HUMAN PARSING WITH A CASCADE OF HIERARCHICAL POSELET BASED PRUNERS

Duan Tran[†] *Yang Wang*[‡] *David Forsyth*[†]

University of Illinois at Urbana Champaign[†] University of Manitoba[‡]

ABSTRACT

We address the problem of human parsing using part-based models. In particular, we consider part-based models that exploit rich pairwise relationship between parts, e.g. the color symmetry between left/right limbs. This poses a computational challenge since the state space of each part is very large, and algorithmic tricks (e.g. the distance transform) cannot be applied to handle these types of pairwise relationships. We propose to prune the state space of each part using a cascade of pruners. These pruners can filter out 99.6% of the states per part to about 500 states per part, while keeping the ground-truth states in the pruned state most of the time. In the pruned space, we can afford to apply human parsing models with more complex pairwise relationships between parts, such as the color symmetry. We demonstrate our method on a challenging human parsing dataset.

Index Terms— human pose estimation, gesture analysis

1. INTRODUCTION

Part-based models (e.g. pictorial structures [1]) are a class of popular approaches in human parsing. These models represent the human pose as a collection of parts. Each part is represented by one or more appearance templates. An undirected graphical model is used to capture the pairwise relationships between pairs of parts that are connected.

Many existing part-based approaches can be broadly categorized into three (possibly overlapping) classes: (1) Tractable models (e.g. kinematic tree) with exact inference (e.g. [1, 2]). These approaches typically use tree-structured models and simple pairwise relationships between parts. Inference can often be done exactly. Tree-structured models have been shown to be very successful, in particular [2] has achieved the state-of-the-art results on several benchmark datasets. But they are severely restrictive in what can be modeled. For example, the color symmetry of left/right limbs are usually not captured in those models. (2) Intractable models with approximate inference (e.g. [3]). These approaches use non-tree loopy graph models and offer general insights on how to move beyond simple tree models in order to exploit richer relationships of parts. However, due to the computational difficulty, the pairwise relationship is still limited to only simple spatial relationships in most of these models. (3) Pruning-based approaches (e.g. [4, 5]). These approaches aim to reduce the search space using various strategies. We believe pruning-based approaches are important since they provide generic ideas that can be used with both tree-structured and non-tree models. In addition, both learning and inference can benefit from efficient pruning strategies.

To motivate the need for pruning, let us take a closer look at the computational complexity of the inference algorithms in part-based models. For tree-structured models, the search over the joint pose space requires $O(M \cdot K^2)$ computation, where M is the number of parts and K is the number of states for a part. In human parsing, M is usually small. However, the state space K for a part is often quite large. For example, in [1], it is roughly the number of pixels in an image. A brute-force search algorithm is often infeasible. In the literature, people have taken two approaches to address this computational issue. First, one can impose some restriction on the form of pairwise relationship in the model, so that fast algorithmic tricks can be applied. For example, the pictorial structure in [1] assumes the pairwise relationship is a Mahalanobis distance between the locations of a pair of parts. In that case, distance transform can be applied to reduce the complexity to $O(M \cdot K)$. Unfortunately this approach can only handle some special forms of pairwise relationships. More complex pairwise relationships (e.g. color symmetry between left/right limbs) cannot be directly used due to the computational issue, even though they are very helpful. The second approach is to develop a strategy to prune the state space of each part, so we can apply brute-force search. Examples of such approaches include [5, 6]. Our work falls under the second approach. We learn a cascade of pruners to progressively filter out the state space for each part. Some pruners in the cascade are easy to compute and can be applied to quickly remove a large portion of the state space. Then more complex pruners are applied to further prune the space. In the end, we are left with a small state space for each part. In this pruned space, we can afford to apply rich models that involve complex pairwise relationships between parts. We demonstrate our approach on a challenging dataset involving very aggressive pose variations. We show that even though our final human parsing model is applied on a pruned state space, we can still outperforms other competing methods, since we can leverage the power of more complex pairwise relationships (e.g. color symmetry).

Related work: Part-based representation is a popular approach in human parsing. Tree-structured models [7, 1, 8, 2] are commonly used due to their efficiency. Loopy models [9, 10, 11, 3] have also been developed, but they usually require approximate inference.

Many part-based models use discriminative learning to train the model parameters. Examples include the conditional random fields [12, 8], max-margin learning [13, 3, 2] and boosting [7, 5, 14]. Previous approaches have also explored various features, including image segments (superpixels) [15, 16, 17, 18, 5, 19], color features [8, 4], gradient features [7, 20, 3, 2].

Our cascaded pruning strategy is mostly related to Sapp et al. [5]. In their work, they use a coarse-to-fine cascade of pictorial structures for human parsing. Our work is different from [5] in that we consider larger parts in addition to rigid parts.

The part-based model used in our work is mostly related to the hierarchical poselet human parsing in [3]. Most previous work of part-based models only considers rigid parts (e.g. torso, head, half limbs). The method in [3] extends traditional part-based models by introducing larger parts that are compositions of multiple rigid parts. Their argument is that those larger parts usually have distinctive appearance patterns that are easier to detect. Similar ideas have also been used in [21]. In our work, we apply hierarchical poselets in the setting of state space pruning.

Although we build upon the part-based models in [3], it is important to keep in mind that we are proposing a general pruning framework that can be used with any part-based models.

2. PART-BASED MODELS FOR HUMAN PARSING

We briefly review part-based models for human parsing in this section. In particular, we focus on the hierarchical poselet models introduced in [3] which our method builds upon.

The human body is represented as a collection of M parts. A standard pictorial structure [1, 8, 2] uses 10 parts including head, torso, half limbs. We can use an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent the pose configuration. A vertex $i \in \mathcal{V}$ in the graph corresponds to the i-th part. Its label l_i represents the configuration of this part. In standard pictorial structures, l_i encodes the (x, y) location and the orientation of the part. An edge $(i, j) \in \mathcal{E}$ in the graph captures certain constraints (e.g. spatial and/or appearance) between a pair of parts.

Recently, Wang et al. [3] have extended standard part-based models by introducing larger parts that are compositions of several rigid parts into their model. In particular, they define 20 parts represented by a loopy graph shown in Fig. 1 (left). Each part is represented by several *poselets* – a concept first introduced by Bourdev et al. [22, 23]. In a nutshell, poselets refer to body parts that are tightly clustered. In [3], the configuration l_i of the i-th part encodes the (x, y) location and the poselet index of this part.

Part-based models use a scoring function in the following form to measure the compatibility of an image $\mathcal{I}$ and a pose configuration L:

$$C(L; \mathcal{I}, \mathcal{W}) = \sum_{i \in \mathcal{V}} w_i^\top \Phi_i(l_i; \mathcal{I}) + \sum_{i,j \in \mathcal{E}} w_{ij}^\top \Phi_{ij}(l_i, l_j; \mathcal{I}) \quad (1)$$

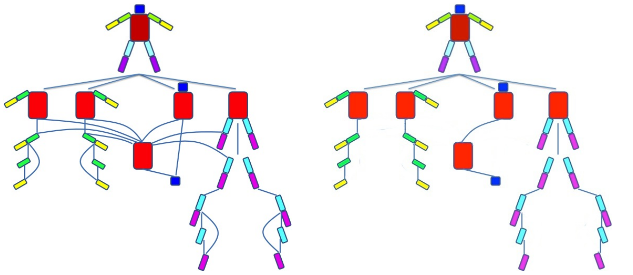


Fig. 1*:* (Left) The relational graph of the hierarchical poselet model in [3]. Each vertex represents a part. Each edge represents the pairwise relationship between two parts. (Right) A tree structured model obtained by removing some edges in the model on the left.

where $\mathcal{W} = \{w_i, w_{ij} : i \in \mathcal{V}, (i,j) \in \mathcal{E}\}$ are model parameters. $\Phi_i(l_i; \mathcal{I})$ is a unary potential of part i. $\Phi_{ij}(l_i, l_j; \mathcal{I})$ is the pairwise relation between part i and part j. In standard pictorial structures, the $\mathcal{E}$ forms a tree. While in [3], $\mathcal{E}$ forms a loopy graph. Finding the best pose configuration L^* involves solving the following inference problem (also known as MAP assignment): $L^* = \arg\max_{L \in \mathcal{L}} C(L; \mathcal{I}, \mathcal{W})$. This inference problem can be solved by belief propagation (BP) when $\mathcal{E}$ is a tree (as in [1, 8, 2]) or by loopy belief propagation (LBP) when $\mathcal{E}$ has cycles (as in [3]).

As mentioned earlier, the computational bottleneck of applying BP/LBP is that it involves a computation of $O(K^2)$ where K is the number of possible labels of a part. Typically, K can be about 10,000 for each part. Brute-force BP/LBP is obviously infeasible. When the pairwise potential has some special form, one can use some algorithmic tricks (e.g. distance transform [1]) to reduce the computation to $O(K)$. Unfortunately, the distance transform cannot handle all forms of pairwise potentials. For example, the color symmetry between left/right limbs is a very useful cue for human parsing. But it is rarely used in previous work, mainly because it requires a pairwise potential that cannot be handled by the distance transform or other computation tricks.

3. HUMAN PARSING WITH CASCADED PRUNERS

The main idea of our work is to develop a strategy to efficiently prune the search space of each part. In the pruned space, we will be able to use more complex models that involve richer pairwise relationships between parts (e.g. color symmetry between left/right limbs) by applying brute-force BP/LBP. In this section, we introduce a sequence of increasingly complex pruners that progressively reduce the search space.

Our pruning strategy is inspired by Sapp et al. [5] that learn a coarse-to-fine cascade of pictorial structure models for human parsing. In particular, they prune the search space of l_i using the *max-marginal* defined as follows: $C(l_i; \mathcal{I}, \mathcal{W}) = \max_{l' \in \mathcal{L}} \{C(l'; \mathcal{I}, \mathcal{W}) : l'_i = l_i\}$. Let us define the best MAP assignment as $C^*(\mathcal{I}; \mathcal{W}) = \max_{l \in \mathcal{L}} \{C(l; \mathcal{I}; \mathcal{W})\}$. The method in [5] learns to prune the state space by considering the two competing objectives that trade off accuracy and efficiency:

$$t(\mathcal{I},\mathcal{W},\alpha)=\alpha C^*(\mathcal{I};\mathcal{W})+\frac{1-\alpha}{M}\sum_{i=1}^{M}\frac{1}{|\mathcal{L}_i|}\sum_{l\in\mathcal{L}_i}C(l_i;\mathcal{I}) \qquad (2)$$

where α is a parameter to control how aggressively to prune. When $\alpha = 1$, only the MAP assignment is kept. When $\alpha = 0$, approximately half of the space is pruned.

Sapp et al. [5] learn the parameter $\mathcal{W}$ for pruning using a max-margin approach. To do so, they minimize the hinge loss of the incorrectly pruned part states over the example set $X = \{(\mathcal{I}^n, L^n)\}_{n=1}^{N}$ as follows:

$$\min \lambda\frac{1}{2}\parallel\mathcal{W}\parallel^2+\frac{1}{N}\sum_{n\in examples}\xi_n(L^n,\mathcal{I}^n;\mathcal{W}) \qquad (3)$$

where $\xi_n(L^n,\mathcal{I}^n;\mathcal{W}) = \max\{0, 1 + t^n(\mathcal{I}^n,\mathcal{W},\alpha) - C(L^n;\mathcal{I}^n;\mathcal{W})\}$ is the hinge loss to measure the margin between the scores of the MAP assignment and the ground truth. This essentially learns $\mathcal{W}$ to ensure the score of ground-truth label is larger than the score of any other competing labels by a margin of 1. The update rule of $\mathcal{W}$ at iteration $t+1$ is:

$$\mathcal{W}^{t+1} \leftarrow \mathcal{W}^t + \eta(-\lambda\mathcal{W}^t + \nabla_{\mathcal{W}^t}), \quad \text{where } \nabla_{\mathcal{W}^t} = \Phi(L;\mathcal{I}) - \alpha\Phi(L^*;\mathcal{I}) - (1-\alpha)\frac{1}{M}\sum_i\frac{1}{|\mathcal{L}_i|}\sum_{l_i\in\mathcal{L}_i}\Phi(l_i;\mathcal{I})$$

where η is the learning rate, $\Phi(L;\mathcal{I})$ is the feature vector at the ground truth configuration L of the example $\mathcal{I}$, $\Phi(L^*;\mathcal{I})$ is the feature vector of the MAP assignment found with respect to $\mathcal{W}^t$.

Our pruners build upon Sapp et al. [5]. But there are several important distinctions. First, the cascade method in [5] uses the *same tree-structured model* at each stage at *different spatial resolutions* in a coarse-to-fine manner. The coarse-level model uses simple features and is only applied at several coarse spatial positions in the image. The fine-level model uses more complex features and is applied at finer resolution in the space pruned by coarser models. In contrast, we use *different models* at different levels of the cascade.

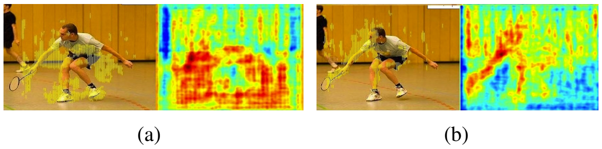
Second, the pruning framework in [5] is based on appearances of small parts (i.e. torso, head, half limbs) connected in the pictorial structure model. However, we would like to argue that at coarse levels, one should not even consider small parts since they are hard to distinguish. Instead, we should try to use appearance models from larger body parts that are easy to identify at the coarse level. In this paper, we use the hierarchical poselets introduced in [3] to learn appearance models for both small rigid parts, as well as large parts that are compositions (e.g. torso+legs) of several rigid parts. The detector responses of large parts will provide a contextual information to allow us to prune the search space of small parts.

It is also interesting to compare our work with the pruning method by Ferrari et al. [4]. In that work, the human parsing algorithm is run only within the spatial region localized by a pre-trained upper-body detector. In that case, the upper-body detector plays a similar role as the pruner based on appearance models of large parts (i.e. upper-body).

In the following, we introduce three types of increasingly complex pruners used in our cascade, namely *part pruners* (Sec 3.1), *tree-based pruners* (Sec 3.2), and *enhanced tree-based pruners* (Sec 3.3). These pruners are essentially models in forms similar to Eq. 1, but they differ in their definitions of *state space* of a part and the structure of their relational models. For part pruners and tree-based pruners, the state space l_i of a part consists of the 2D locations in the image, i.e. $l_i = (x_i, y_i)$. For enhanced tree-based pruners, the state space l_i also includes the poselet index, i.e. $l_i = (x_i, y_i, z_i)$. For notational convenience, we will use l_i in the following, but it is important to keep in mind that l_i has slightly different meanings in different pruners.

3.1. Part pruners

We call the first type of pruners the *part pruners*. Here the state space for a part is the 2D locations in the image. We consider a scoring function without pairwise potentials in the following form: $C_1(L;\mathcal{I},\mathcal{W}) = \sum_{i\in\mathcal{V}} w_i^\top\Phi_i(l_i;\mathcal{I})$, where $\Phi_i(l_i;\mathcal{I})$ is a vector of poselet responses at location l_i. Instead of only considering the poselets corresponding to the i-part, we also consider the poselet responses from other larger supporting parts. For example, if the i-th part corresponding to the *left upper arm*, one of the supporting part is the *left arm*. The feature vector $\Phi_i(l_i;\mathcal{I})$ is a vector of poselet responses from all these parts (both *left upper arm* and *upper arm* in this case). The parameters w_i re-weight the poselet responses from both part i and its supporting parts. Our intuition is that the appearance of larger parts (left arm) will provide some contextual information that help distinguishing small parts (left upper arm). This can be demonstrated in Fig. 2, where we show examples of space pruning with and without supporting larger parts. We can see that with contextual information provided from larger parts, the pruned states tend to peak around the ground-truth locations.



(a) (b)

Fig. 2*:* An example of state pruning (right lower arm) by part pruners with (a) and without (b) support from large parts. Each figure shows the remaining states and the heat map of part state confidence (warm color indicates high confidence of remaining un-pruned). Part pruners are able to prune away at least 75% unlikely states. The remaining states in (a) concentrate around the correct location while those in (b) scatter all over the image. This indicates that large parts are helpful in resolving ambiguity of small parts (best viewed in color).

Since there are no pairwise relations in part pruners, finding the MAP assignment can be done very efficiently by searching the best state for each part independently of other parts. We learn the parameters using the update rules in Eq. 4.

In our work, we build three level of part pruners instead of just one level, which allow pruners at different levels to focus on different types of unlikely part states. This is in analog to the different weak learners used at various stages in the Viola-Jones detector [24].

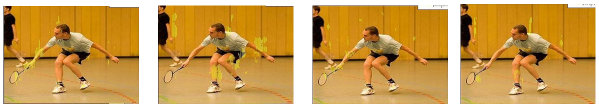
We choose the trade-off parameter α such that after these three levels of part pruners, we filter more than 75% states while safely keeping the correct locations in the remaining state space most of the time.

3.2. Tree-based pruners

The part pruners can only effectively prune a portion of the search space. If we set α to prune more aggressively, we will start filtering out the correct locations as well. In order to effectively prune more space, we need a more complex model. In this section, we introduce pruners based on hierarchical tree structured models shown in Fig. 1(right). This is a simplified structure of the loopy relational model used in [3] (Fig. 1(left)) by dropping edges in the loops. The model is more powerful than the part pruners at the expense of more expensive computation. But since the structure is a tree and the fact that we only need to consider states that pass the part pruners, the computation is still manageable.

Given an example $\mathcal{I}$ and a configuration $l = \{l_i\}_{i=1}^{M}$ of parts in the current state space, we use a compatibility function of the form Eq. 1. Similar to part pruners, the state space of a part is the 2D locations in the image. Bu the difference is that now the compatibility function has pairwise potentials that capture the spatial relationships between certain pairs of parts. Similarly, we learn the parameters of tree-based pruners using the update rules in 4. But now we need to consider pairwise potentials when computing the MAP assignment and max-marginals.

We build four levels of tree-based pruners. As shown in the experiments (Sec. 5), there are about 300-500 remaining states (2D locations) for each part after these levels.



(a) (b) (c) (d)

Fig. 3*:* Examples of remaining states of left lower arm and left lower leg after the tree-based pruners (3(a) and 3(b)) and after the enhanced tree-based pruners (3(c) and 3(d)). Once part pruners have been applied, tree-based pruners are applied to further prune away unlikely part states (more than 95%) and leave a small number of 2D part states (around 300-500 2D locations per part). However, when adding one more dimension of the part poselet index to the search spaces (8 to 15 poselets per part), the number of states per part is still relatively large (2400-4000 actual states per part). We perform enhanced tree-based pruners to work on the full state representation to prune to about 500 states per part. This set of states is small enough that exhaustive search is affordable.

3.3. Enhanced tree-based pruners

After the tree-based pruners, each part is left with about 300-500 2D locations to search over. We could apply a human parsing method (e.g. [3]) at this stage. However, the part configuration in [3] also involves a poselet index in addition to the 2D location. If we search over all part poselets (ranging from 8 poselets for the torso to 15 poselets for the half limbs), the search space is still too large (approximately 2500-4000 states per part). Therefore, we develop an enhanced tree-based pruners to prune the state space of the full representation (2D locations and poselet indices) used in [3].

We use the same tree structure in tree-based pruners (Fig. 1 (right)). But the difference is that the tree-based pruners in Sec. 3.2 only consider the (x, y) location of each part. In contrast, we now take the poselet index into the part state to make a full representation. Each state l_i for a part i is a triple of $l_i = (x_i, y_i, z_i)$ corresponding its 2D location and the poselet index.

In enhanced tree-based pruners, we also add part poselet priors and poselet co-occurrence priors to the compatibility function of Eq. 1. Yang et al. [2] have demonstrated the benefit of these priors for human parsing problems, although their notions of parts are small parts (patches around body joints).

Let $b_i(z_i)$ be the poselet prior of poselet z_i of part i, and $b_{ij}(z_i, z_j)$ be the co-occurrence prior of poselet z_i, z_j of part i and part j. Now, the compatibility function becomes:

$$
\begin{aligned}
C_b(l;\mathcal{I};\mathcal{W}_T) &= B(l) + \sum_{i\in\mathcal{V}} w_i^\top \Phi_i(l_i;\mathcal{I}) + \sum_{i,j\in\mathcal{E}} w_{ij}^\top \Phi_{ij}(l_i, l_j;\mathcal{I}) \\
B(l) &= \sum_{i\in\mathcal{V}} b_i(z_i) + \sum_{i,j\in\mathcal{E}} b_{ij}(z_i, z_j) \qquad (4)
\end{aligned}
$$

where $l = \{l_i = (x_i, y_i, z_i)\}_{i=1}^{M}$ are the 2D locations and poselet indices of parts.

Similarly, we use the update rules in Eq. 4 to learn the parameters. But the difference is that we use the compatibility function defined in Eq. 4. We build two levels of *enhanced tree-based pruners*. After these levels, we have only a small set of states (about 500 states per part). This set is small enough to build a parser using appearance models. Figure 3 shows an example of pruning after the tree-based pruners and the enhanced tree-based pruners. We will quantify the reduction rates after each level of the cascade in Sec. 5.1.

3.4. Final Human Parser

After the three pruning steps in Sec 3, we are left with a small number (∼500) of remaining states for each part. At this stage, we can afford to use complex models and pairwise features. Our parsing model is based on the hierarchical poselet model [3] which represents the configuration of the human pose using a 20-part model shown in Fig. 1 with a few more additional edges in the graph (details below). The configuration l_i of the i-th part is parameterized by the (x, y) location and the poselet index of the i-th part.

Table 1*:* Reduction rates for rigid part after each type of the cascade of pruners. Pruners are sequentially applied to prune on the current set of states. Each row shows the percentage of part states are filtered out after that pruner level. For advanced tree pruners we tune the thresholds such that about 99.6% of part states that are removed after all these levels. This ensures the number of part states are small enough for the final parser.

Part **Pruner**	torso	head	u-arm	l-arm	u-leg	l-leg
Parts	88.8	79.0	75.6	75.0	76.5	77.9
Trees	97.5	95.8	95.1	95.0	95.3	95.4
Enhanced trees	99.6	99.6	99.6	99.6	99.6	99.6

Table 2*:* $PCP_{0.2}$ rates for oracle parser after each type of the cascade of pruners. Oracle states are the best states (assuming access to the ground-truth) chosen from the pool of the current remaining part states (see details in section 5). After applying pruners, the oracle torso gets 76.5% $PCP_{0.2}$ and lower arms get 54.9% $PCP_{0.2}$ (Note that Sapp's pruner [5] gets 54% $PCP_{0.2}$ for lower arm on the Buffy dataset). Note that $PCP_{0.2}$ is a more restrictive criterion than $PCP_{0.5}$. The 76.5% $PCP_{0.2}$ rate for torso means that for 76.5% of the examples, the states after pruners contain at least one state whose distance to the ground-truth is within 20% of the length of the torso.

Part **Pruner**	torso	head	u-arm	l-arm	u-leg	l-leg
Parts	88.3	81.8	80.1	75.9	86.3	83.5
Trees	79.8	69.3	65.5	61.9	72.4	70.9
Enhanced trees	76.5	61.2	57.2	54.9	67.7	64.1

In [3], the pairwise potentials are limited to simple spatial constraints since the inference algorithm (i.e. message passing) can be done efficiently for this type of pairwise potentials. Their method cannot exploit other richer pairwise potentials (e.g. color symmetry between left/right limbs) because the message passing algorithm involves $O(K^2)$ computation. But in our case, we have pruned K to a very small number (∼500). We can now afford this $O(K^2)$ computation using exhaustive search and exploit richer forms of pairwise potentials that depend on the image. We augment the graph in Fig. 1 by adding edges (not shown in Fig. 1 to keep it less cluttered) between left/right half-limbs and use the color symmetry as the pairwise appearance relations. Precomputing those pairwise potentials allows us to solve the MAP assignment efficiently.

4. IMPLEMENTATION DETAILS

We describe some implementation details in this section.

State space resolution: The 2D-location state space is on image grid of 100 × 100 (about 4 pixels for each grid cell size). This 2D resolution is the same in all levels of pruners. For enhanced poselet pruners, we expand to the full state representation by adding the part poselet index dimension. Poselet dimension sizes vary between 8 to 15 depending on parts. At the beginning there are 10,000 2D-states per part (w.r.t the grid size 100 × 100). Part pruners and tree-based pruners can filter out more than 95% states leaving about 300-500 2D-states. We then expand these 2D states to full representation by enumerating all poselet indexes. Then the enhanced tree-based pruners are applied to filter to about at most 500 states per part. The final human parsing algorithm is applied to search for the best pose configuration in the remaining set of states.

Unary features: For all models (pruners and parsers), we use part poselet responses as unary features. Poselet responses are precomputed for all examples from pre-trained poselet templates (using SVM on HOG features). For part pruners, the unary features at each part are augmented with the poselet responses from supporting parts at the same part location.

Pairwise features: For tree-based pruners and enhanced tree-based pruners, there are no pairwise potentials for the color symmetry. We only use pairwise potentials to capture the spatial constraints between pairs of parts. We use the binning scheme in [8] to represent the spatial constraint. For enhanced tree-based pruners, we represent features for pairwise relations of part poselets by a vector of size *(#poselet_part_i)* × *(#poselet_part_j)*. The corresponding index in the vector of a pair of poselets z_i, z_j will be 1, others are zeros. For the parser, we use color appearance models for related primitive parts to capture the similarity of symmetric parts (e.g. limbs on the left are similar to limbs on the right).

5. EXPERIMENTS

We evaluate our method on the challenging UIUC Sports dataset introduced by [3]. This dataset is an extension of the people dataset in [6] by adding more sport images downloaded from the Internet. There are totally 1299 examples of more than 20 sport categories including: badminton, acrobatics, cycling, American football, croquet, hockey, figure skating, soccer, golf, horseback riding, rugby, etc. Following [3], we use 650 images for training and 649 for testing.

5.1. Evaluation on the pruners

A good pruner should remove most of the states without filtering out the ground-truth states. We use the PCP measurement (percentage of correctly localized parts) defined in [4] to evaluate the pruners. $PCP_{k\%}$ means a predicted part is considered correct if its two end points lie within $k\%$ of the ground-truth length of the part. We evaluate our pruners by checking whether the pruned state space still contains a part within $PCP_{0.2}$ of the right answer (given by an oracle parser). The oracle parser will choose the best state for each part in the current state space. This essentially gives an upper bound performance of the parsing results on the pruned space.

We show the reduction rates for rigid parts up to each pruner level in Table 1 and the corresponding $PCP_{0.2}$ rates by an oracle parser in Table 2.. We can see that after applying the three types of pruners, we can filter out 99.6% of part states (keeping about 300-500 states per part) while still achieving high $PCP_{0.2}$ rates.

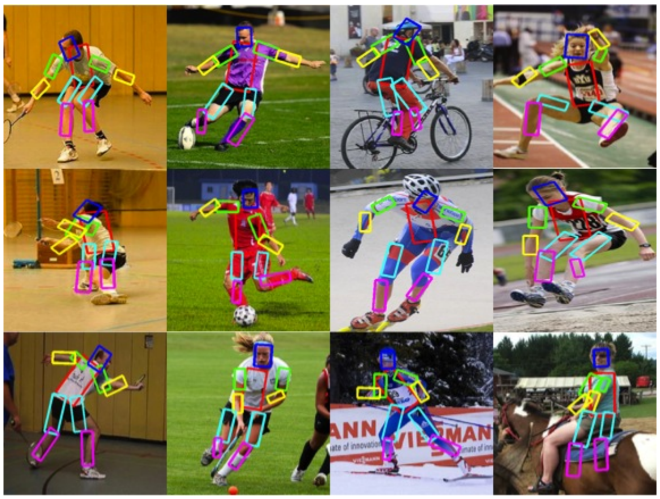


Fig. 4*:* Sample parsing results of our approach on the UIUC Sports dataset.

Table 3*:* Comparisons of parsing results with other methods on the UIUC Sports dataset. The percentage of correctly estimated parts ($PCP_{0.5}$) rigid body parts. If two numbers are shown in one cell, they indicate the left/right body parts. Note that the improvement of our method over [3] shows the benefit of modeling color symmetries between left/right limbs as pairwise potentials in the model.

Method	torso	upper leg		lower leg		upper arm		lower arm		head
[8]	28.7	7.4	7.2	17.6	20.8	8.3	6.6	20.2	21	12.9
[7]	71.5	44.2	43.1	30.7	31	28	29.6	17.3	15.3	63.3
[2]	81.1	57.2	55.4	53.9	52.0	42.9	44.9	27.7	29.9	67.1
[3]	75.3	50.1	48.2	42.5	36.5	23.3	27.1	12.2	10.2	47.5
Ours	82.0	53.2	52.0	46.7	44.3	29.7	31.5	12.8	12.2	49.0

5.2. Evaluation on human parsing

Figure 4 shows some typical parsing examples by our method. To evaluate the final human parsing results, we use $PCP_{0.5}$, which is a measurement commonly used by previous approaches. We compare our parsing results with other state-of-the-art methods in Table 3. The comparison with [3] is probably the most informative one, since [3] uses similar image features and model formulation, but without complex pairwise relationship of color symmetry. We can see that our method outperforms [3] for all the parts. This is a strong evidence of the benefit of using complex pairwise relationships in the pruned space. The only method that outperforms ours is the recent work by Yang and Ramanan [2]. However, we would like to point out that our work provides a general framework that can be adopted in any human parsing method. As future work, we plan to explore how to combine this framework with [2] to further improve the results.

6. CONCLUSIONS

We have proposed a cascade of pruners using hierarchical poselets. These pruners can reduce the state space to allow the use of complex models. What distinguishes our work from [5] is that our pruners are learned using both large parts and small rigid parts. Large parts provide useful contextual information for small parts. The pruner cascade effectively filter out more than 99.6% part states to about 500 states per part. We have demonstrated an improvement of the final human parsing model on the pruned set using complex pairwise relationships between parts.

7. REFERENCES

[1] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, 2005. 1, 2

[2] Yi Yang and Deva Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *CVPR*, 2011. 1, 2, 4, 6

[3] Yang Wang, Duan Tran, and Zicheng Liao, "Learning hierarchical poselets for human parsing," in *CVPR*, 2011. 1, 2, 3, 4, 5, 6

[4] Vittorio Ferrari, Manuel Marín-Jiménez, and Andrew Zisserman, "Progressive search space reduction for human pose estimation," in *CVPR*, 2008. 1, 2, 3, 5

[5] Benjamin Sapp, Alexander Toshev, and Ben Taskar, "Cascaded models for articulated pose estimation," in *ECCV*, 2010. 1, 2, 3, 5, 6

[6] Duan Tran and David Forsyth, "Improved human parsing with a full relational model," in *ECCV*, 2010. 1, 5

[7] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *CVPR*, 2009. 1, 2, 6

[8] Deva Ramanan, "Learning to parse images of articulated bodies," in *NIPS*, 2006. 1, 2, 5, 6

[9] Xiaofeng Ren, Alexander Berg, and Jitendra Malik, "Recovering human body configurations using pairwise constraints between parts," in *ICCV*, 2005. 1

[10] Tai-Peng Tian and Stan Sclaroff, "Fast globally optimal 2d human detection with loopy graph models," in *CVPR*, 2010. 1

[11] Yang Wang and Greg Mori, "Multiple tree models for occlusion and spatial constraints in human pose estimation," in *ECCV*, 2008. 1

[12] Deva Ramanan and Cristian Sminchisescu, "Training deformable models for localization," in *CVPR*, 2006. 2

[13] Pawan Kumar, Andrew Zisserman, and Philip H. S. Torr, "Efficient discriminative learning of parts-based models," in *ICCV*, 2009. 2

[14] Vivek Kumar Singh, Ram Nevatia, and Chang Huang, "Efficient inference with multiple heterogenous part detectors for human pose estimation," in *ECCV*, 2010. 2

[15] Sam Johnson and Mark Everingham, "Combining discriminative appearance and segmentation cues for articulated human pose estimation," in *MLVMA*, 2009. 2

[16] Greg Mori, Xiaofeng Ren, Alyosha Efros, and Jitendra Malik, "Recovering human body configuration: Combining segmentation and recognition," in *CVPR*, 2004. 2

[17] Greg Mori, "Guiding model search using segmentation," in *ICCV*, 2005. 2

[18] Benjamin Sapp, Chris Jordan, and Ben Taskar, "Adaptive pose priors for pictorial structures," in *CVPR*, 2010. 2

[19] Praveen Srinivasan and Jianbo Shi, "Bottom-up recognition and parsing of the human body," in *CVPR*, 2007. 2

[20] Sam Johnson and Mark Everingham, "Clustered pose and nonlinear appearance models for human pose estimation," in *BMVC*, 2010. 2

[21] Min Sun and Silvio Savarese, "Articulated part-base model for joint object detection and pose estimation," in *ICCV*, 2011. 2

[22] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik, "Detecting people using mutually consistent poselet activations," in *ECCV*, 2010. 2

[23] Lubomir Bourdev and Jitendra Malik, "Poselets: Body part detectors training using 3d human pose annotations," in *ICCV*, 2009. 2

[24] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, 2001. 4