Cascaded Models for Articulated Pose Estimation

Benjamin Sapp, Alexander Toshev, and Ben Taskar

University of Pennsylvania,
Philadelphia, PA 19104 USA
{bensapp,toshev,taskar}@cis.upenn.edu

Abstract. We address the problem of articulated human pose estimation by learning a coarse-to-fine cascade of pictorial structure models. While the fine-level state-space of poses of individual parts is too large to permit the use of rich appearance models, most possibilities can be ruled out by efficient structured models at a coarser scale. We propose to learn a sequence of structured models at different pose resolutions, where coarse models filter the pose space for the next level via their max-marginals. The cascade is trained to prune as much as possible while preserving true poses for the final level pictorial structure model. The final level uses much more expensive segmentation, contour and shape features in the model for the remaining filtered set of candidates. We evaluate our framework on the challenging Buffy and PASCAL human pose datasets, improving the state-of-the-art.

1 Introduction

Pictorial structure models [1] are a popular method for human body pose estimation [2,3,4,5,6]. The model is a Conditional Random Field over pose variables that characterizes local appearance properties of parts and geometric part-part interactions. The search over the joint pose space is linear time in the number of parts when the part-part dependencies form a tree. However, the individual part state-spaces are too large (typically hundreds of thousands of states) to allow complex appearance models be evaluated densely. Most appearance models are therefore simple linear filters on edges, color and location [2,4,5,6]. Similarly, because of quadratic state-space complexity, part-part relationships are typically restricted to be image-independent deformation costs that allow for convolution or distance transform tricks to speed up inference [2]. A common problem in such models is poor localization of parts that have weak appearance cues or are easily confused with background clutter (accuracy for lower arms in human figures is almost half of that for torso or head [6]). Localizing these elusive parts requires richer models of individual part shape and joint part-part appearance, including contour continuation and segmentation cues, which are prohibitive to compute densely.

In order to enable richer appearance models, we propose to learn a cascade of pictorial structures (CPS) of increasing pose resolution which progressively

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 406–420, 2010.

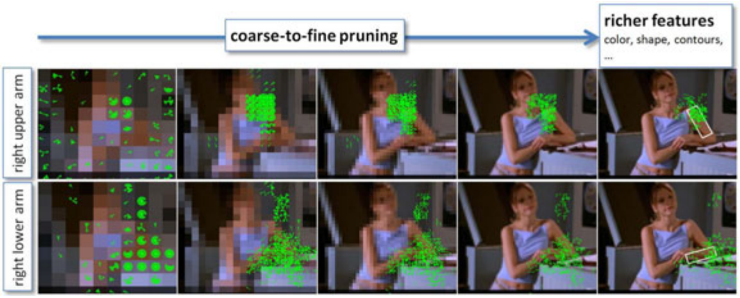

Fig. 1. Overview: A discriminative coarse-to-fine cascade of pictorial structures filters the pose space so that expressive and computationally expensive cues can be used in the final pictorial structure. Shown are 5 levels of our coarse-to-fine cascade for the right upper and lower arm parts. Green vectors represent position and angle of unpruned states, the downsampled images correspond to the dimensions of the resepective state space, and the white rectangles represent classification using our final model.

filter the pose state space. Conceptually, the idea is similar to the work on cascades for face detection [7,8], but the key difference is the use of structured models. Each level of the cascade at a given spatial/angular resolution refines the set of candidates from the previous level and then runs inference to determine which poses to filter out. For each part, the model selects poses with the largest max-marginal scores, subject to a computational budget. Unlike conventional pruning heuristics, where the possible part locations are identified using the output of a detector, models in our cascade use inference in simpler structured models to identify what to prune, taking into account global pose in filtering decisions. As a result, at the final level the CPS model has to deal with a much smaller hypotheses set which allows us to use a rich combination of features. In addition to the traditional part detectors and geometric features, we are able to incorporate object boundary continuity and smoothness, as well as shape features. The former features represent mid-level and bottom-up cues, while the latter capture shape information, which is complementary to the traditional HoG-based part models. The approach is illustrated in the overview Figure 1. We apply the presented CPS model combined with the richer set of features on the Buffy and PASCAL stickmen benchmark, improving the state-of-the-art on arm localization.

2 Related Work

The literature on human pose estimation is vast and varied in settings: applications range from highly-constrained MOCAP environments (*e.g.* [9]) to extremely articulated baseball players (*e.g.* [10]) to the recently popular "in the wild" datasets Buffy (from TV) and the PASCAL Stickmen (from amateur photographs) [5]. We focus our attention here on the work most similar in spirit to ours, namely, pictorial structures models. First proposed in [1], efficient inference methods focusing on tree-based models with quadratic deformation costs were introduced in [2]. Ramanan [4] proposed learning PS parameters discriminitively by maximizing conditional likelihood and introduced further improvements using iterative EM-like parsing [11]. Ferrari et al. [5,12] also prune the search space for computational efficiency and to avoid false positives. Our end goal is the same, but we adopt a more principled approach, expressing features on regions and locations and letting our system learn what to eliminate at run-time given the image.

For unstructured, binary classification, cascades of classifiers have been quite successful for reducing computation. Fleuret and Geman [7] propose a coarse-to-fine sequence of binary tests to detect the presence and pose of objects in an image. The learned sequence of tests is trained to minimize expected computational cost. The extremely popular Viola-Jones classifier [8] implements a cascade of boosting ensembles, with earlier stages using fewer features to quickly reject large portions of the state space.

Our cascade model is inspired by these binary classification cascades, and is based on the structured prediction cascades framework [13]. In natural language parsing, several works [14,15] use a coarse-to-fine idea closely related to ours and [7]: the marginals of a simple context free grammar or dependency model are used to prune the parse chart for a more complex grammar.

Recently, Felzenszwalb et al. [16] proposed a cascade for a structured parts-based model. Their cascade works by early stopping while evaluating individual parts, if the combined part scores are less than fixed thresholds. While the form of this cascade can be posed in our more general framework (a cascade of models with an increasing number of parts), we differ from [16] in that our pruning is based on thresholds that adapt based on inference in each test example, and we explicitly learn parameters in order to prune safely and efficiently. In [7,8,16], the focus is on preserving established levels of accuracy while increasing speed. The focus in this paper is instead developing more complex models—previously infeasible due to the original intractable complexity—to improve state-of-the-art performance.

A different approach to reduce the intractable number of state hypotheses is to instead propose a small set of likely hypotheses based on bottom-up perceptual grouping principles [10,17]. Mori et al. [10] use bottom-up saliency cues, for example strength of supporting contours, to generate limb hypotheses. They then prune via hand-set rules based on part-pair geometry and color consistency. The shape, color and contour based features we use in our last cascade stage are inspired by such bottom-up processes. However, our cascade is solely a sequence of discriminatively-trained top-down models.

3 Framework

We first summarize the basic pictorial structure model and then describe the inference and learning in the cascaded pictorial structures.

Classical 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 standard PS model decomposes as a tree structure into unary potentials (also referred to as appearance terms) and pairwise terms 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 [4,2,5,12,6], 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, $|\mathcal{L}_i|$, the number of possible locations in the image times the number of predefined discretized angles. For example, standard PS implementations typically model the state space of each part in a roughly 100×100 grid for $l_{ix} \times l_{iy}$, with 24 different possible values of angles, yielding $|\mathcal{L}_i| = 100 \times 100 \times 24 = 240,000$. The standard PS formulation (see [2]) is usually written in a log-quadratic form:

$$p(l|x) \propto \prod_{ij} \exp(-\frac{1}{2}||\Sigma_{ij}^{-1/2}(T_{ij}(l_i) - l_j - \mu_{ij})||_2^2) \times \prod_{i=1}^{M} \exp(\mu_i^T \phi_i(l_i, x)) \qquad (1)$$

The parameters of the model are μ_i, μ_{ij} and Σ_{ij}, and $\phi_i(l_i, x)$ are features of the (image) data x at location/angle l_i. The affine mapping T_{ij} transforms the part coordinates into a relative reference frame. 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 $\phi_{ij}(l) = T_{ij}(l_i) - l_j$. The unary terms pull the springs toward locations l_i with higher scores $\mu_i^T \phi_i(l_i, x)$ which are more likely to be a location for part i.

This form of the pairwise potentials allows inference to be performed faster than $O(|\mathcal{L}_i|^2)$: MAP estimates $\arg\max_l p(l|x)$ 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 $O(|\mathcal{L}_i| \log |\mathcal{L}_i|)$ [2].

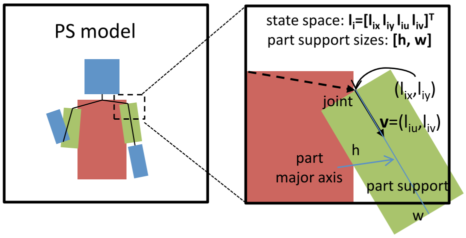
PS model
state space: $l_i=[l_{ix}\ l_{iy}\ l_{iu}\ l_{iv}]^T$
part support sizes: [h, w]
(l_{ix},l_{iy})
joint
$\mathbf{v}=(l_{iu},l_{iv})$
h
part
major axis
part support
w

Fig. 2. Basic PS model with state l_i for a part L_i

While fast to compute and intuitive from a spring-model perspective, this model has two significant limitations. One, the pairwise costs are unimodal Gaussians, which cannot capture the true multimodal interactions between pairs of body parts. Two, the pairwise terms are only a function of the geometry of the state configuration, and are oblivious to the image cues, for example, appearance similarity or contour continuity of the a pair of parts.

We choose instead to model part configurations as a general log-linear Conditional Random Field over pairwise and unary terms:

$$p(l|x) \propto \exp\left[\sum_{ij} \theta_{ij}^T \phi_{ij}(l_i, l_j, x) + \sum_i \theta_i^T \phi_i(l_i, x)\right] = e^{\theta^T \phi(l,x)}. \qquad (2)$$

The parameters of our model are the pairwise and unary weight vectors θ_{ij} and θ_i corresponding to the pairwise and unary feature vectors $\phi_{ij}(l_i, l_j, x)$ and $\phi_i(l_i, x)$. For brevity, we stack all the parameters and features into vectors using notation $\theta^T \phi(l, x)$. The key differences with the classical PS model are that (1) our pairwise costs allow data-dependent terms, and (2) we do not constrain our parameters to fit any parametric distribution such as a Gaussian. For example, we can express the pairwise features used in the classical model as $l_i \cdot l_i$, $l_j \cdot l_j$ and $l_i \cdot l_j$ without requiring that their corresponding weights can be combined into a positive semi-definite covariance matrix.

In this general form, inference can not be performed efficiently with distance transforms or convolution, and we rely on standard $O(|\mathcal{L}_i|^2)$ dynamic programming techniques to compute the MAP assignment or part posteriors. Many highly-effective pairwise features one might design would be intractable to compute in this manner for a reasonably-sized state space—for example an 100×100 image with a part angle discretization of 24 bins yields $|\mathcal{L}_i|^2 = 57.6$ billion part-part hypotheses.

In the next section, we describe how we circumvent this issue via a cascade of models which aggressively prune the state space at each stage typically without discarding the correct sequence. After the state space is pruned, we are left with a small enough number of states to be able to incorporate powerful data-dependent pairwise and unary features into our model.

Structured Prediction Cascades

The recently introduced Structured Prediction Cascade framework [13] provides a principled way to prune the state space of a structured prediction problem via a sequence of increasingly complex models. There are many possible ways of defining a sequence of increasingly complex models. In [13] the authors introduce higher-order cliques into their models in successive stages (first unary, then pairwise, ternary, etc.). Another option is to start with simple but computationally efficient features, and add more complex features downstream as the number of states decreases. Yet another option is to geometrically coarsen the original state space and successively prune and refine. We use a coarse-to-fine state space approach with simple features until we are at a reasonably fine enough state space resolution and left with few enough states that we can introduce more complex features. We start with a severely coarsened state space and use standard pictorial structures unary detector scores and geometric features to perform quick exhaustive inference on the coarse state space.

More specifically, each level of the cascade uses inference to identify which states to prune away and the next level refines the spatial/angular resolution on the unpruned states. The key ingredient to the cascade framework is that states are pruned using *max-marginal* scores, computed using dynamic programming techniques. For brevity of notation, define the score of a joint part state l as $\theta_x(l)$ and the max-marginal score of a part state as follows:

$$\theta_x(l) = \theta^T \phi(l,x) = \sum_{ij} \theta_{ij}^T \phi_{ij}(l_i, l_j, x) + \sum_i \theta_i^T \phi_i(l_i, x) \tag{3}$$

$$\theta_x^\star(l_i) = \max_{l' \in L} \quad \{\theta_x(l') \ : \ l'_i = l_i\} \tag{4}$$

In words, the max-marginal for location/angle l_i is the score of the best sequence which constrains $L_i = l_i$. In a pictorial structure model, this corresponds to fixing limb i at location l_i, and determining the highest scoring configuration of other part locations and angles under this constraint. A part could have weak individual image evidence of being at location l_i but still have a high max-marginal score if the rest of the model believes this is a likely location. Similarly, we denote the MAP assignment score as $\theta_x^\star = \max_{l \in L} \theta_x(l)$, the unconstrained best configuration of all parts.

When learning a cascade, we have two competing objectives that we must trade off, accuracy and efficiency: we want to minimize the number of errors incurred by each level of the cascade and maximize the number of filtered max marginals. A natural strategy is to prune away the lowest ranked states based on max-marginal scores. Instead, [13] prune the states whose max-marginal score is lower than an data-specific threshold t_x: l_i is pruned if $\theta_x^\star(l_i) < t_x$. This threshold is defined as a convex combination of the MAP assignment score and the mean max-marginal score, meant to approximate a percentile threshold:

$$t_x(\theta, \alpha) = \alpha \theta_x^\star + (1-\alpha) \frac{1}{M} \sum_{i=1}^{M} \frac{1}{|\mathcal{L}_i|} \sum_{l_i \in \mathcal{L}_i} \theta_x^\star(l_i),$$

where $\alpha \in [0,1]$ is a parameter to be chosen that determines how aggressively to prune. When $\alpha = 1$, only the best state is kept, which is equivalent to finding the MAP assignment. When $\alpha = 0$ approximately half of the states are pruned (if the median of max-marginals is equal to the mean). The advantage of using $t_x(\theta, \alpha)$ is that it is convex in θ, and leads to a convex formulation for parameter estimation that trades off the proportion of incorrectly pruned states with the proportion of unpruned states. Note that α controls efficiency, so we focus on learning the parameters θ that minimize the number of errors for a given filtering level α. The learning formulation uses a simple fact about max-marginals and the definition of $t_x(\theta, \alpha)$ to get a handle on errors of the cascade: if $\theta_x(l) > t_x(\theta, \alpha)$, then for all i, $\theta_x^\star(l_i) > t_x(\theta, \alpha)$, so no part state of l is pruned. Given an example (x, l), this condition $\theta_x(l) > t_x(\theta, \alpha)$ is sufficient to ensure that no correct part is pruned.

To learn one level of the structured cascade model θ for a fixed α, we try to minimize the number of correct states that are pruned on training data by solving the following convex margin optimization problem given N training examples (x^n, l^n):

$$\min_{\theta} \quad \frac{\lambda}{2}||\theta||^2 + \frac{1}{N}\sum_{n=1}^{N} H(\theta; x^n, l^n), \tag{5}$$

where H is a hinge upper bound $H(\theta; x, l) = \max\{0, 1 + t_x(\theta, \alpha) - \theta_x(l)\}$. The upper-bound H is a hinge loss measuring the margin between the filter threshold $t_{x^n}(\theta, \alpha)$ and the score of the truth $\theta^T \phi(l^n, x^n)$; the loss is zero if the truth scores above the threshold by margin 1. We solve (5) using stochastic sub-gradient descent. Given an example (x, l), we apply the following update if $H(\theta; x, l)$ (and the sub-gradient) is non-zero:

$$\theta' \leftarrow \theta + \eta \left(-\lambda\theta + \phi(l, x) - \alpha\phi(l^\star, x) - (1-\alpha)\frac{1}{M}\sum_i \frac{1}{|\mathcal{L}_i|}\sum_{l_i \in \mathcal{L}_i} \phi(l^\star(l_i), x) \right).$$

Above, η is a learning rate parameter, $l^\star = \arg\max_{l'} \theta_x(l')$ is the highest scoring assignment and $l^\star(l_i) = \arg\max_{l':l'_i = l_i} \theta_x(l')$ are highest scoring assignments constrained to l_i for part i. The key distinguishing feature of this update as compared to structured perceptron is that it subtracts features included in all max-marginal assignments $l^\star(l_i)$[1].

The stages of the cascade are learned sequentially, from coarse to fine, and each has a different θ and $\mathcal{L}_i$ for each part, as well as α. The states of the next level are simply refined versions of the states that have not been pruned. We describe the refinement structure of the cascade in Section 5. In the end of a coarse-to-fine cascade we are left with a small, sparse set of states that typically contains the groundtruth states or states relatively close to them—in practice we are left with around 500 states per part, and 95% of the time we retain a state the is close enough to be considered a match (see Table 2). At this point we have the freedom to add a variety of complex unary and pairwise part interaction features involving geometry, appearance, and compatibility with perceptual grouping principles which we describe in Section 4.

Why not just detector-based pruning? A naive approach used in a variety of applications is to simply subsample states by thresholding outputs of part or sparse feature detectors, possibly combined with non-max suppression. Our approach, based on pruning on max-marginal values in a first-order model, is more sophisticated: for articulated parts-based models, strong evidence from other parts can keep a part which has weak individual evidence, and would be pruned using only detection scores. The failure of prefiltering part locations in human pose estimation is also noted by [6], and serves as the primary justification

[1] Note that because (5) is λ-strongly convex, if we chose $\eta_t = 1/(\lambda t)$ and add a projection step to keep θ in a closed set, the update would correspond to the Pegasos update with convergence guarantees of $\tilde{O}(1/\epsilon)$ iterations for ϵ-accurate solutions [18]. In our experiments, we found the projection step made no difference and used only 2 passes over the data, with η fixed.

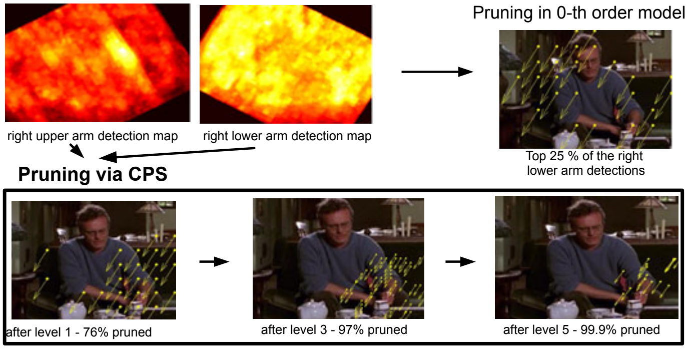
Pruning in 0-th order model
right upper arm detection map
right lower arm detection map
Top 25 % of the right lower arm detections
Pruning via CPS
after level 1 - 76% pruned
after level 3 - 97% pruned
after level 5 - 99.9% pruned

Fig. 3. Upper right: Detector-based pruning by thresholding (for the lower right arm) yields many hypotheses far way from the true one. Lower row: The CPS, however, exploits global information to perform better pruning.

for their use of the dense classical PS. This is illustrated in Figure 3 on an example image from [5].

4 Features

The introduced CPS model allows us to capture appearance, geometry and shape information of parts and pairs of parts in the final level of the cascade—much richer than the standard geometric deformation costs and texture filters of previous PS models [2,4,5,6]. Each part is modeled as a rectangle anchored at the part joint with the major axis defined as the line segment between the joints (see Figure 2). For training and evaluation, our datasets have been annotated only with this part axis.

Shape: We express the shape of limbs via region and contour information. We use contour cues to capture the notion that limbs have a long smooth outline connecting and supporting both the upper and lower parts. Region information is used to express coarse global shape properties of each limb, attempting to express the fact the limbs are often supported by a roughly rectangular collection of regions—the same notion that drives the bottom-up hypothesis generation in [10,17].

Shape/Contour: We detect long smooth contours from sequences of image segmentation boundaries obtained via NCut [24]. We define a graph whose nodes are all boundaries between segments with edges linking touching boundaries. Each contour is a path in this graph (see Fig. 4, middle left). To reduce the number of possible paths, we restrict ourselves to all shortest paths. To quantify the smoothness of a contour, we compute an angle between each two touching segment boundaries[2]. The smoothness of a contour is quantified as the maximum

[2] This angle is computed as the angle between the lines fitted to the segment boundary ends, defined as one third of the boundary.

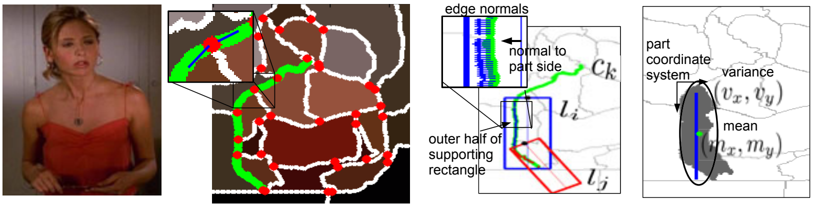

Fig. 4. Left: input image; Middle left: segmentation with segment boundaries and their touching points in red. Middle right: contour edges which support part l_i and have normals which do not deviate from the part axis normal by more than ω. Right: first and second order moments of the region lying under the major part axis.

angle between boundaries along this contour. Finally, we find among all shortest paths those whose length exceeds ℓ_{th} pixels and whose smoothness is less then s_{th} and denote them by $\{c_1, \dots c_m\}$.[3]

We can use the above contours to define features for each pair of lower and upper arms, which encode the notion that those two parts should share a long smooth contour, which is parallel and close to the part boundaries. For each arm part l_i and a contour c_k we can estimate the edges of c_k which lie inside one of the halves of the supporting rectangle of l_i and whose edge normals build an angle smaller than ω with the normal of the part axis (see Fig. 4, right). We denote the number of those edges by $q_{ik}(\omega)$. Intuitively, a contour supports a limb if it is mostly parallel and enclosed in one of the limb sides, i.e. the value $q_{ik}(\omega)$ is large for small angles ω. A pair of arm limbs l_i, l_j should have a high score if both parts are supported by a contour c_k, which can be expressed as the following two scores

$$\mathrm{cc}^{(1)}_{ijk}(\omega,\omega')=\frac{1}{2}\left(\frac{q_{ik}(\omega)}{h_i}+\frac{q_{jk}(\omega')}{h_j}\right) \quad \text{and} \quad \mathrm{cc}^{(2)}_{ijk}(\omega,\omega')=\min\left\{\frac{q_{ik}(\omega)}{h_i},\frac{q_{jk}(\omega')}{h_j}\right\}$$

where we normalize q_{ik} by the length of the limb h_i to ensure that the score is in $[0,1]$. The first score measures the overall support of the parts, while the second measures the minimum support. Hence, for l_i, l_j we can find the highest score among all contours, which expresses the highest degree of support which this pair of arms can receive from any of the image contours:

$$\mathrm{cc}^{(t)}_{ij}(\omega,\omega') = \max_{k\in\{1,\dots,m\}} \mathrm{cc}^{(t)}_{ijk}(\omega,\omega'), \quad \text{for} \quad t\in\{1,2\}$$

By varying the angles ω and ω' in a set of admissible angles Ω defining parallelism between the part and the contour, we obtain $|\Omega|^2$ contour features[4].

Shape/Region Moments: We compute the first and second order moments of the segments lying under the major part axis (see Fig. 4, right)[5] to coarsely

[3] We set $\ell_{\text{th}} = 60$ pixels, $s_{\text{th}} = 45°$ resulting in 15 to 30 contours per image.
[4] We set $\Omega = \{10°, 20°, 30°\}$, which results in 18 features for both scores.
[5] We select segments which cover at least 25% of the part axis.

express shape of limb hypotheses as a collection of segments, R_{l_i}. To achieve rotation and translation invariance, we compute the moments in the part coordinate system. We include convexity information $|conv(R_{l_i})|/|R_{l_i}|$, where $conv(\cdot)$ is the convex hull of a set of points, and $|R_{l_i}|$ is the number of points in the collection of segments. We also include the number of points on the convex hull, and the number of part axis points that pass through R_{l_i} to express continuity along the part axis.

Appearance/Texture: Following the edge-based representation used in [19], we model the appearance the body parts using Histogram of Gradient (HoG) descriptor. For each of the 6 body parts – head, torso, upper and lower arms – we learn an individual Gentleboost classifier [20] on the HoG features using the Limbs Annotated in Movies Dataset[6].

Appearance/Color: As opposed to HoG, color drastically varies between people. We use the same assumptions as [21] and build color models assuming a fixed location for the head and torso at run-time for each image. We train Adaboost classifiers using these pre-defined regions of positive and negative example pixels, represented as RGB, Lab, and HSV components. For a particular image, a 5-round Adaboost ensemble [22] is learned for each color model (head, torso) and reapplied to all the pixels in the image. A similar technique is also used by [23] to incorporate color. Features are computed as the mean score of each discrimintative color model on the pixels lying in the rectangle of the part.

We use similarity of appearance between lower and upper arms as features for the pairwise potentials of CPS. Precisely, we use the χ^2 distance between the color histograms of the pixels lying in the part support. The histograms are computed using minimum-variance quantization of the RGB color values of each image into 8 colors.

Geometry: The body part configuration is encoded in two set of features. The location (l_{ix}, l_{iy}) and orientation (l_{iu}, l_{iv}), included in the state of a part, are used added as absolute location prior features. We express the relative difference between part l_i its parent l_j in the coordinate frame of the parent part as $T_{ij}(l_i) - l_j$. Note we could introduce second-order terms to model a quadratic deformation cost akin to the classical PS, but we instead adopt more flexible binning or boosting of these features (see Section 5).

5 Implementation Details

Coarse-to-Fine Cascade. While our fine-level state space has size $80 \times 80 \times 24$, our first level cascade coarsens the state-space down to $10 \times 10 \times 12 = 1200$ states per part, which allows us to do exhaustive inference efficiently. We always train and prune with $\alpha = 0$, effectively throwing away half of the states at each stage. After pruning we double one of the dimensions (first angle, then the minimum of width or height) and continue (see Table 2). In the coarse-to-fine stages we only use standard PS features. HoG part detectors are run once over the original

[6] LAMDa is available at `http://vision.grasp.upenn.edu/video`

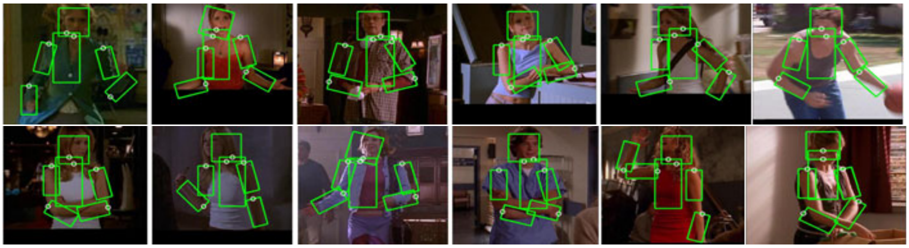

Fig. 5. Examples of correctly localized limbs under different conditions (low contrast, clutter) and poses (different positions of the arms, partial self occlusions)

state space, and their outputs are resized to for features in coarser state spaces. We also use the standard relative geometric cues as described in Sec. 4. We bin the values of each feature uniformly, which adds flexibility to the standard PS model—rather than learning a mean and covariance, multi-modal pairwise costs can be learned.

Sparse States, Rich Features. To obtain segments, we use NCut[24]. For the contour features we use 30 segments and for region moments – 125 segments. As can be seen in Table 2, the coarse-to-fine cascade leaves us with roughly 500 hypotheses per part. For these hypotheses, we generate all features mentioned in Sec. 4. For pairs of part hypotheses which are farther than 20% of the image dimensions from the mean connection location, features are not evaluated and an additional feature expressing this condition is added to the feature set. We concatenate all unary and pairwise features for part-pairs into a feature vector and learn boosting ensembles which give us our pairwise clique potentials[7]. This method of learning clique potentials has several advantages over stochastic subgradient learning: it is faster to train, can determine better thresholds on features than uniform binning, and can combine different features in a tree to learn complex, non-linear interactions.

6 Experiments

We evaluate our approach on the publicly available Buffy The Vampire Slayer v2.1 and PASCAL Stickmen datasets [21]. We use the upper body detection windows provided with the dataset as input to localize and scale normalize the images before running our experiments as in [21,5,6]. We use the usual 235 Buffy test images for testing as well as the 360 detected people from PASCAL stickmen. We use the remaining 513 images from Buffy for training and validation.

Evaluation Measures. The typical measure of performance on this dataset is a matching criteria based on both endpoints of each part (e.g., matching the elbow and the wrist correctly): A limb guess is correct if the limb endpoints are

[7] We use OpenCV's implementation of Gentleboost and boost on trees of depth 3, setting the optimal number of rounds via a hold-out set.

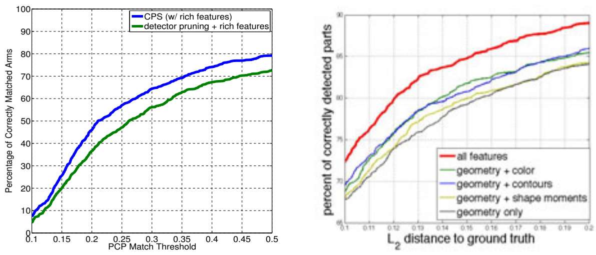

Fig. 6. Left: PCP curves of our cascade method versus a detection pruning approach, evaluated using PCP on arm parts (see text). **Right:** Analysis of incorporating individual types of features into the last stage of our system.

on average within r of the corresponding groundtruth segments, where r is a fraction of the groundtruth part length. By varying r, a performance curve is produced where the performance is measured in the percentage of correct parts (PCP) matched with respect to r.

Overall system performance. As shown in Table 1, we perform comparably with the state-of-the-art on all parts, improving over [25] on upper arms on both datasets and significantly outperforming earlier work. We also compare to a much simpler approach, inspired by [16] (detector pruning + rich features): We prune by thresholding each unary detection map individually to obtain the same number of states as in our final cascade level, and then apply our final model with rich features on these states. As can be seen in Figure 6/left, this baseline performs significantly worse than our method (performing about as well as a standard PS model as reported in [25]). This makes a strong case for using max-marginals (e.g., a global image-dependent quantity) for pruning, as well as learning how to prune safely and efficiently, rather than using static thresholds on individual part scores as in [16].

Our previous method [25] is the only other PS method which incorporates image information into the pairwise term of the model. However it is still an exhaustive inference method. Assuming all features have been pre-computed, inference in [25] takes an average of 3.2 seconds, whereas inference using the sparse set of states in the final stage of the cascade takes on average 0.285 seconds—a speedup of 11.2x[8].

In Figure 6/right we analyze which features are most effective, measured in L_2 distance to the groundtruth state, normalized by the groundtruth length of the part. We start only with the basic geometry and unary HoG detector features available to basic PS systems, and add different classes of features individually. Skin/torso color estimation gives a strong boost in performance, which is consistent with the large performance boost that the results in [21] obtained over their previous results [12]. Using contours instead of color is nearly as effective. The

[8] Run on an Intel Xeon E5450 3.00GHz CPU with an $80 \times 80 \times 24$ state space averaged over 20 trials. [25] uses MATLAB's optimized fft function for message passing.

Table 1. Comparison to other methods at $\text{PCP}_{0.5}$. See text for details. We perform comparably to state-of-the-art on all parts, improving on upper arms.

method	torso	head	upper arms	lower arms	total
Buffy					
Andriluka et al. [6]	90.7	95.5	79.3	41.2	73.5
Eichner et al. [21]	98.7	97.9	82.8	59.8	80.1
APS [25]	100	100	91.1	65.7	85.9
CPS (ours)	100	96.2	95.3	63.0	85.5
Detector pruning	99.6	87.3	90.0	55.3	79.6
PASCAL stickmen					
Eichner et al. [21]	97.22	88.60	73.75	41.53	69.31
APS [25]	100	98.0	83.9	54.0	79.0
CPS (ours)	100	90.0	87.1	49.4	77.2

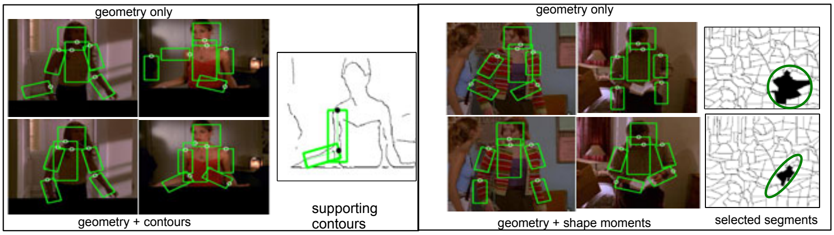

Fig. 7. Detections with geometry (top) and with additional cues (bottom). Left: contour features support arms along strong contours and avoid false positives along weak edges. Right: after overlaying the part hypothesis on the segmentation, the incorrect one does not select an elongated set of segments.

features combine to outperform any individual feature. Examples where different cues help are shown in Figure 7.

Coarse-to-fine Cascade Evaluation: In Table 2, we evaluate the drop in performance of our system after each successive stage of pruning. We report PCP scores of the best possible as-yet unpruned state left in the original space. We choose a tight $\text{PCP}_{0.2}$ threshold to get an accurate understanding whether we have lost well-localized limbs. As seen in Table 2, the drop in $\text{PCP}_{0.2}$ is small and linear, whereas the pruning of the state space is exponential—half of the states are pruned in the first stage. As a baseline, we evaluate the simple detector-based pruning described above. This leads to a significant loss of correct hypotheses, to which we attribute the poor end-system performance of this baseline (in Figure 6 and Table 1), even after adding richer features.

Future work: The addition of more powerful shape-based features could further improve performance. Additional levels of pruning could allow for (1) faster inference, (2) inferring with higher-order cliques to, e.g., express compatability between left and right arms or (3) incorporating additional variables into the state space—relative scale of parts to model foreshortening, or occlusion variables. Finally, our approach can be naturally extended to pose estimation in video where the cascaded models can be coarsened over space and time.

Table 2. For each level of the cascade we present the reduction of the size of the state space after pruning each stage and the quality of the retained hypotheses measured using $\mathrm{PCP}_{0.2}$. As a baseline, we compare to pruning the same number of states in the HoG detection map (see text).

cascade	state	# states in the		state space	$\mathrm{PCP}_{0.2}$
stage	dimensions	original	pruned	reduction	arms
		space	space	%	oracle
0	10x10x12	153600	1200	00.00	—
1	10x10x24	72968	1140	52.50	54
3	20x20x24	6704	642	95.64	51
5	40x40x24	2682	671	98.25	50
7	80x80x24	492	492	99.67	50
detection pruning	80x80x24	492	492	99.67	44

References

1. Fischler, M., Elschlager, R.: The representation and matching of pictorial structures. IEEE Transactions on Computers 100, 67–92 (1973)
2. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. IJCV 61, 55–79 (2005)
3. Fergus, R., Perona, P., Zisserman, A.: A sparse object category model for efficient learning and exhaustive recognition. In: Proc. CVPR (2005)
4. Ramanan, D., Sminchisescu, C.: Training deformable models for localization. In: CVPR, pp. 206–213 (2006)
5. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Progressive search space reduction for human pose estimation. In: Proc. CVPR (2008)
6. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: Proc. CVPR (2009)
7. Fleuret, G., Geman, D.: Coarse-to-Fine Face Detection. IJCV 41, 85–107 (2001)
8. Viola, P., Jones, M.: Robust real-time object detection. IJCV 57, 137–154 (2002)
9. Lan, X., Huttenlocher, D.: Beyond trees: Common-factor models for 2d human pose recovery. In: Proc. ICCV, pp. 470–477 (2005)
10. Mori, G., Ren, X., Efros, A., Malik, J.: Recovering human body configurations: Combining segmentation and recognition. In: CVPR (2004)
11. Ramanan, D.: Learning to parse images of articulated bodies. In: NIPS (2006)
12. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Pose search: retrieving people using their pose. In: Proc. CVPR (2009)
13. Weiss, D., Taskar, B.: Structured prediction cascades. In: Proc. AISTATS (2010)
14. Carreras, X., Collins, M., Koo, T.: TAG, dynamic programming, and the perceptron for efficient, feature-rich parsing. In: Proc. CoNLL (2008)
15. Petrov, S.: Coarse-to-Fine Natural Language Processing. PhD thesis, University of California at Bekeley (2009)
16. Felzenszwalb, P., Girshick, R., McAllester, D.: Cascade Object Detection with Deformable Part Models. In: Proc. CVPR (2010)
17. Srinivasan, P., Shi, J.: Bottom-up recognition and parsing of the human body. In: ICCV 2005, pp. 824–831. IEEE Computer Society, Los Alamitos (2007)
18. Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: Primal estimated sub-gradient SOlver for SVM. In: Proc. ICML (2007)

19. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part Based Models. In: PAMI (2008)
20. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. The Annals of Statistics 28, 337–374 (2000)
21. Eichner, M., Ferrari, V.: Better appearance models for pictorial structures. In: Proc. BMVC (2009)
22. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. JCSS 55, 119–139 (1997)
23. Ramanan, D., Forsyth, D., Zisserman, A.: Strike a pose: Tracking people by finding stylized poses. In: Proc. CVPR, vol. 1, p. 271 (2005)
24. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: Proc. CVPR (2005)
25. Sapp, B., Jordan, C., Taskar, B.: Adaptive pose priors for pictorial structures. In: CVPR (2010)