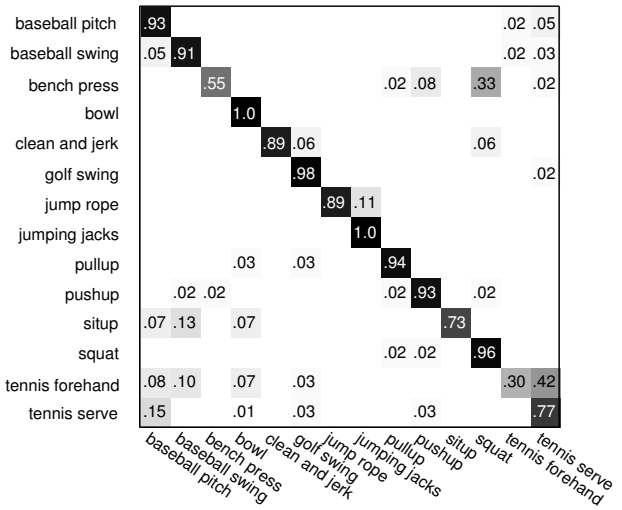

Figure 6. The confusion matrix of our method on Penn Action Dataset.

We compare pose estimation accuracy with Yang et al. [29] and Park et al. [16]. We use their evaluation criteria and set the threshold to 0.2. The results are illustrated in Table. 1. Our method outperforms theirs at every part. It is reasonable that the action specific motion information can help our method to select better parts which are not always the oen with highest score provided by single image based pose estimation.

6.2. Evaluation on sub-JHMDB Dataset

The sub-JHMDB dataset contains 316 clips with 12 action categories. It provides action labels, rough-view labels and 15 human joints for each frame. All joints are inside

	Penn Action Dataset							
	Head	Shou	Elbo	Wris	Hip	Knee	Ankle	mean
[29]	57.9	51.3	30.1	21.4	52.6	49.7	46.2	44.2
[16]	62.8	52.0	32.3	23.3	53.3	50.2	43.0	45.3
[3]	–	–	–	–	–	–	–	–
Ours	**64.2**	**55.4**	**33.8**	**24.4**	**56.4**	**54.1**	**48.0**	**48.0**

	sub-JHMDB Dataset							
	Head	Shou	Elbo	Wris	Hip	Knee	Ankle	mean
[29]	73.8	57.5	30.7	**22.1**	69.9	58.2	48.9	51.6
[16]	79.0	60.3	28.7	16.0	74.8	59.2	49.3	52.5
[3]	47.4	18.2	0.08	0.07	–	–	–	16.4
Ours	**80.3**	**63.5**	**32.5**	21.6	**76.3**	**62.7**	**53.1**	**55.7**

Table 1. Pose estimation accuracy in %. The left table shows the results of Penn Action Dataset and the right table shows the results of sub-JHMDB Dataset.

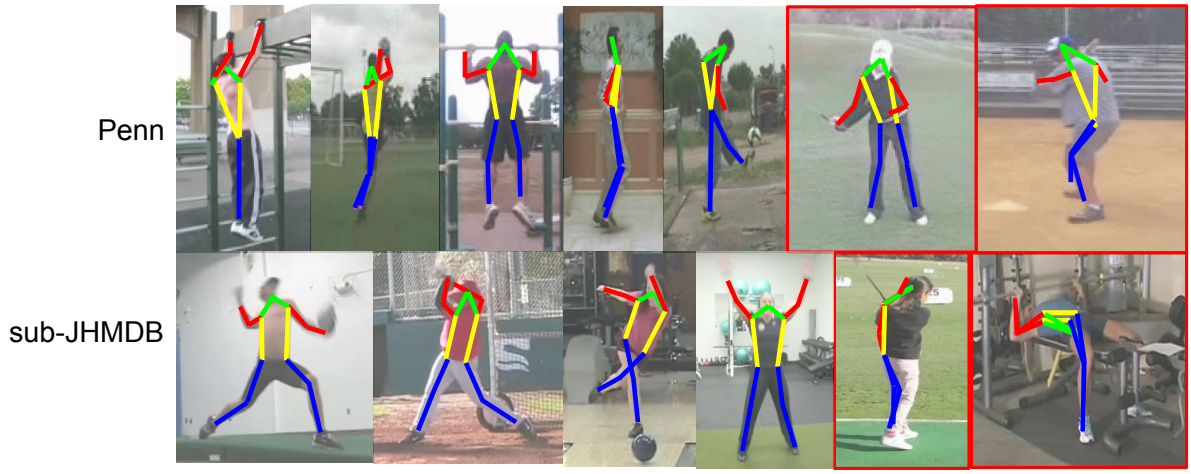

Figure 5. Some pose estimation results of our method on the two datasets. The last two columns show failure examples with red rectangle.

the image and there are no un-annotated joints. We use 13 human joints to train the single image pose estimation. We also do clustering on all frames using the whole pose features and select a total 1500 images from clusters for training. The part mixture number is set to 6.

We use the 3-fold cross validation setting provided by the dataset to do experiments. The number of mixture components of 5 ST parts are 36, 42, 39, 64 and 64. The parameters $\lambda = 20$ and $\beta = 0.01$ for detection score and smooth score are decided by cross-validation and the training converges in 3 iterations.

Method	Accuracy
Dense[8]	46.0%
MST[27]	45.3%
Pose[8]	52.9%
Ours(fine)	55.7%
Ours(all)	**61.2**%

Table 3. Recognition accuracy on sub-JHMDB dataset.

Table. 3 compares our action recognition performance with others. We use the numbers of 'Dense' and 'Pose' from [8]. For Pose[8], we use the highest number they obtained by using pose features extracted from pose estimation. With only fine-level features our method already outperforms others. With coarse/mid features the accuracy is increased by nearly 6 percent because there are many low-resolution videos with large errors of pose estimation.

The comparison of pose estimation is illustrated in Table. 1. Our method outperforms [29] the most at parts 'Head' and 'Hip' by nearly 7%, however for the parts 'Elbows' and 'Wrists' our performance is comparable which we believe is caused by those parts that are very subtle and because the specific action motion information may prefer the distinguished part locations which are never in the right positions. To compare with [3], we re-train their method on our dataset, and they only estimate the joints in upper body. Results show that the pairwise smoothness features they use are not working well in the action dataset with large motion and appearance changing.

7. Conclusion

We have proposed a new framework to joint action recognition and pose estimation, which are traditionally trained separately and combined sequentially. One limitation of our method is that we do not handle the self-occlusion explicitly which always appears in action datasets and is a big challenge for pose estimation. In the future, we are going to integrate the 3D pose estimation with the current framework, because only with the help of 3D information we can solve the occlusion issue.

Acknowledgement: This work is supported by three

grants: DARPA MSEE FA 8650-11-1-7149, ONR MURI N00014-10-1-0933, and NSF IIS-1423305.

References

[1] L. Bourdev and J. Malik. Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations. In *ICCV*, 2009.

[2] W. Chen, C. Xiong, and J. J. Corso. Actionness ranking with lattice conditional ordinal random fields. In *CVPR*, 2014.

[3] A. Cherian, J. Mairal, K. Alahari, and C. Schmid. Mixing Body-Part Sequences for Human Pose Estimation. In *CVPR*, 2014.

[4] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[5] V. Delaitre, J. Sivic, and I. Laptev. Learning person-object interactions for action recognition in still images. In *NIPS*, 2011.

[6] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior Recognition via Sparse Spatio-Temporal Features. In *ICCV VS-PETS*, 2005.

[7] J. Gall, A. Yao, and L. Gool. 2D Action Recognition Serves 3D Human Pose Estimation. In *ECCV*, 2010.

[8] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, 2013.

[9] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. In *IJCNN*, 1993.

[10] I. Laptev and R. Cedex. On space-time interest points. *IJCV*, 64(2-3):107–123, 2005.

[11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[12] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng. Learning hierarchical spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

[13] B. Li, W. Hu, T. Wu, and S. C. Zhu. Modeling occlusion by discriminative and-or structures. In *ICCV*, 2013.

[14] B. Li, T. Wu, and S. C. Zhu. Integrating context and occlusion for car detection by hierarchical and-or model. In *ECCV*, 2014.

[15] S. Maji, L. Bourdev, and J. Malik. Action Recognition from a Distributed Representation of Pose and Appearance. In *CVPR*, 2011.

[16] D. Park and D. Ramanan. N-best maximal decoders for part models. In *ICCV*, 2011.

[17] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong Appearance and Expressive Spatial Models for Human Pose Estimation. In *ICCV*, 2013.

[18] M. Raptis, L. Kokkinos, and S. Soatto. Discovering Discriminative Action Parts from Mid-Level Video Representations. In *CVPR*, 2012.

[19] B. Rothrock, S. Park, and S. Zhu. Discriminative Pose Estimation using Grammar and Segmentation. In *CVPR*, 2013.

[20] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *CVPR*, 2013.

[21] B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *CVPR*, 2011.

[22] H. Shen, S. Yu, D. Meng, and A. Hauptmann. Unsupervised video adaptation for parsing human motion. In *ECCV*, 2014.

[23] X. Song, T. Wu, Y. Jia, and S. C. Zhu. Discriminatively trained and-or tree models for object detection. In *CVPR*, 2013.

[24] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *ICCV*, 2013.

[25] C. Wang, Y. Wang, and A. Yuille. An Approach to Pose-Based Action Recognition. In *CVPR*, 2013.

[26] H. Wang, A. Klaser, C. Schmid, and C. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013.

[27] J. Wang, B. X. Nie, Y. Xia, Y. Wu, and S. C. Zhu. Cross-view Action Modeling, Learning and Recognition. In *CVPR*, 2014.

[28] W. Yang, Y. Wang, and G. Mori. Recognizing Human Actions from Still Images with Latent Poses. In *CVPR*, 2010.

[29] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *PAMI*, 35(12):2878–2890, 2012.

[30] Y. Yang, I. Saleemi, and M. Shah. Discovering motion primitives for unsupervised grouping and one-shote learning of human actions, gestures, and expressions. *PAMI*, 35(7):1635–1648, 2013.

[31] A. Yao, J. Gall, and L. Gool. Coupled action recognition and pose estimation from multiple views. *IJCV*, 100(1):16–37, 2012.

[32] B. Yao and F. F. Li. Recognizing human actions in still images by modeling the mutual context of objects and human poses. *PAMI*, 34(9):1691–1703, 2012.

[33] B. Yao and S. C. Zhu. Learning deformable action templates from cluttered videos. In *ICCV*, 2009.

[34] B. Z. Yao, B. X. Nie, Z. Liu, and S. C. Zhu. Animated pose templates for modeling and detecting human actions. *PAMI*, 36(3):436–452, 2013.

[35] W. Zhang, M. Zhu, and K. Derpanis. From Actemes to Action: A Strongly-supervised Representation for Detailed Action Understanding. In *ICCV*, 2013.

[36] F. Zhou and F. D. Torre. Spatio-temporal Matching for Human Detection in Video. In *ECCV*, 2014.

[37] S. C. Zhu and D. Mumford. A stochastic grammar of images. *Foundations on Trends in Computer Graphics and Vision*, 2(4):259–362, 2006.