Adaptive Pose Priors for Pictorial Structures

Benjamin Sapp Chris Jordan Ben Taskar
University of Pennsylvania,
Philadelphia, PA 19104, USA,
{bensapp,wjc,taskar}@seas.upenn.edu

Abstract

Pictorial structure (PS) models are extensively used for part-based recognition of scenes, people, animals and multi-part objects. To achieve tractability, the structure and parameterization of the model is often restricted, for example, by assuming tree dependency structure and unimodal, data-independent pairwise interactions. These expressivity restrictions fail to capture important patterns in the data. On the other hand, local methods such as nearest-neighbor classification and kernel density estimation provide non-parametric flexibility but require large amounts of data to generalize well. We propose a simple semi-parametric approach that combines the tractability of pictorial structure inference with the flexibility of non-parametric methods by expressing a subset of model parameters as kernel regression estimates from a learned sparse set of exemplars. This yields query-specific, image-dependent pose priors. We develop an effective shape-based kernel for upper-body pose similarity and propose a leave-one-out loss function for learning a sparse subset of exemplars for kernel regression. We apply our techniques to two challenging datasets of human figure parsing and advance the state-of-the-art (from 80% to 86% on the Buffy dataset [8]), while using only 15% of the training data as exemplars.

1. Introduction

Part-based models for recognition of articulated objects, proposed nearly forty years ago by Fischler and Elschlager [10], represent an object as a collection of distinctive parts and geometric relationships between them. The model characterizes local visual properties of object parts and posits spring-like connections between pairs of parts, which express variability of part locations. The model determines an object match in an image by selecting part locations that minimize appearance matching costs and deformation costs for pairs of connected parts. Improved methods for estimating the model parameters from data account for some of its current popularity. Recent work [5, 7, 17, 8, 6, 1] has shown promising recognition results for human figures, animals, faces and many other multi-part objects.

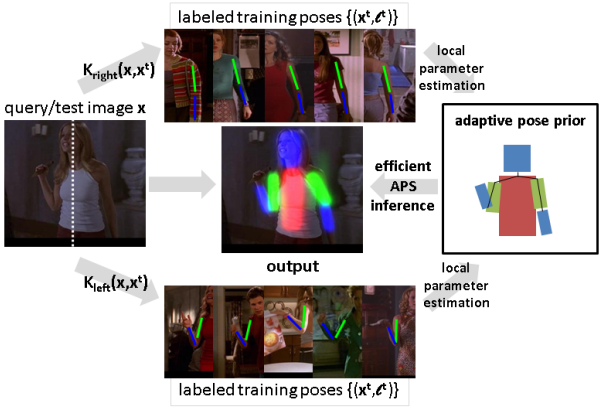

Figure 1. Overview of our system. For each test example, we estimate a subset of the pictorial structure parameters as a kernel-weighted sum of training examples, based on their similarity to the test image. The form of the model and inference remain the same, but we gain more flexibility by adapting the model to the image.

However, a common problem in such models is poor localization of parts that have weak appearance cues or are easily confused with background clutter (for example, state-of-the-art accuracy for lower arms in human figures is almost half of that for torso or head [1]). This problem is due in large extent to restrictions on the expressivity of the model to achieve tractability of inference. For example, [5] assume a tree structure of interactions between parts and unimodal, data-independent pairwise deformation costs. These expressivity restrictions fail to capture important patterns in the data. Adding a latent "pose" variable into the model [13] partially addresses this problem. Reasoning about occlusions of parts captures important non-local dependencies, but leads to intractable inference [19]. Several recent works used iterative procedures which re-estimate part-appearance models in terms of color and location based on initial predictions of the model [17, 8].

In this paper, we propose to increase the expressivity of pictorial structure models while maintaining efficiency of inference by allowing a subset of model parameters to be non-parametric functions of the input. Non-parametric methods, such as nearest-neighbor classification and kernel density estimation, provide expressive flexibility but require large amounts of data to generalize well in high-dimension. We take a semi-parametric approach that combines the tractability of a part-based model representation with the flexibility of non-parametric methods.

Given a query image, the parameters of the model are produced by a kernel-weighted combination of a sparse subset of labeled training poses. This allows for query-adaptive, image-dependent PS parameters. In particular, the pairwise parameters can now adapt to the query image's appearance, unlike in previous PS models in which the means and covariances of relative part locations are fixed after training and do not use image cues. The adaptive model is applied to the image using standard efficient inference methods. Figure 1 shows an idealized overview of the process.

Our kernel is based on shape information which is complementary to texture and color cues used in previous PS models [16, 8, 9, 3, 1]. It relies on contour similarity and simple figure/ground information proposed by exemplar groundtruth information.

We also address the inherent issues with nearest-neighbor methods by learning a sparse set of exemplars from our training set. This adds robustness to the non-uniform sampling of the example space when using a finite training set and to outliers which can hurt kernel regression estimates. In practice we can discard 85% of the training data and significantly increase the performance and computational efficiency of our method.

Our contributions are **(1)** the Adaptive Pictorial Structures model (APS) **(2)** a simple greedy procedure to obtain a sparse set of exemplars minimizing a leave-one-out loss function **(3)** the design of an effective kernel based on shape information **(4)** state-of-the-art performance on two challenging upper body human pose estimation datasets, without post-processing the output of our model.

2. Related Work

The literature on human pose estimation is vast, as well as the variation in settings: applications range from highly-constrained MOCAP environments (*e.g.* [13]) to extremely articulated baseball players (*e.g.* [15]) to the recently popular "in the wild" datasets Buffy [8] and the PASCAL person layout challenge [4].

We focus our attention here on the work most similar in spirit to ours, namely, pictorial structures models. First proposed in [10], efficient computation methods were introduced in [5]. Advancements were made by Ramanan [17] who proposed learning PS parameters discriminatively by maximizing conditional likelihood. Further improvements were made using iterative parsing [16]—the model is run once using generic detectors, and then image-specific appearance terms are included based on the first parse, and the model is run again. Further gains have been made by restricting the state space [8, 9], adding additional pairwise terms that break the tree-structured assumption [9], and estimating color distributions using *a priori* estimates of where the parts should be [3].

We differ from this progression of PS-based models [17, 16, 8, 9, 3] in several ways: **(1)** We do not employ multiple iterations of parsing, or loopy belief propagation. Instead we perform inference once with a tree-structured model. **(2)** We do not use color information [1]—the driving force behind our kernel is shape information from regions and contours. **(3)** We perform no post-processing of the beliefs of our inference; rather we trust them to be our final answer. Our basic PS implementation most closely resembles [1], but our parameters are discriminitively trained.

3. Adaptive Pictorial Structures

The main contribution of this paper is the Adaptive Pictorial Structures model, which we will refer to as APS. This framework is a modular extension to the classic Pictorial Structures model (PS) and can easily be incorporated into existing implementations. We begin by describing the basic PS model in the next section, and describe APS in Section 3.2.

3.1. Basic PS Model

Pictorial Structures are a class of graphical models where the nodes of the graph represents object parts, and edges between parts encode pairwise geometric relationships. For modeling human pose, the PS model decomposes as a tree structure into unary potentials (also referred to as appearance terms) and pairwise potentials between pairs of physically-connected parts. Figure 2 shows a PS model for 6 upper body parts, with lower arms connected to upper arms, and upper arms and head connected to torso. In previous work [17, 5, 8, 9, 1], the pairwise terms do not depend on data and are hence referred to as a spatial or structural prior. The state of part L_i, denoted as $l_i \in \mathcal{L}_i$, encodes the joint location of the part in image coordinates and the direction of the limb as a unit vector: $l_i = [l_{ix}\ l_{iy}\ l_{iu}\ l_{iv}]^T$. The state of the model is the collection of states of M parts: $p(L = l) = p(L_1 = l_1, \ldots, L_M = l_M)$. The size of the state space for each part, $|\mathcal{L}_i|$, is the number of possible locations in the image times the number of pre-defined discretized angles. The standard PS formulation (see [5]) is

[1] Modulo what is used to compute superpixels and Pb, described in Section 4.1

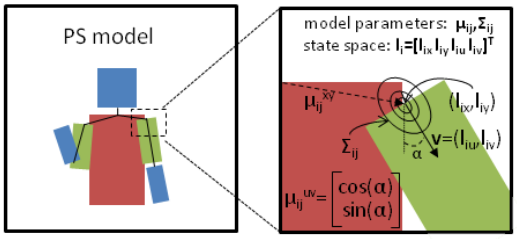

Figure 2. Basic PS model with Gaussian parameters.

usually written

$$p(l|x) \propto \prod_{ij} \exp(-\frac{1}{2}||\Sigma_{ij}^{-1/2}((l_i - l_j) - \mu_{ij})||^2) \quad (1)$$

$$\times \prod_{i=1}^{M} \exp(\mu_i^T \psi_i(l_i, x)) \quad (2)$$

where the parameters of the model are μ_i, μ_{ij} and Σ_{ij}, and $\psi_i(l_i, x)$ are features of the (image) data x at location l_i. The PS model can be interpreted as a set of springs at rest in default positions μ_{ij}, and stretched according to tightness Σ_{ij}^{-1} and displacement $\psi_{ij}(l) = l_i - l_j$. The unary terms pull the springs toward locations with higher scores $\mu_i^T \psi_i(l_i, x)$ which are more likely to be a location for part i.

This log-quadratic form allows inference to be performed faster than $O(|\mathcal{L}_i|^2)$: MAP estimates $\arg\max_{l \in \mathcal{L}} p(L = l|x)$ can be computed efficiently using a generalized distance transform for max-product message passing in $O(|\mathcal{L}_i|)$ time. Marginals of the distribution, $p(L_i|x)$, can be computed efficiently using FFT convolution for sum-product message passing in $O(|\mathcal{L}_i| \log |\mathcal{L}_i|)$ [5].

3.2. APS Model

Our APS model has the following canonical form

$$p(l|x) \propto p_0(l|x) \exp\left(-\frac{1}{2}||\Sigma^{-1/2}(\mu(x) - \phi(l))||^2\right) \quad (3)$$

We treat $p_0(l|x)$ as a fixed portion of our model learned discriminatively *a priori*. The remaining term contains vectors $\mu(x)$ and $\phi(l)$ which include both the unary and pairwise factors. The key to this formulation is that all parameters (unary and pairwise) can have a dependence on data x. We assume that we always have access to a training set, which comes in image data/labels pairs $\{(x^t, l^t)\}_{t=1}^T$, and define each component of μ as a kernel regression estimate of features of the labels in the training set:

$$\mu_i(x) = \frac{\sum_{t \in T} K_i(x, (x^t, l^t))\phi_i(l^t)}{\sum_{t \in T} K_i(x, (x^t, l^t))} \quad (4)$$

The kernel $K_i(x, (x^t, l^t)) \equiv K_i^t(x)$ denotes the similarity between x and training example t, which is a function of the data x and data and labels from the training set. At run-time, no learning is required—discriminative parameters in p_0 are fixed, and kernel-estimated parameters are computed simply via a weighted summation, assuming the kernel function is known[2].

Exemplar selection

Sparse kernel methods, e.g., SVM or RVM, which keep a subset of training data to use for prediction, have been shown theoretically and empirically to lead to better generalization than their dense counterparts [12, 20]. In addition, they require much less computational effort at test time. In real applications, choosing a sparse set of exemplars addresses common training set issues: For one, the distribution of examples does not evenly cover the parameter space—in our setting, for example, there are many redundant poses with arms straight down. Furthermore, outliers which have erroneous high similarity may hurt regression estimates.

For our model, we would like to select a subset of training examples which can provide good kernel regression estimates to the whole training set:

$$s^\star = \arg \min_{s \in \{0,1\}^T} \mathcal{J}(s) \quad (5)$$

$$\mathcal{J}(s) \triangleq \sum_{t=1}^{T} err\left(f(l^t), \frac{\sum_{t'} K^{t'}(x_t) s_{t'} f(l^{t'})}{\sum_{t'} K^{t'}(x_t) s_{t'}}\right) \quad (6)$$

where $err(\cdot)$ is some error function between features of the groundtruth, $f(l^t)$ and their kernel regression estimate. Selection vector s is a binary vector whose components indicate whether corresponding training examples are selected or not. We constrain $K^t(x_t) = 0$, thus this can be viewed as a type of leave-one-out-error loss function on the training set. This binary optimization/subset selection problem is NP-hard. Even a relaxation of the problem to $s \in [0, 1]$ with a convex $err(\cdot)$ function is still non-convex. We instead approximately solve the original problem with a simple greedy, forward selection of training examples: (1) Start with $s \leftarrow \mathbf{0}$. (2) Find an example t' from the set of unselected examples which reduces $\mathcal{J}(s)$ the most when added to the selected set. Set $s_{t'} \leftarrow 1$. (3) Repeat until $s = \mathbf{1}$. (4) Choose $s^\star$ from all vectors s seen during the algorithm as the one with the smallest value $\mathcal{J}(s)$.

As a simple, efficiently computable surrogate to an error function induced by PS inference, we choose $err(\cdot)$ to be the L_1-distance between groundtruth and kernel-estimated arm joint locations.

[2] Throughout the paper, we use the term "kernel" in the statistical sense of a weighting function, as in kernel density estimation—*not* in the sense of the positive semi-definite matrices used in kernel methods to map features to higher-dimensional spaces.

Pairwise potentials

We define pairwise features $\phi_{ij}(l) = l_i - l_j = [l_{ix} - l_{jx} \;\; l_{iy} - l_{jy} \;\; l_{iu} - l_{ju} \;\; l_{iv} - l_{jv}]^T$, for each pair of connected parts. This captures the displacement in position and angle between part i and part j. We express the parameters $\mu_{ij}(x)$ in a locally-parametric form as a weighted sum of displacements in the training set: $\mu_{ij}(x) = \sum_{t=1}^{T} K_{ij}^t(x)\phi_{ij}(l^t) / \sum_{t=1}^{T} K_{ij}^t(x)$.

The pairwise term in the APS thus takes the same form as the standard PS pairwise term $\exp(-\frac{1}{2}||\Sigma_{ij}^{-1/2}(\phi_{ij} - \mu_{ij})||^2)$. In the standard PS framework, the means are directly learned either discriminatively by maximizing conditional log-likelihood (*e.g.* [17]) or generatively by maximizing joint likelihood (*e.g.*, [1]). In our framework, we instead estimate the position and angle of limbs by taking a sum of training instance displacements, weighted by how similar the test and training images appear via $K_{ij}^t(x)$. These produce an example-specific structural prior/part skeleton, as illustrated in Figure 1. If the weights $K_{ij}^t(x)$ are uniform, this is similar to maximizing the joint likelihood of the data with respect to the pairwise parameters.

Unary potentials

We define unary feature $\phi_{il'}(l)$ for each state location l' in each part i. Let $\mathbf{bin}_{uv}(l)$ denote which angular bin l falls into. Then

$$\phi_{il'}(l) = \mathbf{1}(|l_{ix} - l'_{ix}| < \omega_x) \cdot \mathbf{1}(|l_{iy} - l'_{iy}| < \omega_y) \cdot \quad (7)$$

$$\mathbf{1}(|\mathbf{bin}_{uv}(l_i) - \mathbf{bin}_{uv}(l'_i)| < \omega_{uv}) \quad (8)$$

We set each component of ω to be 15% of the corresponding image dimension. In words, unary feature $\phi_{il}(l)$ is "on" when l_i is close to location l'_i.

Our corresponding unary parameters are $\mu_{il'}(x) = \sum_{t=1}^{T} K_i^t(x)\phi_{il'}(l^t) / \sum_{t=1}^{T} K_i^t(x)$. Intuitively, this is a weighted sum of labeled joint locations in the training set, with smoothing and robustness to labeling error by incorporating a neighborhood of locations defined by ω. When the weights are uniform, $\mu_{il'}(x)$ is simply a smoothed empirical average of joint locations in the training set. This type of uniform-weighted location prior is key to the success of the best results to date [3]. We will also refer to this as a "global" location prior. A well-weighted $\mu_{il'}(x)$, on the other hand, has the potential to be much more informative than a global location prior because it can adapt to the appearance of test image x. Figure 3 shows examples of a global location prior and a particular image's adaptive location prior. We can write the unary terms as:

$$\exp\Big(-\frac{1}{2}||\Sigma_i^{-1/2}(\phi_i - \mu_i)||^2\Big) \quad (9)$$

$$= \exp\Big(-\frac{1}{2}\Big(\phi_i^T\Sigma_i^{-1}\phi_i + \mu_i^T\Sigma^{-1}\mu_i\Big)\Big)\exp(\phi_i^T\Sigma_i^{-1}\mu_i) \quad (10)$$

$$\propto \exp(\phi_i^T\Sigma_i^{-1}\mu_i) \quad (11)$$

because, for a particular image x, both $\phi_i^T\Sigma_i^{-1}\phi_i$ and $\mu_i^T\Sigma_i^{-1}\mu_i$ are constant, and can be folded into the normalization factor of the overall probability distribution. Thus we see that the unary term can be written in the form of Equation 2 taking $\psi_i = \phi_i\Sigma^{-1}$, allowing us to use the usual efficient PS inference methods.

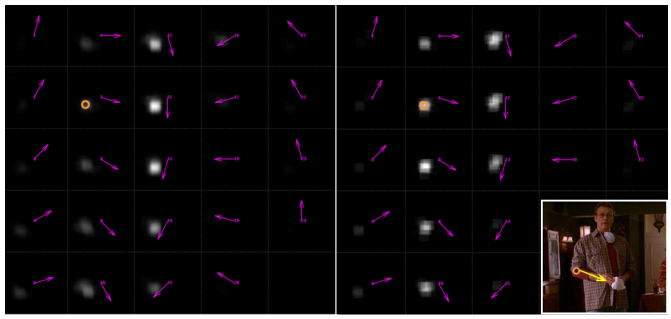

Figure 3. **Left**: Global location prior learned with uniform weights for the right lower arm, *i.e.* smoothed empirical average of joint locations in the training set. Each tile shows all image coordinates for a single discretized angle, whose orientation is indicated by the magenta vector. We see that the mode lower arm position is pointing straight down. **Right:** Location prior parameter estimated adaptively for the query image shown in inset. The adaptive location prior has considerably more mass in the correct location than the global prior. The correct location is marked by an orange circle in the left and right plots, which corresponds to the location/direction of the vector in the inset image.

4. Kernels for human pose estimation

Recent advances in PS performance have come from either improving local appearance models ([1, 3]), iteratively constraining the state space ([8, 9]), and/or adding pairwise terms—the model in [9] adds an edge between the left and right arms which encodes a repulsive force between the limbs. In the APS model on the other hand, we shift the focus from improving appearance or adding pairwise terms to instead finding good training examples via the kernels $K_i^t(x)$ and $K_{ij}^t(x)$, which can lead to query-specific improvements in unary and pairwise terms.

One natural choice for an ideal kernel is one that reflects the true similarity between the groundtruth pose of the query x and the groundtruth pose of training example t, for example $K_{i,gt}^t(x) = \exp(-||l_i^x - l_i^t||_2^2/\sigma_{i,gt}^2)$ for part i. Results in Section 5 show that using such an oracle kernel within the APS framework significantly beats any known method (going from 85.9% to 92.3% average part detection accuracy), as one might expect. We want our constructed kernel to be as close as possible to $K_{i,gt}^t$, but without access to l^x. Thus we require a representation that is independent of the tremendous amount of variation between image x and model x^t: differences in skin and clothing color, lighting conditions, background clutter, and deformations due to articulation and projective distortion. These require-

ments motivate our reliance on shape information as a robust indicator of pose similarity. We found that kernels based on dense appearance information such as HoG descriptors failed to capture the right information, most likely due to the large amount of clutter. A similar conclusion can be reached from the pose-retrieval experiments in [9].

In the construction of our kernel, we specifically focus on correct retrieval of upper and lower arms, since these are the most challenging parts to detect, and where almost all variation occurs after the initial localization.

4.1. Shape-based pose kernel

Consider a query image x which we wish to compare to a single training instance (x^t, l^t). To handle minor deformations, we expand (x^t, l^t) into a set of examples, all generated from example t by a set affine transformations $a \in \mathcal{A}$ varying scale and location of points: $(x^t, l^t) \mapsto \{(x_a^t, l_a^t)\}$ (Figure 4h).

To compute our kernel, we first filter the set of affine transforms using a quick, coarse region support distance $\mathbf{d}_{region}$ to get a shortlist of plausible affine transformations, $\mathcal{A}'$. We then use a distance based on contour information, $\mathbf{d}_{contour}$, to define our kernel value. Define $\mathbf{d}^{\star}_{contour}(x,t) = \min_{a\in\mathcal{A}'} \mathbf{d}_{contour}(x,(x_a^t,l_a^t))$ to be the best-matched affine transformation in the shortlist, based on our contour distance. Then our kernel similarity value between image x and example t is

$$K^t(x) = \exp\Big(-\mathbf{d}^{\star}_{contour}(x,t))/\sigma_K^2\Big) \qquad (12)$$

We chose $\sigma_K^2 = 40$ via cross-validation. We next explain $\mathbf{d}_{region}$ and $\mathbf{d}_{contour}$. Figure 4 illustrates the concepts.

Contour distance $\mathbf{d}_{contour}(x,(x_a^t,l_a^t))$: Salient contours in each image are extracted using the Probability of boundary detector (Pb) [14]. We discard contours which contain less than 15 points and are left with a set C^x for the query image and affinely-mapped contours C_a^t for the test image. We enforce rough figure/ground consistency by further removing contour points in C^x that are inconsistent with the foreground hypothesized by the affinely-mapped groundtruth l_a^t[3]. The remaining contour sets are lists of points with corresponding orientations discretized into 8 angular bins: $C = [c_1 \dots c_{|C|}]$, $c_i = [c_{ix}\ c_{iy}\ c_{i\theta}]^T$. We build histograms over orientations $h_\theta(C) \equiv h_\theta(C;x,y,r) = \sum_{c\in C} \mathbf{1}(c_\theta = \theta)\cdot\mathbf{1}(||[c_x;c_y]-[x;y]||_2 < r)$, placed at different coordinates (x,y) and over varying radii of support r. In practice we place 18 histograms spaced uniformly along the affinely-transformed groundtruth arm axes at 2 different radii, yielding 36 histograms with 8 bins each. We define

[3]In detail, if the contour points are a distance 0.25 times the image width away from the groundtruth line segment inferred from l_a^t, they are discarded.

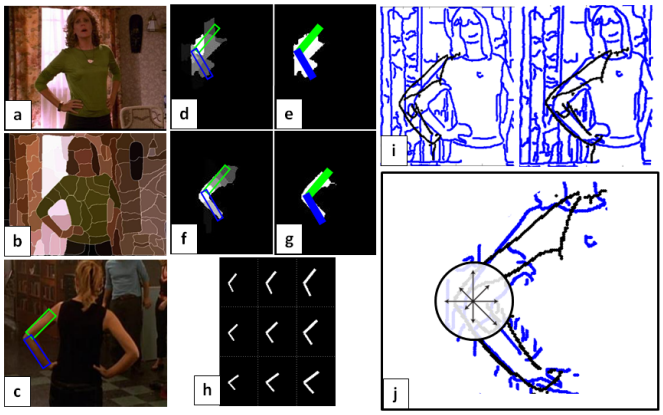

Figure 4. Construction of a shape based kernel; see Section 4.1 for details.**(a, b)** Test image x and its superpixelization. **(c)** Training example x^t with labeled pose l_{llarm} and l_{luarm} shown. We render the groundtruth arm with various affine transformations to produce binary masks M_a^t shown in **(h)**. We then count how much of the mask M_a^t overlaps each superpixel—**(d)** shows counts of a bad affine alignment of the groundtruth, **(f)** shows counts of good alignment. The counts are thresholded to produce corresponding binary masks M_a^x in **(e)** and **(g)**, and we use their intersection-over-union as a region distance. **(i,left)** shows the default alignment (no affine transform) of the groundtruth contours (black) with the test image contours (blue), and **(i,right)** shows a good candidate. In **(j)** we show a placement of one histogram $h_\theta(\cdot;x,y,r)$ at a particular location $[x;y]$ and radius r.

our contour distance to be

$$\mathbf{d}_{contour}(x,(x_a^t,l_a^t)) = \sum_{x,y,r} \chi^2(h_\theta(C^x), h_\theta(C_a^t)).^4 \qquad (13)$$

Figure 4j depicts an example histogram at a particular location and radius.

Region distance $\mathbf{d}_{region}(x,(x_a^t,l_a^t))$: This distance is inspired by template matching methods over superpixels, *e.g.* [2]. We convert the groundtruth into a binary template mask M_a^t by rendering the upper and lower arms as rectangles, with length and position given by groundtruth l_a^t, and width set to one-third of the length. Figure 4h shows binary masks for all warpings (minus translations) of the groundtruth that we use.

Superpixels are an over-segmentation of the image into perceptually coherent regions, as in Figure 4b. We use publicly available code from [18], set to obtain about 125 superpixels per image. If at least 10% of the limb mask M_a^t is contained in a superpixel, that superpixel is considered as supporting the groundtruth hypothesis. The union of all such supporting superpixels yield a binary mask M_a^x. Figure 4d,f show superpixel counts and Figure 4e,g show supporting binary masks M_a^x. We score how well the groundtruth mask and superpixel mask agree using the intersection-over-union measure to obtain our region dis-

[4]The χ^2-distance measures similarity between histograms: $\chi^2(h_1,h_2) = \frac{1}{2}\sum_{b=1}^{\#bins}(h_1(b)-h_2(b))^2/(h_1(b)+h_2(b))$.

tance:

$$\mathbf{d}_{region}(x,(x_a^t,l_a^t)) = \frac{\sum_{(r,c)} M_a^x(r,c) \cap M_a^t(r,c)}{\sum_{(r,c)} M_a^x(r,c) \cup M_a^t(r,c)} \quad (14)$$

where (r,c) index all (row, column) pairs in the masks.

In essence, this method proposes many different hypotheses for model t to match x, and these hypotheses must be consistent with coherent regions in the image to have a small distance. We obtain the shortlist $\mathcal{A}'$ by taking the top k closest matches according to $\mathbf{d}_{region}$. In experiments, we fix $k = 30$.

Discussion: Distances $\mathbf{d}_{region}$ and $\mathbf{d}_{contour}$ each have strengths and weaknesses. $\mathbf{d}_{region}$ is robust to foreground and background clutter, but is coarse and does not discriminate well between matches. $\mathbf{d}_{contour}$, on the other hand, is susceptible to noise from extraneous contours in clutter, but works well to refine what is the best match from a small set of good choices. These tradeoffs motivate our use of these distances: $\mathbf{d}_{region}$ is used as a quick, coarse filtering step that keeps only a shortlist of reasonable candidates. We then use the more discriminative $\mathbf{d}_{contour}$ for a final distance.

5. Experiments

We apply various of PS and APS models to baselines and previous work on two challenging upper-body human pose estimation datasets.

5.1. Implementation Details

Unary potentials: Individual part detectors were learned separately for all 6 parts: head, torso, upper and lower arms. For each part, we learn a Gentleboost classifier [11] on Histogram of Gradient (HoG)-based features [6][5]. We learn a discriminative weight for each part detector to determine how to balance the detector output with the jointly-learned means and covariances of the parts discriminatively by maximizing the conditional likelihood of the training set. We consider these unary terms part of the fixed, non-adaptive portion of our model, p_0 (see Section 3.2).

Pairwise potentials: As mentioned above, we learn Σ_{ij}'s discriminatively and keep them fixed. We set scalars Σ_i for the location prior (Section 3.2) using cross-validation data. All other parameters $\mu_{ij}(x)$ and $\mu_i(x)$ were estimated using our kernel regression framework for APS (Section 3), with respect to the kernel described in Section 4. We have a separate kernel for the left and right side which are exactly the same up to a horizontal flip (see Figure 1), and set $K_{head} = K_{torso} = \frac{1}{2}(K_{left} + K_{right})$.

Inference: We perform forward-backward sum-product message passing using FFT in $O(M \cdot |\mathcal{L}_i| \log |\mathcal{L}_i|)$ time. In practice $|\mathcal{L}_i| = 110 \times 122 \times 24 = 322,080$ possible labelings for each part.

This inference results in marginal distributions $p(L_i = l_i|x)$ for all parts. We make a hard decision of the location and direction of each limb by taking the max-marginal: $l_i^\star = \max_{l_i \in \mathcal{L}_i} p(L_i = l_i|x)$, and infer the skeleton pose configuration by assuming each part has length set *a priori* as the average part length in the training set.

Data and code for our experiments are available at:
http://www.vision.grasp.upenn.edu/video

[5] We train detectors using a dataset of hand-labeled body parts (916 arms, 1,386 torsos and heads) from TV shows and a movie (Lost, Friends and Good Will Hunting), available on our website.

5.2. Datasets

Buffy. We use the dataset described in [8], 400 frames from Buffy the Vampire Slayer, 100 images from 4 episode. Other results reported on this dataset [8, 9, 1, 3] test on a subset of 235 frames that were correctly localized (within 50% overlap of the groundtruth) using a detector from [8].

It is important in our system that test examples have similar training examples to use to locally estimate parameters. This is an issue with the standard test/train protocol because certain poses in this dataset are rather sparse: In the whole dataset there are only 4 shots of people with arms folded, with none in episode 2, and only 2 shots of people with arms raised above their heads. In light of this, we use 3 out of 4 episodes for training, and test on the remaining episode. We do this 4 times, each episode taking its turn as the test data. To facilitate comparison to previous work, we report average test results on the standard test set of 235 frames, but we are assured that no (test,train) image pair comes from the same episode.

ETHZ PASCAL Stickmen. This more challenging test set of real-world images is a subset of the PASCAL VOC 2008 [4]. We use all 400 images from Buffy as a training set of exemplars, keeping all parameters fixed from the experiments on Buffy, and report results on all 360 images in this test set.

5.3. Methods

We compare the following variations of the PS and APS framework and a template match baseline to previous work. Quantitative results are shown in Table 1 and Figure 5.

PS: A baseline pictorial structure model trained discriminatively as described in Section 5.1.

PS+global lp: PS model with an additional global location prior unary potential, described in Section 3.2 and shown in Figure 3/left.

APS: Variations of the APS models which include either kernel estimation of the means (μ), kernel estimation of the location prior (lp), or both (μ,lp). All variations use the kernel described in Section 4.

sparse APS: This is the APS(μ,lp) model along with the exemplar selection explained in Section 3.2.

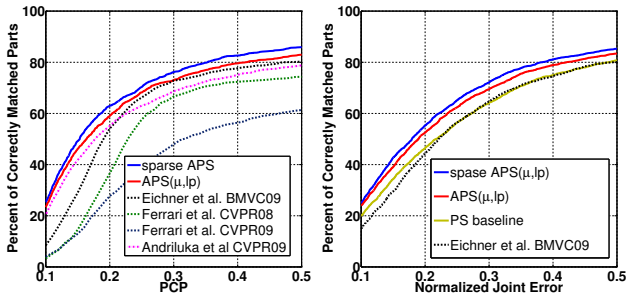

Figure 5. Results on Buffy. **Left:** PCP curves of our method versus previous work [8, 3, 9]. **Right:** Normalized joint error.

Normalized Joint Error (NJE), Buffy [8]						totals	
	sho.	elb.	wrists	torso	head	NJE	PCP
sparse APS	.132	**.218**	**.659**	.080	.165	**.238**	**85.9**
APS(μ,lp)	**.130**	.224	.721	**.079**	.163	.250	83.5
APS(μ)	.135	.231	.728	.082	.167	.257	83.3
APS(lp)	.140	.239	.748	.080	**.158**	.267	83.3
PS+glbl lp	.138	.241	.782	.082	.162	.272	83.3
template	.150	.251	.798	.112	.217	.294	83.0
PS baseline	.175	.251	.775	.110	.220	.298	81.5
Andril. [1]	-	-	-	-	-	-	78.8
Eichner [3]	.174	.304	.731	.132	.177	.307	80.1
APS+K_{gt}	.128	.178	.351	.086	.190	.179	92.3

Normalized Joint Error (NJE), PASCAL Stickmen [3]						totals	
	sho.	elb.	wrists	torso	head	NJE	PCP
sparse APS	**.263**	**.334**	**.848**	.138	.432	**.353**	**79.0**
PS baseline	.290	.358	.896	.156	.482	.386	75.7
Eichner [3]	.304	.401	1.010	**.118**	**.181**	.416	72.3

Table 1. Results of different methods on Buffy and PASCAL Stickmen. Numbers reported are a trimmed average normalized joint error, throwing out the worst 5% of matches for each method. PCP totals are computed using the publicly available code from [3].

APS+K_{gt}: This is the APS(μ,lp) model, but using an oracle kernel defined on the left side (right is symmetric) by $K^t_{gt/left}(x) = \exp\left(-||l^x_{larm} - l^t_{larm}||^2_2/\sigma^2_{gt}\right)$ which measures the distance between groundtruth (gt) arm locations in the test and training examples. We choose σ_{gt} optimally, and found that the best σ_{gt} gave significant weight to about 10 to 20 nearest training examples for each test example. Using a nearest-neighbor type oracle—taking the top k closest matches and giving them equal weight to learn adaptive parameters—did not perform as well. This method serves as a realistic upper bound on how well our method could perform.

template:. We can make use of the affine transform $a^\star$ from our kernel construction (Section 4.1), which is the best determined alignment of the groundtruth to the test example. Let $l^t_{a^\star}$ be the best affinely-mapped groundtruth found of training example t. Then the *template* method guesses a configuration $l^{template} = \sum_{t=1}^{T} K^t(x) l^t_{a^\star}$, a weighted sum of template matches from all training poses.

Evaluation measures: In previous work there has been some discrepancy in evaluation measures—[1] uses a stricter criterion than [3] in defining a limb as correctly matched. Thankfully, the authors of [3, 1] have provided their predictions and/or evaluation code publicly available, allowing us to compare performance accurately. In [3], a part is considered matched if the distance from the groundtruth part endpoints is less than some fraction of the length of the groundtruth part. By varying this fraction a curve of matching thresholds versus percentage of correct parts (referred to as *PCP*) can be generated—Figure 5/left.

We also report Euclidean distance to groundtruth endpoints, divided by the length of the groundtruth segments. We refer to this as *Normalized Joint Error*, and obtain a curve in Figure 5/right by varying a threshold on this value, and report average results in Table 1. Qualitative results are shown in Figure 6. More qualitative results are included in the supplemental material, and on our website [6].

5.4. Discussion

There are several interesting trends in the results (Figure 5 and Table 1). First, all variations of our APS are better than the state-of-the-art. Using our simple exemplar selection strategy results in selecting only 16% of the training examples, and gives a significant boost in performance. Figure 7 shows the top sparse exemplars. In variations of APS, estimating location priors and means both improve results.

Our basic PS model performs comparably to the previous state-of-the-art [3, 1] on this dataset in both measures of performance. This may be because our part detectors were trained with a large, outside training set using more powerful features (HoG) and classifier (Gentleboost)—similar to [1], except trained discriminatively—whereas previous works [8, 9, 3] only use linear filters on edge maps (at least in their first inference pass).

Using our kernel construction method as a form of template matching also works well—close to the basic PS model. This suggests that it is feasible to apply more sophisticated shape-matching and exemplar-distance based methods to the problem of articulated pose estimation. In addition, the huge gains in performance using an oracle kernel suggest that better kernel design is a worthwhile endeavor. Finally, our significant improvement when applying our method to a new environment (i.e., using exemplars from Buffy and applying them to the real-world photographs in PASCAL) makes a strong case for the generalization capabilities of APS.

6. Conclusion

We have presented the Adaptive Pictorial Structures framework which combines the flexibility of non-

[6] http://www.vision.grasp.upenn.edu/video/

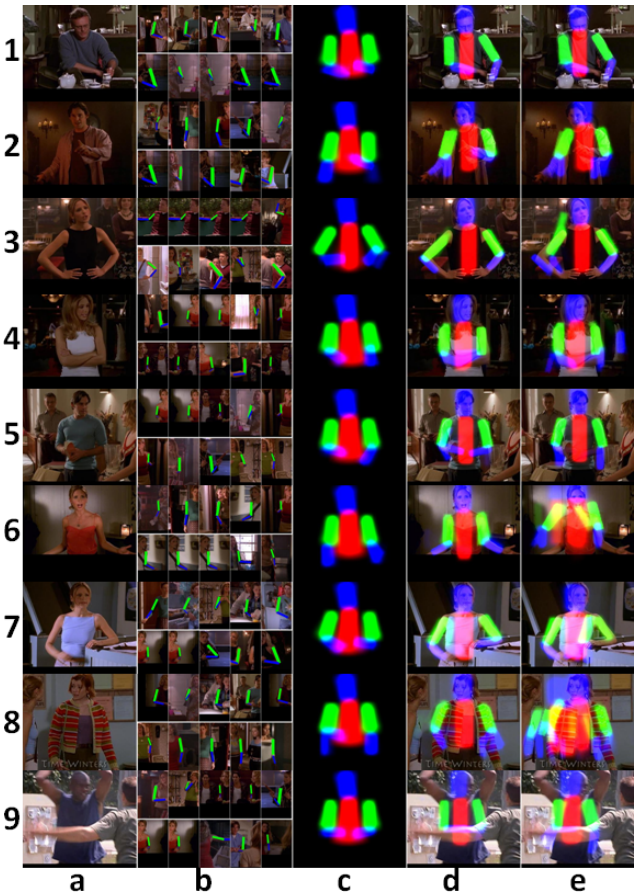

Figure 6. **(a)** Test image **(b)** Top 5 nearest neighbors retrieved via our kernel function for the left (top) and right (bottom) half-poses. **(c)** Belief in the APS model *before* running inference, *i.e*. based solely on image adaptive means and location prior. **(d)** Marginals of APS(μ,lp) model. **(e)** Marginals of baseline PS model. Rows 2-6 show examples where our model outperformed the baseline. Rows 8-9 show failures—sweater lines confusing shape distances in row 8; occluding arm confusion in row 9.

Figure 7. Top sparse exemplars selected, in order.

parametric methods with the tractability of PS models. The keys to success of our method are (1) data-dependent unary *and* pairwise terms, which allow our model to adapt parameters to each test query and (2) learning a sparse exemplar set. The framework is modular, and one can plug in kernel estimates of parameters into existing PS implementations. We develop an effective shape-based kernel, and raise our method's performance to state-of-the-art in two challenging upper-body pose datasets.

Acknowledgements

This work was supported by NSF grant 0803538 and a grant from Google.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. CVPR*, 2009. 1, 2, 4, 6, 7

[2] T. Cour and J. Shi. Recognizing objects by piecing together the segmentation puzzle. In *Proc. CVPR*, 2007. 5

[3] M. Eichner, V. Ferrari, and S. Zurich. Better appearance models for pictorial structures. In *Proc. BMVC*, 2009. 2, 4, 6, 7

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL VOC2009. 2, 6

[5] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005. 1, 2, 3

[6] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. 2008. 1, 6

[7] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proc. CVPR*, 2005. 1

[8] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proc. CVPR*, 2008. 1, 2, 4, 6, 7

[9] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: retrieving people using their pose. In *Proc. CVPR*, 2009. 2, 4, 5, 6, 7

[10] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 100(22), 1973. 1, 2

[11] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The annals of statistics*, 28(2), 2000. 6

[12] T. Graepel, R. Herbrich, J. Shawe-Taylor, and R. Holloway. Generalisation error bounds for sparse linear classifiers. In *In Proc. COLT*, 2000. 3

[13] X. Lan and D. Huttenlocher. Beyond trees: Common-factor models for 2d human pose recovery. In *ICCV*, 2005. 1, 2

[14] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 2004. 5

[15] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, 2004. 2

[16] D. Ramanan. Learning to parse images of articulated bodies. In *NIPS*, 2006. 2

[17] D. Ramanan and C. Sminchisescu. Training deformable models for localization. In *CVPR*, 2006. 1, 2, 4

[18] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. ICCV*, 2003. 5

[19] L. Sigal and M. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Proc. CVPR*, 2004. 1

[20] M. Tipping. Sparse Bayesian Learning and the Relevance Vector Machine. *JMLR*, 2001. 3