Integrating Grammar and Segmentation for Human Pose Estimation

Brandon Rothrock[1], Seyoung Park[1] and Song-Chun Zhu[1,2]

[1]Department of Computer Science, University of California, Los Angeles
[2]Department of Statistics, University of California, Los Angeles
{rothrock, seypark}@cs.ucla.edu, sczhu@stat.ucla.edu

Abstract

In this paper we present a compositional and-or graph grammar model for human pose estimation. Our model has three distinguishing features: (i) large appearance differences between people are handled compositionally by allowing parts or collections of parts to be substituted with alternative variants, (ii) each variant is a sub-model that can define its own articulated geometry and context-sensitive compatibility with neighboring part variants, and (iii) background region segmentation is incorporated into the part appearance models to better estimate the contrast of a part region from its surroundings, and improve resilience to background clutter. The resulting integrated framework is trained discriminatively in a max-margin framework using an efficient and exact inference algorithm. We present experimental evaluation of our model on two popular datasets, and show performance improvements over the state-of-art on both benchmarks.

1. Introduction

Accurate human pose estimation from static images has many practical applications from automated surveillance to human-computer interaction. Humans in particular can appear in a wide range of poses, body proportions, clothing, and backgrounds. One of the key modeling challenges is to effectively represent these variabilities, and capture the contextual relationships on how parts vary together. Our approach aims to address these problems by combining four key aspects: compositional parts, articulated geometry, context-sensitive part compatibility, and background modeling.

Compositional Parts: The fundamental difference between grammar models [27] and conventional hierarchical models [8] is the notion that an object can be composed from its parts in multiple ways. These compositions can occur hierarchically, allowing the grammar to represent a very large space of possible configurations using a small number of concise rules.

Articulated Geometry and **Part Compatibility**: Articulation is a compatibility relation restricting the position and orientation of a pair of parts such that they align with a common hinge point between them, and are in a plausible orientation relative to each other. The majority of articulated models in the literature rely exclusively on this type of relation, either between part pairs [8, 25] or higher-order cliques [21]. There are many other types of part compatibilities, however, that can be exploited. Part size, for example, once normalized by object scale, can vary greatly between examples due to perspective effects or body proportion differences between people. In these cases, the part variations are not independent and largely controlled by image or object-level factors such as viewpoint or body type. These factors can be encoded into our model by defining multiple variants for a compositional part, each specific to a viewpoint, body type, appearance, etc. Context-sensitive compatibility relations are then applied to the relative position and orientation (articulated geometry), relative scale, and cooccurrence between part variants. Furthermore, each compositional variant defines its own articulated hinge points, providing an implicit compatibility between the appearance of that variant and the locations where neighboring parts can attach to it. For example, the frontal-view torso has a wide appearance with hinge points for the arms and legs near the sides, whereas the side-view torso has a narrow appearance with hinge points near the centerline. Selecting the appropriate torso will depend on the torso appearance, alignment of the limbs to the joint locations for each torso variant, and the cooccurrence compatibility between the torso and limb variants.

Background modeling: Many parts of the body have very weak local appearance structure. Forearms, for example, have no prominent local features other than noisy parallel edges which also tend to occur frequently in natural images. To complicate matters, edge features are often

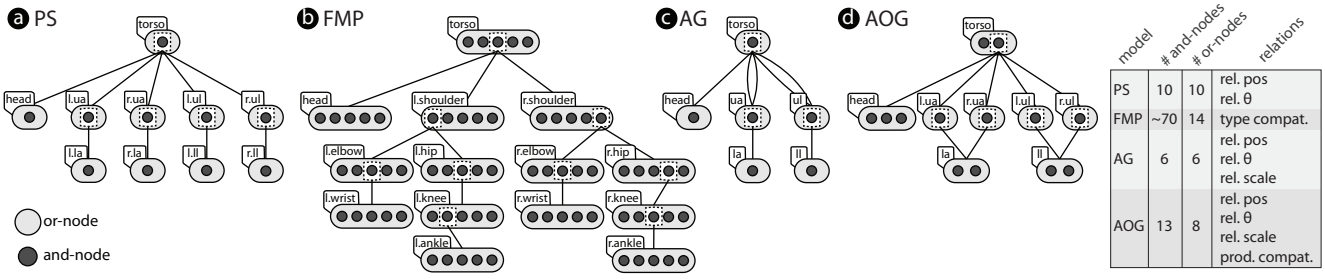

Figure 1. **Comparison of model structure**: Several common models for pose estimation are shown using an and-or graph notation. And-nodes represent distinct part appearance models, while or-nodes can be treated as a local mixtures of and-nodes. Edges represent the contextual relations between parts, which are specified for each model using the table on the right. Pictorial structures [8] (a) has a fixed structure with no shared parts, and uses conventional articulation relations over relative position and orientation. The flexible mixture-of-parts model [25] (b) emulates articulation with a large number of orientation-specific parts and mixtures, using relations only between mixture selections (types). Our baseline and-graph model (c) has similar structure and relations with PS but shares parts between left and right sides and uses relative scale relations. Our final and-or-graph model (d) extends (c) by utilizing several part variants, and compatibility relations between variants (productions).

locally normalized, which produces strong responses in textured regions and often leads to spurious detections in these regions. To combat this, we augment our part appearance models to compute a contrast between the part interior, and the region distributions of the adjoining background. The resulting part appearance model helps eliminate spurious detections in clutter, and as a result improves localization performance.

To justify our approach, we evaluate a simplified case of our grammar omitting the use of part variants or background information (model AG in Fig.1(c)). We demonstrate that this baseline still performs competitively among recent techniques, and is surpassed substantially by the full model. Evaluation results are presented for the PARSE and Leeds datasets, where we demonstrate state-of-art performance on both.

2. Related Work

Image grammar models have been a topic of interest for many years [10, 27], however, there has been limited success in getting these models to perform competitively over their fixed-structure counterparts on realistic problems. Previous work using grammars for human pose modeling include template grammars to parse rich part appearances [3, 16], super pixel based composition grammars for human region parsing [18], and boundary contour grammars for parsing human outlines [26]. [11] extends the popular discriminative deformable part model into a grammar formalism used for human detection, but not part localization. The hierarchical mixture model of [25] differs from our model by replacing articulation geometry with keypoint mixtures, and does not allow reusable or reconfigurable parts. [19] also uses hierarchical mixtures, except with coarse-to-fine appearance templates. Mixtures of higher order geometries are explored in [21] using a latent variable geometry model.

Fixed structure models for human pose estimation generally fall into the family of pictorial structures models [8, 6, 1, 17], that use a kinematic tree of parts to represent the body. These models are popular due to their relative simplicity and computational efficiency. Our model can be viewed as a generalization of these models, as each part composition can be treated as a local pictorial structure model nested in the grammar. Like all tree-structured models, these techniques tend to suffer from the double counting problem where multiple parts can match to the same region of the image due to their conditional independence. These models also do not handle self-occlusion particularly well, as they are forced to explain all parts in the image, even when some are not visible. Poselets [2], are difficult to categorize as a model type, and utilizes a voting scheme of pose-specific parts to interpolate a pose without an explicit model of the body kinematics. The hierarchical variant [24] does, however, incorporate stronger geometric constraints.

The use of image-specific **background models** to improve human pose estimation is an idea that has been revisited many times in recent literature. An iterative learning scheme for CRF appearance models was presented in [15], which incrementally refines a generic part model using image-specific appearance evidence. The work of [9] utilizes a greedy search space reduction strategy by computing GrabCut segmentations from the bounding boxes produced by a generic upper body detector. Most similar to our approach is the work of [12], which efficiently learns a local pixel-based color segmentation model for each proposed part location. Our model, by comparison, uses a global image segmentation as a reference distribution to compute part-based appearance features.

3. Articulated Grammar Model

3.1. And-Or Graph Grammar

Our articulated grammar model provides a probabilistic framework to decompose the body into modular parts while maintaining the articulated kinematics. The grammar takes the form of an and-or graph, denoted as $\mathcal{G} = (S, P, s_0)$, where S is a set of symbols (or-nodes), P is a set of productions (and-nodes), and s_0 is the root symbol. Each production $p \in P$ takes the form $(\alpha \rightarrow \beta, t, R)$. We refer to $\alpha \in S$ as the proximal symbol, and the set $\beta \subset S$ as the distal symbols. t is an appearance template for α. R is a set of probabilistic relations that control the spatial geometry and compatibility between α and β, and thus expresses contextual information about neighboring parts. Fig.1 illustrates our and-or graph models, as well as several other models for comparison. Each and-node production is drawn as a dot inside its corresponding proximal or-node symbol. For clarity, child edges are only drawn for a selected subset of and-nodes, indicated by a dotted box.

Unlike conventional grammars that only connect to the data through the terminal symbols, our grammar defines an appearance model for every production. For this reason, we require at least one production to expand every symbol. For a terminal symbol, a production of the form $(\alpha \rightarrow \emptyset, t, R)$ is used to provide an appearance and geometry model for the proximal symbol without any further decomposition. Each symbol can be expanded by multiple productions to provide different explanations for not just the appearance of that symbol, but also the geometry and compatibility between the symbol and its constituents.

Part sharing occurs whenever two or more productions use the same distal symbol. The advantages of part sharing are threefold: the resulting model has fewer parameters to learn, shared parts inherently have more training examples, and inference computation can also be shared for these parts. Furthermore, both terminal and nonterminal symbols can be shared, resulting in a potentially large reduction in both model complexity and computational time. For our model, we share the left and right limbs as shown in Fig.1(c,d).

A parse tree pt is constructed from $\mathcal{G}$ by recursively selecting productions to expand symbols starting from the root symbol s_0. Each node in the parse tree $v \in V(pt)$, corresponds to a part with state variables $(x, y, \theta, s, \omega)$, where x, y is the pixel location of the part center, θ is a discrete orientation, s is the part scale, and ω indicates the production that decomposed this part symbol. Similarly, $(v_i, v_j) \in E(pt)$ enumerates the pairs of proximal to distal parts for each production used in the parse. The relations R within each production consist of five distinct types of potential functions:

$f^a(v, I)$	appearance score
$f^{g1}(v)$	geometry orientation score
$f^{g2}(v_i, v_j)$	geometry articulation score
$f^{c1}(v)$	production bias
$f^{c2}(v_i, v_j)$	production compatibility score

We wish to learn the posterior distribution on parses, which we write as the following Gibbs distribution

$$p(pt|I) \propto p(pt)p(I|pt) = \frac{1}{Z}\exp\{-\mathcal{E}(pt, I)\}$$
$$\mathcal{E}(pt, I) = \sum_{v \in V(pt)} f^a_\omega(v, I) + f^{g1}_\omega(v) + f^{c1}_\omega(v) + \sum_{(v_i, v_j) \in E(pt)} f^{g2}_{\omega_i}(v_i, v_j) + f^{c2}_{\omega_i}(v_i, v_j). \quad (1)$$

Each production has a model weight vector corresponding to each of the potential functions $\lambda_i = (\lambda^a_i, \lambda^{g1}_i, \lambda^{g2}_i, \lambda^{c1}_i, \lambda^{c2}_i) : \forall p_i \in P$. The weight vector of the full grammar model is expressed as a concatenation of the production weights $\lambda = \{\lambda_i : i = 1..|P|\}$. Each of these potentials and their corresponding parameters are described in detail in the following sections.

3.2. Articulated Geometry

Each symbol in the grammar is assigned a canonical length and width learned from the part annotations. The geometry of each part in a parse can then be computed by retrieving the canonical dimensions corresponding to the proximal symbol of production ω, centering this rectangle at location (x, y) in the image, rotating by θ, and rescaling by s. Orientation is discretized, typically to 24 increments. The scale corresponds to the index of an image pyramid level, which determines the scale multiplier.

For each articulated pair (v_i, v_j) within each production ω, the hinge point for which these two parts articulate around is estimated from the training annotations by least-squares as in [8]. We now have two coordinate transformations, denoted $T^p_\omega(v_i)$ and $T^d_\omega(v_j)$, to compute the ideal hinge location from either the proximal or distal part states respectively. When two part are perfectly articulated, $T^p_\omega(v_i) = T^d_\omega(v_j)$. This alignment is rarely perfect, however, and we assume the displacement from the ideal hinge location is normally distributed.

The distribution over part orientations can be viewed as a mixture model over discrete orientations. Let k be the number of discrete orientations. The weights λ^{g1}_ω are the mixing weights for each orientation, specific to production ω. The feature vector $\phi^{g1}(\theta)$ is a unit vector of length k with 1 at index θ and zero elsewhere. The geometry orientation score is therefore

$$f^{g1}(v) = \langle \lambda^{g1}_\omega, \phi^{g1}(\theta) \rangle. \quad (2)$$

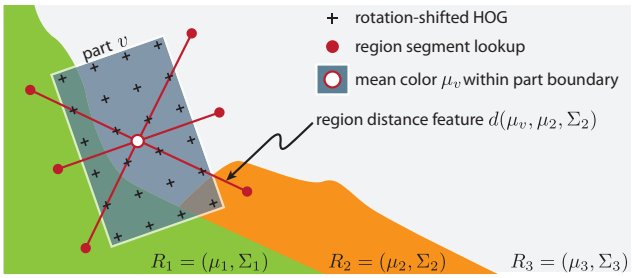

Figure 2. **Part appearance template**: The template utilizes features from both foreground and background. Foreground features use a rotation-shifted variant of HOG [4] collected along a uniform grid, as well as the mean color within the part boundary. Background features consist of distance measures between the mean part color and adjoining external region segments. The template defines multiple background sample points around the perimeter of the part, each of which retrieves the region segment that contains the point and compares it with the interior mean color.

The articulation score consists of three components: hinge displacement, relative scale, and relative orientation. We denote the squared hinge displacement as $dl_{ij}^2 := ||T_{\omega_i}^p(v_i) - T_{\omega_i}^d(v_j)||^2$. Similarly, we also treat the relative scale between parts to be normally distributed and denote the squared scale difference as $ds_{ij}^2 := (s_i - s_j)^2$. The relative orientation is computed as $d\theta := ((\theta_j - \theta_i) \mod k)$. The feature response for relative orientation $\phi^{g2}(v_i, v_j)$ is a unit vector of length k with 1 at index $d\theta$ and zero elsewhere. The articulation score is then

$$f^{g2}(v_i, v_j) = \langle \lambda_{\omega_i}^{g2}, [dl_{ij}^2 \;\; ds_{ij}^2 \;\; \phi^{g2}(v_i, v_j)^\top]^\top \rangle. \quad (3)$$

3.3. Part Compatibility

Part compatibility is the preference of selecting one production over another to explain the same symbol in a parse. The model employs two types of compatibility. The first is a unary bias on each production, analogous to the production frequencies in a stochastic context-free grammar. This bias parameter is a scalar value for each production, and the compatibility potential is simply

$$f^{c1}(v) = \lambda_\omega^{c1}. \quad (4)$$

The second is a pairwise production compatibility between neighboring parts in the parse. The compatibility weights are a matrix of dimension $|P| \times |P|$. The vector λ_ω^{c2} is the matrix row corresponding to production ω, and represents the compatibility of all distal productions with ω. The production compatibility potential is then

$$f^{c2}(v_i, v_j) = \lambda_{\omega_i}^{c2}[\omega_j]. \quad (5)$$

3.4. Appearance Model and Segmentation

Each production defines an appearance template that specifies where to extract features responses from the image for a given part state, as illustrated in Fig.2. To compute part appearances responses at different scales, a fixed-sized template is applied to different levels of the image pyramid. Different orientations are handled by rotating the template, then adjusting the features to compensate for the rotation if necessary. Two types of features are used: a gradient-based edge feature, and a color-based region feature.

The edge features are a variation of the popular HOG feature [4]. In order to compute feature responses of rotated parts, however, the HOG features must be computed densely such that gradients histograms are pooled around the neighborhood of every pixel instead of predetermined cells. Histograms are collected along a uniform grid in the reference frame of the part, illustrated by the crosses in Fig.2. If the part is rotated, then the histogram bins must be shifted to match the orientation of the part. The number of HOG bins is therefore selected to match the number of discrete part orientations.

The region features measure how distinct the foreground part region is from the surrounding background. We represent the image background as a collection of large disjoint regions, where the appearance within each region is well explained using a multivariate Gaussian in $L^*u^*v^*$ color space. Furthermore, we assume that the background regions are large compared to the size of the foreground parts and treat the background process as independent of the foreground. This independence is chosen to avoid the intractable computation of reestimating the background segments for every part state.

Let Λ denote the pixel lattice of image I, which is partitioned into K disjoint regions $\bigcup_{i=1}^K \mathcal{R}_i = \Lambda$, $\bigcap_{i=1}^K \mathcal{R}_i = \emptyset$. The segmentation of the image is represented as $\mathcal{S} = (K, \{(\mathcal{R}_i, \mu_i, \Sigma_i);\ i = 1, 2, ...K\})$. Each region is assumed to be generated independently and normally distributed, thus the image likelihood is $p(I|\mathcal{S}) = \prod_{i=1}^K \mathcal{N}(\mu_i, \Sigma_i)$. A prior model $p(\mathcal{S})$ encourages the number of regions to be small, region volumes to be large, and boundaries smooth. The optimal segmentation maximizes the posterior $p(\mathcal{S}|I) \propto p(I|\mathcal{S})p(\mathcal{S})$. We adopt the same prior model and data-driven MCMC approach of [23] to compute this segmentation using $scale = 1.0$. Please refer to the original work for a full explanation of the prior model and its parameters.

Finally, the region feature is computed as the Mahalanobis distance between the foreground mean μ_v and a background region (μ_i, Σ_i)

$$d(\mu_v, \mu_i, \Sigma_i) = (\mu_v - \mu_i)^\top \Sigma_i^{-1} (\mu_v - \mu_i). \quad (6)$$

This distance can be interpreted as the negative log-probability that the average pixel in the foreground region is generated by the background process. To account for the possibility that multiple background region segments can adjoin the part, the template defines multiple region features that are equally spaced around the part periphery, as

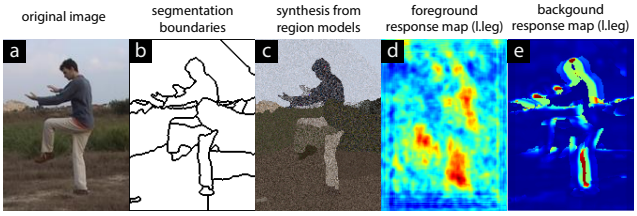

Figure 3. **Segmentation model**: Segmented regions are shown in (b), and a resynthesized image sampled from the region models is shown in (c) to illustrate the model fit. Score maps from the trained model of the l.leg part in the vertical orientation are shown using only HOG features in (d) and only region distance features in (e). Due to local normalization, spurious foreground responses tend to appear particularly around textured regions, whereas the background feature is far more stable in these regions.

shown by the circles in Fig.2. The output of the segmentation model is illustrated in Fig.3, as well as visualizations of template scores using either HOG or the region distance features in isolation.

The full appearance response vector $\phi^a(I,t,v)$ can now be computed as a concatenation of responses from each rotation-shifted gradient histogram feature, and region distance feature in the template. The appearance score is then

$$f^a(v,I) = \langle \lambda^a_\omega, \phi^a(I,t_\omega,v)\rangle. \tag{7}$$

4. Inference

Relations in the grammar are always between proximal and distal parts within the same production, resulting in a tree factorization of the full grammar model. This lends itself naturally to a dynamic programming type algorithm that computes optimal part configurations bottom-up. The basic unit of computation is computing a maximal score map for the proximal part of a production. Each of the distal parts are conditionally independent given the proximal part, and can be maximized individually. The maximal score map for part state v_i given production ω_i can be expressed recursively as

$$M(v_i|\omega_i) = f^a_{\omega_i}(v_i,I) + f^{g1}_{\omega_i}(v_i) + f^{c1}_{\omega_i} + \sum_{(v_i,v_j)\in R_{\omega_i}} \max_{v_j} \left[f^{g2}_{\omega_i}(v_i,v_j) + f^{c2}_{\omega_i}(v_i,v_j) + M(v_j|\omega_j)\right]. \tag{8}$$

Although the production for part v_i is fixed, we must maximize over the full state of the distal parts v_j, including the distal production. The maximization over positions (x_j, y_j) can be computed very efficiently using distance transforms [7] that have linear complexity in the number of positions. The maximization over scale s_j, orientation θ_j, and production ω_j each require quadratic time to compute. The state space for these remaining variables is still quite small, however, and the computation is tractable.

To infer the maximal scoring parse, we recurse through the grammar starting from the root symbol s_0. Terminal symbols have no distal parts, and their maximal score maps consist of only the appearance and unary potentials. Once the maximal score maps are computed for every production, the maximal parse score can be obtained by maxing over all productions that have the root symbol as the proximal part

$$\max_{p_j \in P \ s.t.\ \alpha_j = s_0} \max_{v_i} M(v_i|p_j). \tag{9}$$

The parse tree can be recovered by replacing the $\max$ operators with $\arg\max$ and backtracking through the optimal state maps.

5. Learning

The score of a parse can always be expressed as the inner product of the full model weight vector and a response vector for the entire parse $f^{\mathcal{G}}(pt,I) = \langle \lambda, \phi(pt,I)\rangle$. The model weights λ parameterizes a family of parsers that output the maximal scoring parse $F^{\mathcal{G}}_\lambda(I) = \arg\max_{pt} f^{\mathcal{G}}(pt,I)$ for a given grammar. We define the learning task as the search for a weight vector such that the empirical risk of the associated parser is minimized, which is computed as the expected loss on the training dataset D. Let $\bar{pt}$ be the ground truth parse. The optimal weights are

$$\lambda^* = \arg\min_\lambda E_{(\bar{pt},I)\sim D}[L(F^{\mathcal{G}}_\lambda(I), \bar{pt})] \tag{10}$$

The loss is defined on the structured output space of parses, and must measure the quality of a predicted parse against the ground truth parse. In a general grammar, these parses may have different structure or a different number of parts, making the formulation of such a loss sometimes difficult. All parses from the grammars we define here, however, have the same number of parts and the same branching structure which allows us to compute loss as the sum of part-wise terms. Our loss is motivated by the PCP evaluation metric [6], which computes a score based on the proximity of the part endpoints to the ground truth endpoints. A part is typically considered detected when the PCP score is under 0.5. The loss function is

$$L(pt,\bar{pt}) = \frac{1}{|V(pt)|} \sum_{v\in V(pt)} \min\left(2\cdot \mathrm{pcp}(v,\bar{v}), 1\right) \tag{11}$$

and is bounded between 0 and 1 taking the value 0 only when identical to the ground truth.

To make the learning computationally tractable, we instead minimize a convex upper bound to this loss using the so-called margin-scaled structural hinge loss from [20], resulting in the following max-margin structural SVM objective function

$$\min_\lambda \frac{1}{2}||w||^2 + \frac{C}{|D|}\sum_{i=1}^{|D|}\xi_i \tag{12}$$
$$s.t.\ \ \lambda^\top\left[\phi(\bar{pt}_i, I_i) - \phi(pt, I_i)\right] \geq L(pt,\bar{pt}_i) - \xi_i$$
$$\forall pt \in \Omega_{\mathcal{G}}, \forall i.$$

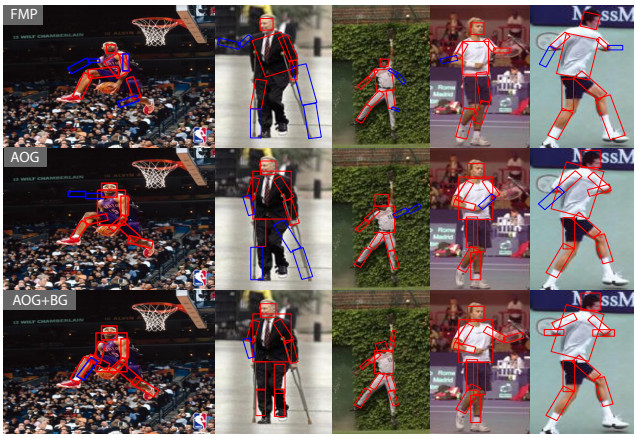

Figure 4. **Influence of region features**: A selection of results from the AOG and AOG+BG models compared with FMP [25]. Textured regions are problematic for both FMP and AOG models, leading to frequent spurious limb detections in these regions. The AOG+BG model includes terms to favor part regions that are distinct from their adjoining background process, and can correctly localize many of these parts.

Due to the exponential number of constraints, it is intractable to minimize this expression directly. Instead, it is still provably efficient to solve this minimization incrementally by adding only the most violated constraints at each iteration, using the following maximization as the so-called separation oracle

$$\hat{pt}_i = \arg\max_{pt} \lambda^\top \phi(pt, I_i) + L(pt, \bar{pt}_i). \qquad (13)$$

This maximization is commonly referred to as a loss-adjusted inference problem. The complexity of this maximization depends on the formulation of the loss function. This is the primary reason we choose an additive loss function, which can be incorporated into the existing inference algorithm in a relatively straightforward manner without impacting the computational complexity.

It can be more clearly seen that this objective is an upper bound for the risk by rearranging the terms of the most violated constraints $\xi_i \geq L(pt, \bar{pt}_i) + \lambda^\top \phi(\hat{pt}_i, I) - \lambda^\top \phi(\bar{pt}_i, I)$. Because the score of the parse $\lambda^\top \phi(\bar{pt}_i, I)$ can never be greater than the score of the maximal parse $\lambda^\top \phi(\hat{pt}_i, I)$, the right-hand-side of the expression can never be lower than the loss.

This minimization can be solved by a multitude of methods. A dual coordinate descent solver was implemented for [25], and the cutting plane method of [22] is also commonly used. For our implementation, we maximize the dual objective using a conventional QP solver.

6. Evaluation

We train and evaluate our method using three different grammar models to illustrate the impact on performance from the addition of reconfigurable parts as well as the background model. For all cases, we discretize the state space of the parts to be 25% of the image width and height, and use 24 part orientations.

AG: And-graph grammar is our baseline model, shown in Fig.1(c), and is the simplest possible model in our framework to represent the full articulated body. Each symbol has only one production, and all limb parts are shared between the left and right sides. This construction is equivalent to a pictorial structures model (Fig.1(a)) with shared parts.

AOG: And-or-graph grammar, shown in Fig.1(d), using productions { front, side } for torso, { left, front, right } for head, and { visible, occluded } for both l.arm and l.leg. Separate symbols are used for left and right u.leg as well as u.arm because side-specific features tend to be strong for these parts. The l.leg and l.arm symbols are still shared between sides.

AOG+BG: This is the same grammar as AOG, but with the addition of the background terms. These terms are only included on the the productions for l.arm and l.leg. To illustrate the influence of these features, Fig.4 shows several examples where the top scoring pose erroneously matches to a strong edge with poor region support, but is corrected when retraining with the background feature.

6.1. Evaluation Protocol

Unfortunately there are several competing evaluation protocols for articulated pose estimation scattered throughout the literature that often produce significantly different results. We adopt the PCP method described in [6], which appears to be the most common variant. This protocol defines a part as detected if the average distance between the centerline endpoints of the proposal and ground truth is less than 50% of the ground truth part length. Because there may be multiple people in a test image, the protocol selects for evaluation the highest scoring skeleton within a window defined by the head and torso. All competing methods used for comparison are evaluated using the same protocol to the best of our knowledge.

There are two notable inconsistencies with the evaluation that are worth mentioning. First is the existence of zero-length parts in the annotations, which are impossible to detect according to this metric. Second is the inconsistency or genuine ambiguity of labeling a limb as left or right. Both datasets that we evaluate on have multiple cases where the left/right annotations are inconsistent with the rest of the dataset. Even perfect results on these examples will still get the limb parts counted as wrong because the left limb is being evaluated against the right side and vice versa. To compensate for this, for each selected skeleton we exchange the left and right labels for arm and leg separately, and take the configuration that has the highest number of correctly localized parts. We mark results evaluated in this way with

Dataset	Method	torso	head	u.leg	l.leg	u.arm	l.arm	avg
PARSE	JEa [13] (2010)	85.4	76.1	73.4	65.4	64.7	46.9	66.2
	TZN [21] (2012)	97.1	92.2	85.1	76.1	71.0	45.1	74.4
	FMP [25] (2011)	97.6	93.2	83.9	75.1	72.0	48.3	74.9
	DR [5] (2012)	-	-	-	-	-	-	77.4
	Ours (AG)	99.5	95.6	81.8	67.0	74.3	54.6	75.0
	Ours (AOG)	**100.0**	96.2	87.0	75.3	73.2	53.9	77.5
	Ours (AOG+BG)	99.5	**97.4**	**88.4**	**78.0**	**74.1**	**56.1**	**79.0**
	Ours (AOG+BG)†	99.5	97.4	89.2	78.3	74.6	56.9	79.5
Leeds	TZN [21] (2012)	95.8	87.8	69.9	60.0	51.9	32.9	61.3
	JEb [14] (2011)	88.1	74.6	74.5	66.5	53.7	37.5	62.7
	Ours (AG)	98.4	92.8	81.2	69.8	61.9	38.2	69.3
	Ours (AOG)	**98.8**	92.7	**83.9**	**74.4**	64.0	41.1	71.8
	Ours (AOG+BG)	98.3	**92.7**	83.7	73.1	**66.0**	**41.4**	**71.9**
	Ours (AOG+BG)†	98.3	92.7	86.8	78.2	70.2	45.1	75.2

Table 1. **Benchmark evaluation results**: We evaluate our baseline model (AG), grammar model (AOG), and grammar model with background features (AOG+BG) on the PARSE and Leeds datasets. The performance of our AOG+BG model outperforms all known methods for all parts on both datasets. The † symbol indicates the use of a modified evaluation protocol, see text for details.

a † symbol, all other results are evaluated in the standard way.

6.2. Benchmarks

PARSE: Introduced by [15], this dataset consists of 100 training and 205 testing images. For each part, we provide an additional annotation to indicate a production label. We observe a performance gain of 2.5% between the AOG and baseline AG model, and a 4.0% gain between AOG+BG and AG. Furthermore, the AOG+BG model outperforms the state-of-art for all parts individually, as well as an average gain of 1.6% over the current best method.

Leeds: Introduced by [13], this dataset consists of 1000 images each for training and testing. In the same manner as PARSE, we provide an additional production label to each part. Our AOG+BG model also outperforms the state-of-art for all parts on this dataset, by an average gain of 9.2%. The contribution of the background feature is minimal on this dataset, however, which we believe may be attributed to the narrow crop margins and general lack of large background regions.

7. Conclusions

We present a framework for human pose estimation using an articulated grammar model, as well as a simple approach to integrate a background model into the grammar that can improve localization performance in cluttered scenes. We also describe a training strategy for learning the model from an empirical risk minimization perspective. Our technique is evaluated on two challenging benchmark datasets with superior performance to the current state-of-art in both cases. Furthermore, we demonstrate consistent per-part performance improvements of adding reconfigurable parts over a baseline fixed-structure model using the same part representations and learning, and additional gains from incorporating the background model. Although we focus specifically on the task of human pose estimation, our model is not tailored to the class and is likely applicable to other highly deformable object classes.

8. Acknowledgements

This project is supported by DARPA MSEE project FA 8650-11-1-7149, MURI ONR N00014-10-1-0933, NSF CNS 1028381, and NSF IIS 1018751.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, pages 1014–1021, 2009.

[2] L. D. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, pages 1365–1372, 2009.

[3] H. Chen, Z. Xu, Z. Liu, and S. C. Zhu. Composite templates for cloth modeling and sketching. In *CVPR (1)*, pages 943–950, 2006.

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

[5] C. Desai and D. Ramanan. Detecting actions, poses, and objects with relational phraselets. In *ECCV (4)*, pages 158–172, 2012.

[6] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *BMVC*, 2009.

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. In *Cornell University*, 2004.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, Jan. 2005.

Figure 5. **Example results**: Bounding boxes are drawn for each of the 10 parts considered for evaluation. Parts localized correctly are shown in red, and incorrectly in blue. The top two rows are examples of challenging poses with perfect localization scores from the PARSE and Leeds datasets respectively. The bottom row illustrates some of the failure modes on both datasets. Common failures are due to double counting, occlusion, and background confounders, which is compounded by extreme perspective, foreshortening, and crumpled poses.

[9] V. Ferrari, M. M. Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, pages 1–8, 2008.

[10] K. S. Fu. A step towards unification of syntactic and statistical pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(3):398–404, 1986.

[11] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Object detection with grammar models. In *NIPS*, 2011.

[12] S. Johnson and M. Everingham. Combining discriminative appearance and segmentation cues for articulated human pose estimation. In *ICCV Workshop on Machine Learning for Vision-based Motion Analysis*, 2009.

[13] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, pages 1–11, 2010.

[14] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR*, pages 1465–1472, 2011.

[15] D. Ramanan. Learning to parse images of articulated bodies. In *NIPS*, pages 1129–1136, 2006.

[16] B. Rothrock and S.-C. Zhu. Human parsing using stochastic and-or grammars and rich appearances. In *ICCV Workshops*, pages 640–647, 2011.

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

[18] P. Srinivasan and J. B. Shi. Bottom-up recognition and parsing of the human body. In *CVPR*, pages 1–8, 2007.

[19] M. Sun and S. Savarese. Articulated part-based model for joint object detection and pose estimation. In *ICCV*, pages 723–730, 2011.

[20] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS*, 2003.

[21] Y. Tian, C. L. Zitnick, and S. G. Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In *ECCV (5)*, pages 256–269, 2012.

[22] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[23] Z. Tu and S. C. Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):657–673, 2002.

[24] Y. Wang, D. Tran, and Z. Liao. Learning hierarchical poselets for human parsing. In *CVPR*, pages 1705–1712, 2011.

[25] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392, 2011.

[26] L. Zhu, Y. Chen, Y. Lu, C. Lin, and A. L. Yuille. Max margin and/or graph learning for parsing the human body. In *CVPR*, 2008.

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