Articulated pose estimation with flexible mixtures-of-parts

Yi Yang Deva Ramanan
Dept. of Computer Science, University of California, Irvine
{yyang8,dramanan}@ics.uci.edu

Abstract

We describe a method for human pose estimation in static images based on a novel representation of part models. Notably, we do not use articulated limb parts, but rather capture orientation with a mixture of templates for each part. We describe a general, flexible mixture model for capturing contextual co-occurrence relations between parts, augmenting standard spring models that encode spatial relations. We show that such relations can capture notions of local rigidity. When co-occurrence and spatial relations are tree-structured, our model can be efficiently optimized with dynamic programming. We present experimental results on standard benchmarks for pose estimation that indicate our approach is the state-of-the-art system for pose estimation, outperforming past work by 50% while being orders of magnitude faster.

1. Introduction

We examine the task of human pose estimation in static images. A working technology would immediately impact many key vision tasks such as image understanding and activity recognition. An influential approach is the pictorial structure framework [7, 12] which decomposes the appearance of objects into local part templates, together with geometric constraints on pairs of parts, often visualized as springs. When parts are parameterized by pixel location and orientation, the resulting structure can model articulation. This has been the dominant approach to human pose estimation. In contrast, traditional models for object recognition use parts parameterized solely by location, which simplifies both inference and learning. Such models have been shown to be very successful for object recognition [2, 9]. In this work, we introduce a novel unified representation for both models that produces state-of-the-art results for human pose estimation.

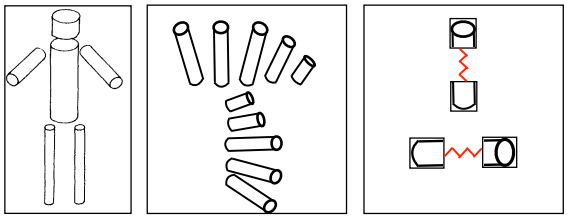

Figure 1: On the **left**, we show the classic articulated limb model of Marr and Nishihara [20]. In the **middle**, we show different orientation and foreshortening states of a limb, each of which is evaluated separately in classic articulated body models. On the **right**, we approximate these transformations with a mixture of non-oriented pictorial structures, in this case tuned to represent near-vertical and near-horizontal limbs.

Representations for articulated pose: Full-body pose estimation is difficult because of the many degrees of freedoms to be estimated. Moreover, limbs vary greatly in appearance due to changes in clothing and body shape, as well as changes in viewpoint manifested in in-plane orientations and foreshortening. These difficulties complicate inference since one must typically search images with a large number of rotated and foreshortened templates. We address these problems by introducing a novel but simple representation for modeling a family of affinely-warped templates: a mixture of non-oriented pictorial structures (Fig.1). We empirically demonstrate that such approximations can outperform explicitly articulated parts because mixture models can capture orientation-specific statistics of background features (Fig.2).

Representations for objects: Current object recognition systems are built on relatively simple structures encoding mixtures of star models defined over tens of parts [9], or implicitly-defined shape models built on hundreds of parts [19, 2]. In order to model the varied appearance of objects (due to deformation, viewpoint,etc.), we argue that one will need vocabularies of hundreds or thousands of parts, where only a subset are instanced at a time. We augment classic spring models with co-occurrence constraints that favor particular combinations of parts. Such constraints can capture notions of *local rigidity* – for example, two parts on the same limb should be constrained to have the same orientation state (Fig.1). We show that one can embed such constraints in a tree relational graph that preserves tractability. An open challenge is that of learning such complex repre-

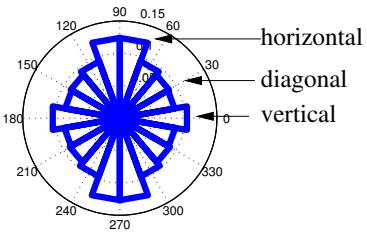

Figure 2: We plot the average HOG feature as a polar histogram over 18 gradient orientation channels as computed from the entire PASCAL 2010 dataset [6]. We see that on average, images contain more horizontal gradients than vertical gradients, and much stronger horizontal gradients as compared to diagonal gradients. This means that gradient statistics are not orientation invariant. In practical terms, we argue that it is easier to find diagonal limbs (as opposed to horizontal ones) because one is less likely to be confused by diagonal background clutter. Articulated limb models obtained by rotating a single template cannot exploit such orientation-specific cues. On the other hand, our mixture models are tuned to detect parts at particular orientations, and so can exploit such statistics.

sentations from data. As in [2], we conclude that *supervision* is a key ingredient for learning structured relational models.

We demonstrate results on the difficult task of pose estimation. We use two standard benchmark datasets [23, 10]. We outperform all published past work on both datasets, reducing error by up to **50%**. We do so with a novel but simple representation that is orders of magnitude faster than previous approaches. Our model requires roughly 1 second to process a typical benchmark image, allowing for the possibility of real-time performance with further speedups (such as cascaded or parallelized implementations).

2. Related Work

Pose estimation has typically been addressed in the video domain, dating back to classic model-based approaches of O'Rourke and Badler [22], Hogg [13], Rohr [25]. Recent work has examined the problem for static images, assuming that such techniques will be needed to initialize video-based articulated trackers. Probabilistic formulations are common. One area of research is the encoding of spatial structure. Tree models are efficient and allow for efficient inference [7], but are plagued by the well-known phenomena of double-counting. Loopy models require approximate inference strategies such as importance sampling [7, 18], loopy belief propagation [28], or iterative approximations [33]. Recent work has suggested that branch and bound algorithms with tree-based lower bounds can globally solve such problems [31, 29]. Another approach to tackling the double-counting phenomena is the use of stronger pose priors, advocated by [17]. However, such approaches maybe more susceptible to overfitting to statistics of a particular dataset, as warned by [28, 32].

An alternate family of techniques has explored the trade-off between generative and discriminative models trained explicitly for pose estimation. Approaches include conditional random fields [24] and margin-based or boosted detectors [27, 16, 1, 29]. A final crucial issue is that of feature descriptors. Past work has explored the use of superpixels [21], contours [27, 26, 30], foreground/background color models [23, 10], and gradient descriptors [1, 15].

In terms of object detection, our work is most similar to pictorial structure models that reason about mixtures of parts [5, 7, 9]. We show that our model generalizes such representations in Sec.3.1. Our model, when instanced as a tree, can be written as a recursive grammar of parts [8].

3. Model

Let us write I for an image, $p_i = (x, y)$ for the pixel location of part i and t_i for the mixture component of part i. We write $i \in \{1, \dots K\}$, $p_i \in \{1, \dots L\}$ and $t_i \in \{1, \dots T\}$. We call t_i the "type" of part i. Our motivating example of types include orientations of a part (e.g., a vertical versus horizontally oriented hand), but types may span semantic classes (an open versus closed hand). For notational convenience, we define the lack of subscript to indicate a set spanned by that subscript (e.g., $t = \{t_1, \dots t_K\}$).

Co-occurrence model: To score of a configuration of parts, we first define a compatibility function for part types that factors into a sum of local and pairwise scores:

$$S(t) = \sum_{i \in V} b_i^{t_i} + \sum_{ij \in E} b_{ij}^{t_i, t_j} \tag{1}$$

The parameter $b_i^{t_i}$ favors particular type assignments for part i, while the pairwise parameter $b_{ij}^{t_i,t_j}$ favors particular co-occurrences of part types. For example, if part types correspond to orientations and part i and j are on the same rigid limb, then $b_{ij}^{t_i,t_j}$ would favor consistent orientation assignments. We write $G = (V, E)$ for a K-node relational graph whose edges specify which pairs of parts are constrained to have consistent relations.

We can now write the full score associated with a configuration of part types and positions:

$$S(I, p, t) = S(t) + \sum_{i \in V} w_i^{t_i} \cdot \phi(I, p_i) + \sum_{ij \in E} w_{ij}^{t_i, t_j} \cdot \psi(p_i - p_j) \tag{2}$$

where $\phi(I, p_i)$ is a feature vector (e.g., HOG descriptor [3]) extracted from pixel location p_i in image I. We write $\psi(p_i - p_j) = \begin{bmatrix} dx & dx^2 & dy & dy^2 \end{bmatrix}^T$, where $dx = x_i - x_j$ and $dy = y_i - y_j$, the relative location of part i with respect to j. Notably, this relative location is defined with respect to

the pixel grid and not the orientation of part i (as in classic articulated pictorial structures [7]).

Appearance model: The first sum in (2) is an appearance model that computes the local score of placing a template $w_i^{t_i}$ for part i, tuned for type t_i, at location p_i.

Deformation model: The second term can be interpreted as a "switching" spring model that controls the relative placement of part i and j by switching between a collection of springs. Each spring is tailored for a particular pair of types (t_i, t_j), and is parameterized by its rest location and rigidity, which are encoded by $w_{ij}^{t_i,t_j}$.

3.1. Special cases

We now describe various special cases of our model which have appeared in the literature. One obvious case is $T = 1$, in which case our model reduces to a standard pictorial structure. More interesting cases are below.

Semantic part models: [5] argue that part appearances should capture semantic classes and not visual classes; this can be done with a type model. Consider a face model with eye and mouth parts. One may want to model different types of eyes (open and closed) and mouths (smiling and frowning). The spatial relationship between the two does not likely depend on their type, but open eyes may tend to co-occur with smiling mouths. This can be obtained as a special case of our model by using a single spring for all types of a particular pair of parts:

$$w_{ij}^{t_i,t_j} = w_{ij} \tag{3}$$

Mixtures of deformable parts: [9] define a mixture of models, where each model is a star-based pictorial structure. This can achieved by restricting the co-occurrence model to allow for only globally-consistent types:

$$b_{ij}^{t_i,t_j} = \begin{cases} 0 & \text{if} \quad t_i = t_j \\ -\infty & \text{otherwise} \end{cases} \tag{4}$$

Articulation: In our experiments, we explore a simplified version of (2) with a reduced set of springs:

$$w_{ij}^{t_i,t_j} = w_{ij}^{t_i} \tag{5}$$

The above simplification states that the relative location of part with respect to its parent is dependant on part-type, but not parent-type. For example, let i be a hand part, j its parent elbow part, and assume part types capture orientation. The above relational model states that a sideways-oriented hand should tend to lie next to the elbow, while a downward-oriented hand should lie below the elbow, regardless of the orientation of the upper arm.

4. Inference

Inference corresponds to maximizing $S(x, p, t)$ from (2) over p and t. When the relational graph $G = (V, E)$ is a tree, this can be done efficiently with dynamic programming. Let $\text{kids}(i)$ be the set of children of part i in G. We compute the message part i passes to its parent j by the following:

$$\text{score}_i(t_i, p_i) = b_i^{t_i} + w_{t_i}^i \cdot \phi(I, p_i) + \sum_{k \in \text{kids}(i)} m_k(t_i, p_i) \tag{6}$$

$$m_i(t_j, p_j) = \max_{t_i} b_{ij}^{t_i,t_j} + \max_{p_i} \text{score}(t_i, p_i) + w_{ij}^{t_i,t_j} \cdot \psi(p_i - p_j) \tag{7}$$

(6) computes the local score of part i, at all pixel locations p_i and for all possible types t_i, by collecting messages from the children of i. (7) computes for every location and possible type of part j, the best scoring location and type of its child part i. Once messages are passed to the root part $(i = 1)$, $\text{score}_1(c_1, p_1)$ represents the best scoring configuration for each root position and type. One can use these root scores to generate multiple detections in image I by thresholding them and applying non-maximum suppression (NMS). By keeping track of the argmax indices, one can backtrack to find the location and type of each part in each maximal configuration.

Computation: The computationally taxing portion of dynamic programming is (7). One has to loop over $L \times T$ possible parent locations and types, and compute a max over $L \times T$ possible child locations and types, making the computation $O(L^2T^2)$ for each part. When $\psi(p_i - p_j)$ is a quadratic function (as is the case for us), the inner maximization in (7) can be efficiently computed for each combination of t_i and t_j in $O(L)$ with a max-convolution or distance transform [7]. Since one has to perform T^2 distance transforms, message passing reduces to $O(LT^2)$ per part.

Special cases: Model (3) maintains only a single spring per part, so message passing reduces to $O(L)$. Models (4) and (5) maintain only T springs per part, reducing message passing to $O(LT)$. It is worthwhile to note that our articulated model is no more computationally complex than the deformable mixtures of parts in [9], but is considerably more flexible (as we show in our experiments). In practice, T is small (≤ 6 in our experiments) and the distance transform is quite efficient, so the computation time is dominated by computing the local scores of each type-specific appearance models $w_i^{t_i} \cdot \phi(I, p_i)$. Since this score is linear, it can be efficiently computed for all positions p_i by optimized convolution routines.

5. Learning

We assume a supervised learning paradigm. Given labeled positive examples $\{I_n, p_n, t_n\}$ and negative examples

$\{I_n\}$, we will define a structured prediction objective function similar to those proposed in [9, 16]. To do so, let us write $z_n = (p_n, t_n)$ and note that the scoring function (2) is linear in model parameters $\beta = (w, b)$, and so can be written as $S(I, z) = \beta \cdot \Phi(I, z)$. We would learn a model of the form:

$$\begin{aligned} \arg\min_{w,\xi_i \geq 0} \quad & \frac{1}{2}\beta \cdot \beta + C \sum_n \xi_n \\ \text{s.t.} \quad \forall n \in \text{pos} \quad & \beta \cdot \Phi(I_n, z_n) \geq 1 - \xi_n \\ \forall n \in \text{neg}, \forall z \quad & \beta \cdot \Phi(I_n, z) \leq -1 + \xi_n \end{aligned} \tag{8}$$

The above constraint states that positive examples should score better than 1 (the margin), while negative examples, for all configurations of part positions and types, should score less than -1. The objective function penalizes violations of these constraints using slack variables ξ_n.

Detection vs pose estimation: Traditional structured prediction tasks do not require an explicit negative training set, and instead generate negative constraints from positive examples with mis-estimated labels z. This corresponds to training a model that tends to score a ground-truth pose highly and alternate poses poorly. While this translates directly to a pose estimation task, our above formulation also includes a "detection" component: it trains a model that scores highly on ground-truth poses, but generates low scores on images without people. We find the above to work well for *both* pose estimation and person detection.

Optimization: The above optimization is a quadratic program (QP) with an exponential number of constraints, since the space of z is $(LT)^K$. Fortunately, only a small minority of the constraints will be active on typical problems (e.g., the support vectors), making them solvable in practice. This form of learning problem is known as a structural SVM, and there exists many well-tuned solvers such as the cutting plane solver of SVMStruct [11] and the stochastic gradient descent solver in [9]. We found good results by implementing our own dual coordinate-descent solver, which we will describe in an upcoming tech report.

5.1. Learning in practice

Most human pose datasets include images with labeled joint positions [23, 10, 2]. We define parts to be located at joints, so these provide part position labels p, but not part type labels t. We now describe a procedure for generating type labels for our articulated model (5).

We first manually define the edge structure E by connecting joint positions based on average proximity. Because we wish to model articulation, we can assume that part types should correspond to different relative locations of a part with respect to its parent in E. For example, sideways-oriented hands occur next to elbows, while downward-facing hands occur below elbows. This means we can use relative location as a supervisory cue to help derive type labels that capture orientation.

Deriving part type from position: Assume that our n^{th} training image I_n has labeled joint positions p_n. Let p_i^n be the relative position of part i with respect to its parent in image I_n. For each part i, we cluster its relative position over the training set $\{p_i^n : \forall n\}$ to obtain T clusters. We use K-means with $K = T$. Each cluster corresponds to a collection of part instances with consistent relative locations, and hence, consistent orientations by our arguments above. We define the type labels for parts t_i^n based on cluster membership. We show example results in Fig.3.

Partial supervision: Because part type is derived heuristically above, one could treat t_i^n as a latent variable that is also optimized during learning. This latent SVM problem can be solved by coordinate descent [9] or the CCP algorithm [34]. We performed some initial experiments with latent updating of part types using the coordinate descent framework of [9], but we found that type labels tend not to change over iterations. We leave such partially-supervised learning as interesting future work.

Problem size: On our training datasets, the number of positive examples varies from 200-1000 and the number of negative images is roughly 1000. We treat each possible placement of the root on a negative image as a unique negative example x_n, meaning we have millions of negative constraints. Furthermore, we consider models with hundreds of thousands of parameters. We found that a careful optimized solver was necessary to manage learning at this scale.

6. Experimental Results

Datasets: We evaluate results using the Image Parse dataset [23] and the Buffy dataset [10]. The Parse set contains 305 pose-annotated images of highly-articulated full-body images of human poses. The Buffy dataset contains 748 pose-annotated video frames over 5 episodes of a TV show. Both datasets include a standard train/test split, and a standardized evaluation protocol based on the probability of a correct pose (PCP), which measures the percentage of correctly localized body parts. Notably, Buffy is also distributed with a set of validated detection windows returned by an upper-body person detector run on the testset. Most previous work report results on this set, as do we. Since our model also serves as a person detector, we can also present PCP results on the full Buffy testset. To train our models, we use the negative training images from the INRIAPerson database [3] as our negative training set. These images tend to be outdoor scenes that do not contain people.

Models: We define a full-body skeleton for the Parse set, and a upper-body skeleton for the Buffy set. To define a fully labeled dataset of part locations and types, we group parts into orientations based on their relative location with respect to their parents (as described in Sec 5.1). We show clustering results in Fig.3. We use the derived type labels to construct a fully supervised dataset, from which we learn

flexible mixtures of parts. We show the full-body model learned on the Parse dataset in Fig.5. We set all parts to be 5×5 HOG cells in size. To visualize the model, we show 4 trees generated by selecting one of the four types of each part, and placing it at its maximum-scoring position. Recall that each part type has its own appearance template and spring encoding its relative location with respect to its parent. This is because we expect part types to correspond to orientation because of the supervised labeling shown in Fig.3. Though we visualize 4 trees, we emphasize that there exists an *exponential* number of trees that our model can generate by composing different part types together.

Structure: We consider the effect of varying T (the number of mixtures or types) and K (number of parts) on the accuracy of pose estimation on the Parse dataset in Fig.4. We experiment with a 14 part model defined at 14 joint positions (shoulder, elbow, hand, etc.) and a 27 part model where midway points between limbs are added (mid-upper arm, mid-lower arm, etc.) to increase coverage. Performance increases with denser coverage and an increased number of part types, presumably because additional orientations are being captured. For reference, we also trained a star model, but saw inferior performance compared to the tree models shown in Fig.4. We saw a slight improvement using a variable number of mixtures (5 or 6) per part, tuned by cross validation. These are the results presented below.

Detection accuracy: We use our model as an upper body detector on the Buffy dataset in Table 1. We correctly detect **99.6%** of the people in the testset. The dataset include two alternate detectors based on a rigid HOG template and a mixtures-of-star models [9] which perform at 85% and 94%, respectively. The latter is widely regarded as a state-of-the-art system for object recognition. These results indicate the potential of our representation and supervised learning framework for general object detection.

Parse: We give quantitative results for PCP in Table 2, and show example images in Fig.6. We refer the reader to the captions for a detailed analysis, but our method outperforms all previously published results by a significant margin. Notably, all previous work uses articulated parts. We reduce error by **25%**. We believe our high performance is to due to the fact that our models leverage orientation-specific statistics (Fig.2), and because parts and relations are simultaneously learned in a discriminative framework. In contrast, articulated models are often learned in stages (using pre-trained, orientation-invariant part detectors) due to the computational burden of inference.

Buffy: We give quantitative results for PCP in Table 3, and show example images in Fig.7. We refer the reader to the captions for a detailed analysis, but we outperform all past approaches, when evaluated on a subset of standardized windows or the entire testset. Notably, all previous approaches use articulated parts. Our algorithm is several orders of magnitude faster than the next-best approaches of [26, 27]. When evaluated on the entire testset, our approach reduces error by **54%**.

Upper body detection on Buffy

Rigid HOG[10]	Mixtures of Def. Parts[9]	Us
85.1	93.8	99.6

Table 1: Our model clearly outperforms past approaches for upper body detection. Notably, [9] use a star-structured model of HOG templates trained with weakly-supervised data. Our results suggest more complex object structure, when learned with supervision, can yield improved results for detection.

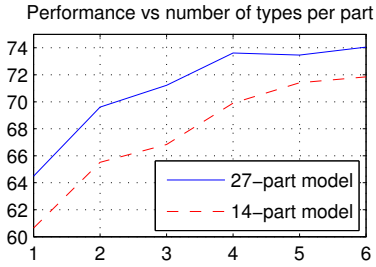

Figure 4: We show the effect of model structure on pose estimation by evaluating PCP performance on the Parse dataset. Overall, increasing the number of parts (by instancing parts at limb midpoints in addition to joints) improves performance. For both cases, increasing the number of mixture components improves performance, likely due to the fact that more orientations can be modeled.

Conclusion: We have described a simple, but flexible extension of tree-based models of part mixtures. When part mixture models correspond to part orientations, our representation can model articulation with greater speed and accuracy than classic approaches. Our representation provides a general framework for modeling co-occurrence relations between mixtures of parts as well as classic spatial relations between the location of parts. We show that such relations capture notions of local rigidity. We are applying this approach to the task of general object detection, but have already demonstrated impressive results for the challenging task of human pose estimation.

Acknowledgements: Funding for this research was provided by NSF Grant 0954083, ONR-MURI Grant N00014-10-1-0933, and support from Google and Intel.

References

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

[2] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *CVPR*, pages 1365–1372. IEEE, 2010.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages I: 886–893, 2005.

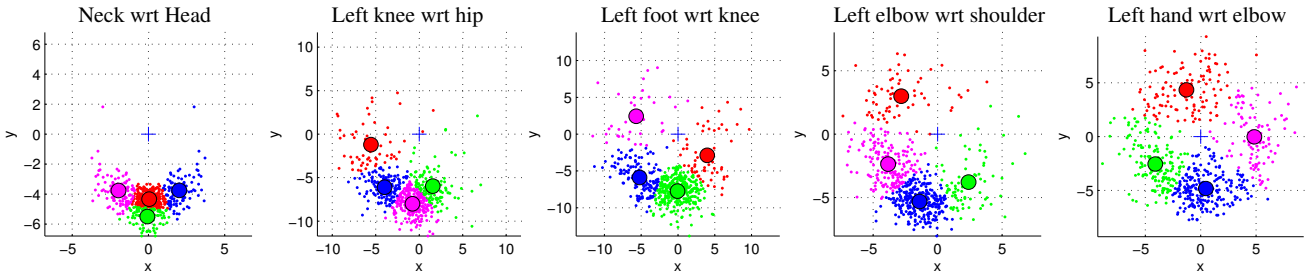

Figure 3: We take a "data-driven" approach to orientation-modeling by clustering the relative locations of parts with respect to their parents. These clusters are used to generate mixture labels for parts during training. For example, heads tend to be upright, and so the associated mixture models focus on upright orientations. Because hands articulate to a large degree, mixture models for the hand are spread apart to capture a larger variety of relative orientations.

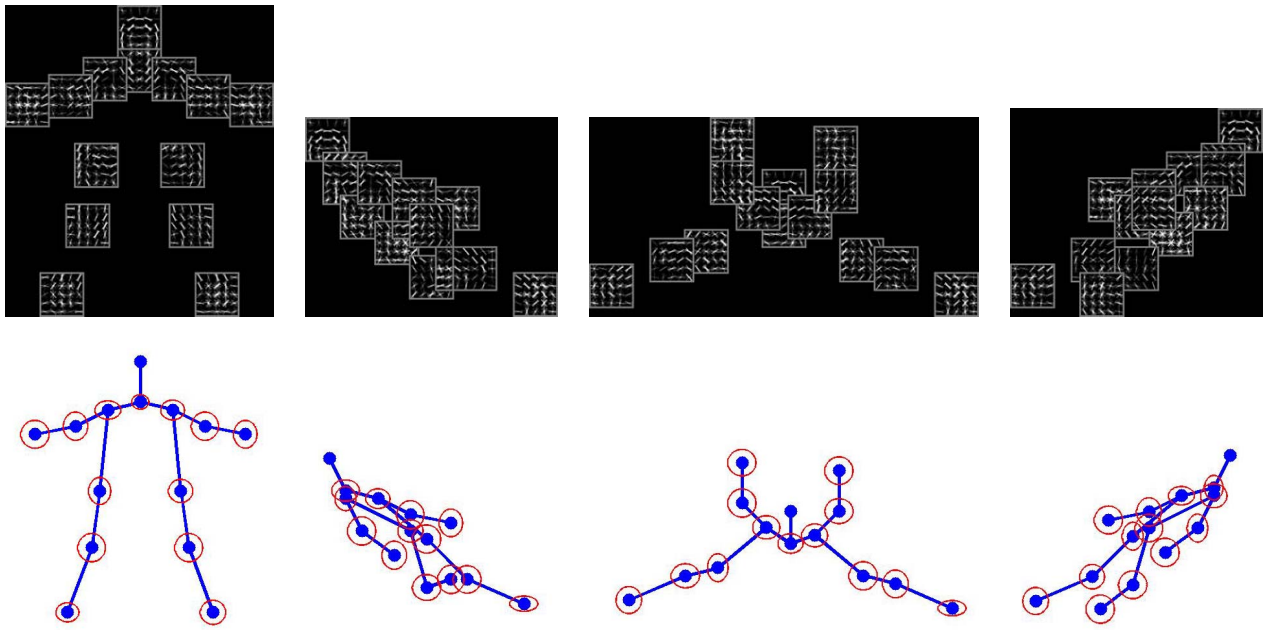

Figure 5: A visualization of our model for $T = 4$, trained on the Parse dataset. We show the local templates **above**, and the tree structure **below**, placing parts at their best-scoring location relative to their parent. Though we visualize 4 trees, there exists an *exponential* number of realizable combinations, obtained composing different part types together. The score associated with each combination decomposes into a tree, and so is efficient to search over (1).

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

[5] B. Epshtein and S. Ullman. Semantic hierarchies for recognizing objects and parts. In *CVPR*, pages 1–8, 2007.

[6] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010.

[7] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[8] P. Felzenszwalb and D. McAllester. Object detection grammars. Technical report, University of Chicago, 2010.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE PAMI*, 99(1), 5555.

[10] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, June 2008.

[11] T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *Proceedings of the 25th international conference on Machine learning*, pages 304–311. ACM New York, NY, USA, 2008.

[12] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *Computers, IEEE Transactions on*, 100(1):67–92, 1973.

[13] D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision computing*, 1(1):5–20, 1983.

[14] S. Johnson and M. Everingham. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *Proc. BMVC*, 2010.

[15] S. Johnson and M. Everingham. Combining discriminative appearance and segmentation cues for articulated human pose estimation. In *ICCV Workshops*, pages 405–412, 2010.

[16] M. Kumar, A. Zisserman, and P. Torr. Efficient discriminative learning of parts-based models. In *CVPR*, pages 552–559. IEEE, 2010.

Image Parse Testset

Method	Torso	Head	Upper legs	Lower legs	Upper arms	Lower arms	Total
R [23]	52.1	37.5	31.0	29.0	17.5	13.6	27.2
ARS [1]	81.4	75.6	63.2	55.1	47.6	31.7	55.2
JEa [15]	77.6	68.8	61.5	54.9	53.2	39.3	56.4
SNH [29]	91.2	76.6	71.5	64.9	50.0	34.2	60.9
JEb [14]	85.4	76.1	73.4	65.4	64.7	46.9	66.2
Our Model	**97.6**	**93.2**	**83.9**	**75.1**	**72.0**	**48.3**	**74.9**

Table 2: We compare our model to all previous published results on the Parse dataset, using the standard criteria of PCP. Our total performance of 74.9% compares favorably to the best previous result of 66.2%. We also outperform all previous results on a per-part basis.

Subset of Buffy Testset

Buffy	Torso	Head	U. arms	L. arms	Total
TF [32]					62.3
ARS [1]	90.7	95.5	79.3	41.2	73.5
EFZ [4]	98.7	97.9	82.8	59.8	80.1
SJT [26]	**100**	**100**	91.1	65.7	85.9
STT [27]	**100**	96.2	95.3	63.0	85.5
Our Model	**100**	99.6	**96.6**	**70.9**	**89.1**

Full Buffy Testset

Torso	Head	U. arms	L. arms	Total
				53.0
77.2	81.3	67.5	35.1	62.6
84.0	83.4	70.5	50.9	68.2
85.1	85.1	77.6	55.9	73.1
85.1	81.9	81.1	53.6	72.8
99.6	**98.9**	**95.1**	**68.5**	**87.6**

Table 3: The Buffy testset is distributed with a subset of windows detected by a rigid HOG upper-body detector. We compare our results to all published work on this set on the **left**. We obtain the best overall PCP while being orders of magnitude faster than the next-best approaches. Our total pipeline requires 1 second to process an image, while [26, 27] take 5 minutes. We outperform or (nearly) tie all previous results on a per-part basis. As pointed out by [32], this subset contains little pose variation because it is biased to be responses of a rigid template. The distributed evaluation protocol also allows one to compute performance on the full test videos by multiplying PCP values with the overall detection rate. We do this for published results on the **right** table. Because our model also serves as a very accurate detector (Table 1), we obtain significantly better results than past work when evaluated on the full testset.

[17] X. Lan and D. Huttenlocher. Beyond trees: Common-factor models for 2d human pose recovery. In *CVPR*, volume 1, pages 470–477. IEEE, 2005.

[18] M. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *CVPR*, volume 2. IEEE, 2004.

[19] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV04 workshop on statistical learning in computer vision*, pages 17–32. Citeseer, 2004.

[20] D. Marr and H. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 200(1140):269–294, 1978.

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

[22] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *PAMI*, 2(6):522–536, 1980.

[23] D. Ramanan. Learning to parse images of articulated bodies. *NIPS*, 19:1129, 2007.

[24] D. Ramanan and C. Sminchisescu. Training deformable models for localization. In *CVPR*, volume 1, pages 206–213. IEEE, 2006.

[25] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP-Image Understanding*, 59(1):94–115, 1994.

[26] B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *CVPR*, pages 422–429. IEEE, 2010.

[27] B. Sapp, A. Toshev, and B. Taskar. Cascaded Models for Articulated Pose Estimation. *ECCV 2010*, pages 406–420, 2010.

[28] L. Sigal and M. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *CVPR*, volume 2, pages 2041–2048. IEEE, 2006.

[29] V. Singh, R. Nevatia, and C. Huang. Efficient inference with multiple heterogenous part detectors for human pose estimation. In *ECCV*, 2010.

[30] P. Srinivasan and J. Shi. Bottom-up recognition and parsing of the human body. In *CVPR*, 2007.

[31] T. Tian and S. Sclaroff. Fast globally optimal 2D human detection with loopy graph models. In *CVPR*, pages 81–88, 2010.

[32] D. Tran and D. Forsyth. Improved Human Parsing with a Full Relational Model. *ECCV*, pages 227–240, 2010.

[33] Y. Wang and G. Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. *ECCV*, pages 710–724, 2008.

[34] A. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

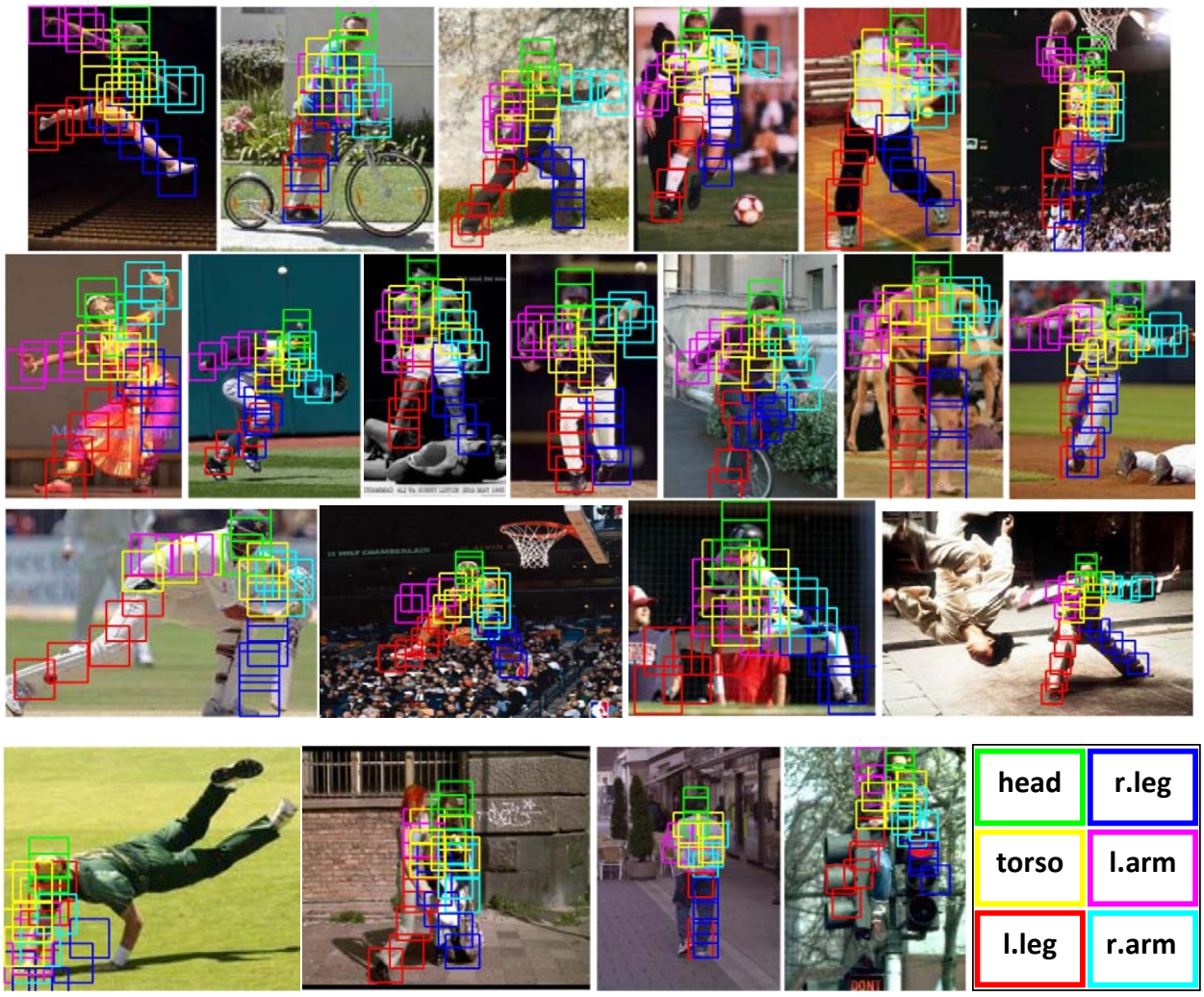

Figure 6: Results on the Parse dataset. We show 27 part bounding boxes reported by our algorithm for each image. The **top 3** rows show successful examples, while the **bottom** row shows failure cases. Examining failure cases from left to right, we find our model is not flexible enough to model horizontal people, is confused by overlapping people, suffers from double-counting phenomena common to tree models (both the left and right legs fire on the same image region), and is confused when objects partially occlude people.

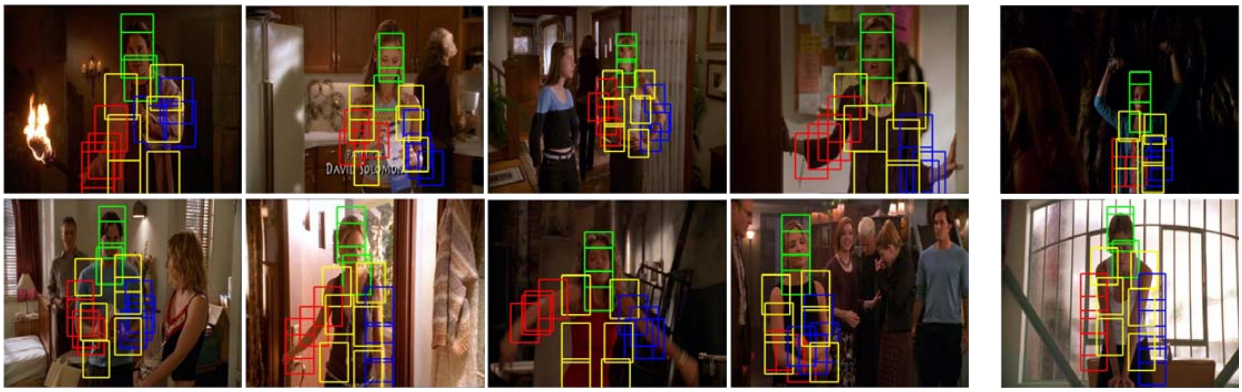

Figure 7: Results on the Buffy dataset. We show 17 part bounding boxes, corresponding to upper body parts, reported by our algorithm. The **left 4** columns show successful examples, while the **right** column show failure cases. From top to bottom, we see that our model still has difficultly with raised arms and is confused by vertical limb-like clutter in the background.