Articulated Part-based Model for Joint Object Detection and Pose Estimation

Min Sun Silvio Savarese
Dept. of Electrical and Computer Engineering, University of Michigan at Ann Arbor, USA
{sunmin,silvio}@umich.edu

Abstract

Despite recent successes, pose estimators are still somewhat fragile, and they frequently rely on a precise knowledge of the location of the object. Unfortunately, articulated objects are also very difficult to detect. Knowledge about the articulated nature of these objects, however, can substantially contribute to the task of finding them in an image. It is somewhat surprising, that these two tasks are usually treated entirely separately. In this paper, we propose an Articulated Part-based Model (APM) for jointly detecting objects and estimating their poses. APM recursively represents an object as a collection of parts at multiple levels of detail, from coarse-to-fine, where parts at every level are connected to a coarser level through a parent-child relationship (Fig. 1(b)-Horizontal). Parts are further grouped into part-types (e.g., left-facing head, long stretching arm, etc) so as to model appearance variations (Fig. 1(b)-Vertical). By having the ability to share appearance models of part types and by decomposing complex poses into parent-child pairwise relationships, APM strikes a good balance between model complexity and model richness. Extensive quantitative and qualitative experiment results on public datasets show that APM outperforms state-of-the-art methods. We also show results on PASCAL 2007 - cats and dogs - two highly challenging articulated object categories.

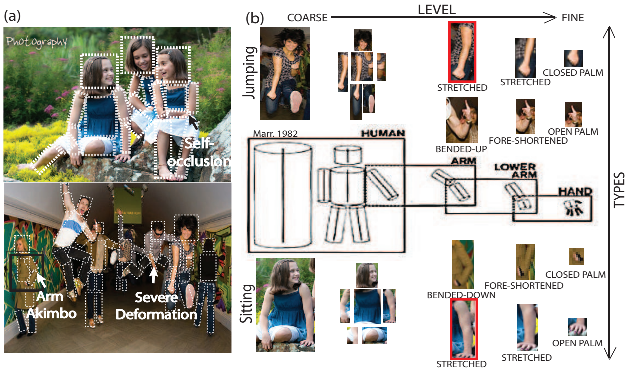


Figure 1. Panel (a) shows large appearance variation and part deformation of articulated objects (people) with different poses (sitting, standing, and jumping, etc). (b) we propose a new model for jointly detecting objects and estimating their pose. Inspired by Marr[14], our model recursively represents the object as a collection of parts from a coarse-to-fine level (e.g., see horizontal dimension) using a parent-child relationship with multiple part-types (e.g., see vertical dimension). We argue that our representation is suitable for "*taming*" such pose and appearance variability.

1. Introduction

Detecting and estimating the pose (i.e., detecting the location of every body parts) of articulated objects (e.g., people, cats, etc.) has drawn much attention recently. This is primarily the result of an increasing demand for an automated understanding of the actions and intentions of objects in images. For example, person detection and pose estimation algorithms have been applied to the fields of automotive safety, surveillance, video indexing, and even gaming. Most of the existing literature treats object detection and pose estimation as two separate problems. On the one hand, most of the state-of-the-art object detectors [8, 13, 19, 2] do not focus on localizing articulated parts (e.g., location of heads, arms, etc.). Such methods have shown excellent results on rigid vehicle-type objects (e.g., cars, motorbikes, etc) but less so on the articulated ones (e.g., human or animals) [7]. On the other hand, most pose estimators [12, 20, 10, 5, 16, 15, 11] assume that either the object locations, the object scales, or both are predetermined by either a specific object detector, or given manually. We argue that these two problems are two faces of the same coin and must be solved jointly. The ability to model parts and their relationship allows to identify objects in arbitrary configurations (e.g., jumping and sitting, see Fig. 1) as opposed to canonical ones (e.g., walking and standing). In turn, the ability to identify the object in the scene provide strong contextual cues for localizing object parts.

Some recent works partially attempt to solve the problems in a joint fashion. [1] combines a tree-model with discriminative part detectors to achieve good pose estimation and object detection performance. However, good detection performance is only demonstrated on the TUD-UprightPeople and TUD-Pedestrians datasets [1], which have fairly restricted poses. Alternatively, [4, 3] propose a holistic representation of human body using a large number of overlapping parts, called poselets, and achieve the best performance on PASCAL 2007∼2010 person category. However, poselet can only generate a distribution of pos-

2011 IEEE International Conference on Computer Vision
978-1-4577-1102-2/11/$26.00 ©2011 IEEE

sible locations for each part's end points independently, which make it difficult to infer the best joint configuration of parts for the entire object.

Our Model. We present a new model for jointly detecting articulated objects and estimating their part configurations (Fig. 1(a)). Since the building blocks of this model are object parts and their spatial relationship in the image, we call it the Articulated Part-based Model (APM). Our approach based on APM seeks to satisfy the following properties.

Hierarchical (coarse-to-fine) Representation. Inspired by the articulated body model in the 1980s [14] which recursively represents objects as generalized cylinders at different coarse to fine levels (Fig. 1(b)), our model jointly models the 2D appearance and relative spatial locations of 2D parts (Fig. 1(b)) recursively at different Coarse-to-Fine (CF) levels. We argue that a coarse-to-fine representation is valuable because distinctive features at different levels can be used to jointly improve detection performance. For example, the whole body appearance features are very useful to prune out false positive detection from the background, whereas detail hand appearance features can be used to further reinforce or lower the confidence of the detection.

Robustness to Pose Variability by Part Sharing. Articulated objects exhibit tremendous appearance changes because of variability in: i) view point location (e.g. frontal view, side view, etc); ii) object part arrangement (e.g. sitting, standing, jumping, etc); iii) self-occlusions among object parts (Fig. 1(a)). We refer to the combination of these effects as to the *pose* of the object. Methods such as [8, 25] capture such appearance variations by introducing a number of fully independent models where each model is specialized to detect the object observed under a specific pose. Clearly such representation is extremely redundant as appearance and spatial relationship of parts are likely to be shared across different poses (e.g., a ”stretched arm” is observed in both a sitting (Top) and standing (Bottom) person as Fig. 1(b) highlights in red). While this representation may be suitable for rigid objects (for which appearance changes are mostly dictated by the view point location of the observer), it may be less so for articulated objects. In order to obtain a more parsimonious representation while keeping the ability to capture rich pose variability, we introduce the concept of ”*part-type*”. A part-type allows to characterize each part with attributes associated to semantic or geometrical properties of the part. For example a human arm can be characterized by part-types such as ”stretched” or ”fore-shortened” at a given level of the hierarchy. The introduction of part-types lets parts be shared across object poses if they can be associated to the same part-type. By having the APM to share parts, we seeks to strike a good balance between model richness (i.e., the number of distinct poses) and model complexity (i.e., the number of part-types) (Sec. 3.3).

Methods	CF	Type	Sub.	E.I
PS [10]	N	N	N	Y
Yao et. al. [23]	N	Y	N	N
Wang et. al. [21]	Y	Y	Y	N
Yang et. al. [22]	N	Y	Y	Y
Grammar [26, 24]	Y	Y	Y	N
APM (Ours)	Y	Y	Y	Y

Figure 2. A comparison of the properties satisfied by our APM model versus other models. The CF column indicates if a coarse-to-fine recursive representation is supported or not. ”Type” indicates if multiple part-types are supported or not. ”Sub” indicates if the model complexity grows sublinearly as function of the number of poses. ”E.I” indicates if exact inference is tractable (Sec. 2).

Efficient Exact Inference & learning Following the recursive structure of an APM, we use efficient dynamic programming algorithms to jointly (and exactly) infer the best object location and estimate their pose (Sec. 3.1, 3.2). We learn the parameters regulating part appearance and their relationships across coarse-to-fine levels by using a Structured Support Vector Machine (SSVM) [18] with a loss function penalizing incorrect pose estimation (Sec. 4).

Novel Evaluation metric. Because the detection and pose estimation are often performed separately, no standard method exists for evaluating algorithms that address both problems. The popular Percentage of Correctly estimated body Parts (PCP) metric measures the percentage of correctly detected parts for the objects that have been correctly detected. This is problematic in that PCP can be high while detection accuracy is low. To fix this, we propose to directly compare the recall vs False Positive Per Image (FPPI) curves of the whole object and all parts. Using this new measure as well as standard evaluation metrics, we show that APM outperforms state-of-the-art methods. We also show, for the first time, promising pose estimation results on two very challenging categories of PASCAL: cats and dogs.

The rest of the paper is organized as follows. Sec.2 describes related work. Model representation, recognition, learning, and implementation details are discussed in Sec.3, 4, and 5 respectively. Experimental results are given in Sec. 6.

2. Related Work

Pictorial Structure (PS) based methods such as [10, 8] are the most common approaches for pose estimation and object recognition. Similarly to our model, the PS object representation is part-based. Unlike ours, however, in PS's model parts are not organized at different coarse-to-fine levels with multiple part-types. Moreover, as discussed earlier, object models are learnt independently for each object pose without having the ability to share parts across poses. As a result, the number independent models used in PS grows linearly with the number of distinct poses that one wishes to capture (Fig. 2).

Our model bears some similarity to recent works (Fig. 2).

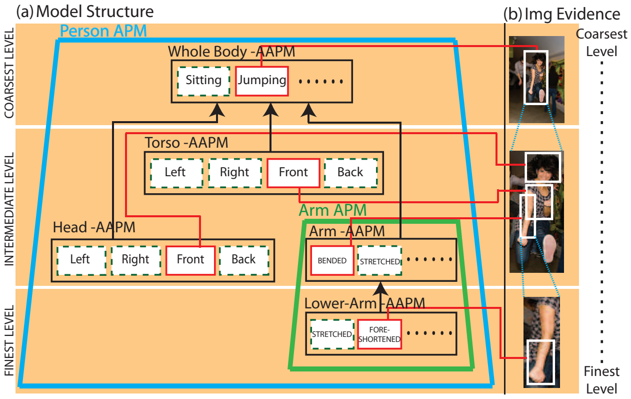


Figure 3. Graphical illustration of the recursive coarse-to-fine structure of APM. Panel (a)-Top: An APM (blue trapezoid) can be obtained by recursively combining atomic APMs (AAPM) (black boxes) such as arm-AAPM, lower-arm-AAPM, etc, into higher-level part-APM such as the arm-APM (green trapezoid). Panel (b) shows examples of selected part locations (white windows) at the different levels. The selected part-types are highlighted by red boxes in panel (a).

In [23] a procedure is presented for capturing the typical relationship between body part configurations and objects. While the concept of part type is used to increase the flexibility of the representation, pairwise relationships between body parts are not shared across classes. As a result, the number of parameters that are used to model spatial relationship grows linearly with the number of pose classes. Moreover, unlike our approach, parts are not organized in a recursive coarse-to-fine structure. [21] propose a hierarchical poselet model for both detection and pose estimation. However, the model requires loopy belief propagation algorithm for approximate inference. [22] propose a mixture of parts model which achieves outstanding performance and speed. However, because the representation is not hierarchical, it is not suitable to detect objects at a small scale.

APM is also related to grammar models for images or objects [26, 24]. In such models, the object is represented by reusing and sharing common object elements. However, these models rely on complex learning and inference procedures which can only be made tractable using approximate algorithms (Fig. 2). On the contrary, despite the sophisticated structure of APMs, we show that a tractable exact inference algorithm can be used (Sec. 3.1 and 3.2).

3. Articulated Object Representation

Given a still image containing many articulated objects (e.g., persons in Fig. 1(a)), our goal is to jointly localize the objects and estimate their poses (i.e., localize articulated parts such as arms, legs, etc).

We introduce a new model called Articulated Part-based Model (APM) to achieve this goal. In designing the APM model, we seek to meet the desiderata discussed in the Sec. 3.3 and propose a representation that is hierarchical, robust to pose variability and parsimonious. An APM for an object category (object-APM) is a hierarchical structure constructed by recursively combining primary elements called atomic APM (AAPM). An AAPM is used to represent an object part at each level of the object representation (e.g., an AAPM can represent a lower arm, the torso, or the whole body, etc.). An AAPM just models the appearance of a part and it is characterized by a number of part types (e.g., an head-AAPM is characterized by types such as left, front, etc.) (Fig. 3). AAPMs can be recursively combined into APMs by following a parent-child relationship. E.g., an arm-AAPM and a lower-arm-AAPM are subject to a parent-child relationship (a lower-arm is part of an arm) and are combined into an APM called arm-APM. As an other example, children APMs such as the arm-APM, or head-APM can be combined with their parent (the body-AAPM) and form the person-APM (Fig. 3). An APM models the part appearance of both parent and children as well as the 2D geometrical relationships between parent and children.

Since each AAPM can be characterized by several part types, and since AAPM or APMs can be *reused* toward constructing new APMs, an object-APM has the nice property of being able of capturing an exponentially large number of pose configurations by just using a few AAPMs. For instance, suppose that a person is described by 5 parts (head, torso, arm, lower-arm) (thus 5 AAPMs) and that each part is characterized by 4 types. A person-APM model can then encode up to 4^5 different poses in total by only using the 5 AAPMS. This way, the APM allows us to strike a good balance between model richness (i.e., the number of distinct object poses that the model can capture) and model complexity (i.e., the number of model parameters) (See Sec. 3.3(i)).

The structure of the APM model (i.e., number of parts, part-types, and parents-child relationships) may be predefined following the kinematic construction of the object. Given such a structure, the goal of learning is to jointly learn the appearance model for every part-type and parent-child geometric relationships so that the importance of different part-types and the discriminative power of different parent-child relationships can be automatically discovered. During recognition (inference), the goal is to determine the most likely configuration of object parts and part-types that is compatible with the observation and the learnt APM model. The next section describes how to utilize the recursive structure of APM to efficiently estimate the most likely configuration.

3.1. Recognition

Finding the best part configuration (i.e., in our case, both the part locations and types) for arbitrary part-based models corresponding to the highest likelihood or score is in general computationally intractable since the configuration space grows exponentially with the number of parts. By leveraging the recursive structure of an APM, we show that an efficient top-down search strategy for exploring pose configuration hypotheses in the image is possible. Then we show how to compute a matching score for each hypothesis with a time that is at most quadratic with the number of hypothesis per part by using a bottom-up score assignment scheme. This matching scores are used to guide the top-down search to reach the best pose configuration hypothesis. The result is an efficient inference algorithm that reaches the optimal solution in at most quadratic time.

Top-down Search strategy. The image is explored at different levels of decompositions (from coarse-to-fine) using a recursive strategy. At each level, the image is decomposed into regions (windows) and each region is associated with a part type. Based on the selected part type and the parent-child relationship, each image region is further processed and the next level of decomposition is initiated. The example below clarifies this process.

Let us consider an APM for the object *person* (Fig. 3). At the first (coarsest) level of decomposition only a single part is considered. This corresponds to the whole object (*person*). Part types are different human poses (sitting, jumping, standing, etc). The image is explored at different locations (i.e., a score is assigned at different locations following a sliding window approach) and a part type (hypothesis) is associated to each window location. E.g., the white window in Fig. 3(b) is associated with the part type jumping. Following the structure of the APM, jumping is a parent of a number of child parts (head, torso, left arm, etc), and the goal is to identify each of these child parts within the current working window. Now the next level of decomposition is initiated. Let us consider the child left-arm part as an APM. The area within the current working window is explored at different locations and each of these are associated to a left-arm part type (hypothesis). At this level, part types are, for instance, stretched or foreshortened. E.g., the white window in Fig. 3(b) is associated with the part type stretched. Following the structure of the APM, left-arm is a parent of a number of child parts (upper-arm, lower arm), and the goal is to identify each of these child parts within the current working window. This initiates the next level of decomposition. The process terminates when all the image windows are explored, all parts are processed and no additional decompositions are allowed. In the Fig. 3, the active part types across levels are highlighted by red edges. Notice that the levels of recursion depends on the structure design of the model.

Bottom-up matching score assignment. While the best hypothesis is found using a top-down strategy, the process of assigning a matching score to each hypothesis follows a bottom-up procedure. The benefit of such procedure is that all the scores can be computed in time at most quadratic to the number of hypothesis per part. Notice that special forms of geometric relationship can even be computed in linear time as in [9]. In details, each matching score is computed by combining an appearance score and a deformation score. The appearance score is obtained by matching the evidence within the working image window against the learned part type appearance model. The deformation score is obtained by: i) computing the parent-child geometrical configuration - that is, the location and orientation (angle) of a part within its parent reference frame; ii) matching this configuration with the learnt parent-child geometrical configuration. These scores are collected and combined bottom-up so as to obtain a final score that indicates the confidence that an image window (at the coarsest level) contains a person with a certain pose and part configuration. Details are explained in Sec. 3.2.

3.2. Matching Scores

Let us first introduce the parameterization of a part hypothesis in an APM. A part hypothesis is described by the location $h = (x, y, l, \theta)$ and type s of the part, where (x, y) is the part reference position (e.g.,the top-left corner of the part), (l, θ) are the part scale (coarse-to-fine) and 2D orientation, respectively. The task of joint object detection and pose estimation is equivalent to finding a set of part hypotheses $H = \{(h_0, s_0), \dots, (h_k, s_k), \dots\}$ such that the location $h = (x, y, l, \theta)$ and type s is specified for all parts.

As previously introduced, the matching scores can be divided into two classes: *appearance* and *deformation* scores. The appearance score of a specific part-type is obtained by matching the feature $\psi_a(h, I)$ extracted from the image within the window specified by the part location h against the learned appearance model A, and the score is defined as

$$f^A(h; I) = A^T \psi_a(h, I) \tag{1}$$

The deformation score is obtained by: i) computing the parent-child geometrical relationship - that is, the difference $\psi_d(h, \hat{h}) = (\Delta x, \Delta y, \Delta\theta)$ of position and orientation between the expected child hypothesis $\hat{h}$ and the actually child hypothesis h at the child reference scale; ii) matching this relationship with the learnt parent-child deformation model d. The score is defined as,

$$\begin{aligned} f^D(h, \hat{h}) &= -d^T \psi_d(h, \hat{h}) = -(d_1 \cdot (\Delta x)^2 + d_2 \cdot (\Delta x) \\ &\quad + d_3 \cdot (\Delta y)^2 + d_4 \cdot (\Delta y) + d_5 \cdot (\Delta\theta)^2 + d_6 \cdot (\Delta\theta)) \end{aligned} \tag{2}$$

where $d = (d_1, d_2, d_3, d_4, d_5, d_6)$ is the model parameter for parent-child deformation.

The final score for each person hypothesis is recursively calculated by collecting and combining scores associated to AAPMs into scores associated to APMs from bottom to upper levels. In details, the score $f_{i,s_i}(h_i, I)$ for an APM with index i and type s_i, is obtained by aggregating: $i)$ its own appearance score $f^A_{i,s_i}(h, I)$; $ii)$ the scores from each child APM $f_{c,s_c}(h_c, I)$; $iii)$ the deformation score $f^D(h_c, \hat{h}_c)$ calculated with respect to its child APM as defined in Eq. 2.

This process of estimating the score $f_{i,s_i}(h_i, I)$ by aggregating the scores from its child APMs is achieved by performing the following three steps: i) Child Location Selection step. Given an expected child part hypothesis $\hat{h}_c$ with index c and part type s_c, we select among all the location hypotheses h_c for this part the one associated to the largest score. The score associated to part c of type s_c is then: $f_{c,s_c}(\hat{h}_c, I) = \max_{h_c}\left(f_{c,s_c}(h_c, I) + f^D(h_c, \hat{h}_c)\right)$.

ii) Child Alignment step: we need to align score contributed from each part child. Let us indicate by s_c the type of c^{th} child part. Then, the expected location of the child part c is given by $T(h_i, t_{i,c}^{s_i,s_c})$, such that $T(h,t) = h - t = (x - t_x, y - t_y, l - t_l, \theta - t_\theta)$, where $t_{i,c}^{s_i,s_c}$ is the expected displacement between type s_i of part i and type s_c of part c. iii) Child Type Selection step: For each child part, we need to select the part type corresponding to the highest score as follows:

$$f_c(h_i, I) = \max_{s_c \in \mathcal{S}^c}\left(f_{c,s_c}(T(h_i, t_{i,c}^{s_i,s_c}), I) + b_{i,c}^{s_i,s_c}\right) \qquad (3)$$

where S^c is the set of types for part c, $b_{i,c}^{s_i,s_c}$ is the bias between type s_i of i^{th} part and type s_c of c^{th} part. Such biases capture the property that some types may be more descriptive than other and therefore they can affect the relevant score function differently. We learn such biases during the learning procedure (Sec. 4).

Finally, the score $f_{i,s_i}(h_i, I)$ is obtained as $f_{i,s_i}(h_i, I) = f_{i,s_i}^A(h_i, I) + \sum_{c \in C^i} f_c(h_i; I)$, where C^i is the set of child APMs. Notice that the score $f_{i,s_i}(h_i, I)$ for an atomic APM (AAPM) is simply given by its own appearance score $f_{i,s_i}^A(h_i, I)$. These are computed first as they are the primary elements of the overall object APM structure. Using this way of aggregating the scores, the matching scores for all the parts in the APM structure can be calculated once the scores of its child APMs are computed. Notice that the time required to compute the scores is linearly related to the total number of part-types in the APM.

3.3. Model Properties (APM)

In the following, we discuss the important properties of our APM: **i) Sublinearity.** As illustrated in Fig. 3, a complex APM is constructed by reusing all APMs at finer levels. If an APM contains M parts and each part contains N types, such APM can represent N^M unique combination of part-types (poses) with the cost of storing $N \times M$ appearance and deformation parameters, respectively (i.e., in Eq. 4, A, d are indexed by part i and type s_i). As a result, the number of parameters in APM grows sublinearly with respect to the number of distinct poses; **ii) Efficient Exact Inference.** Despite the complex structure of APM, the "bottom-up" process is efficient, since the scores of different part-types are reused by parent APMs at higher levels. Once the matching scores are assigned, the "top-down" process is efficient as the search for the best part configuration can be done in linear time. Compared to most of the other grammar models which only find the best configuration among a smaller subset of the full configuration space, our method can efficiently explore the full configuration space (e.g., inference on a 640 × 480 image across ~ 30 scales and 24 orientation in about 2 minutes) making exact inference tractable.

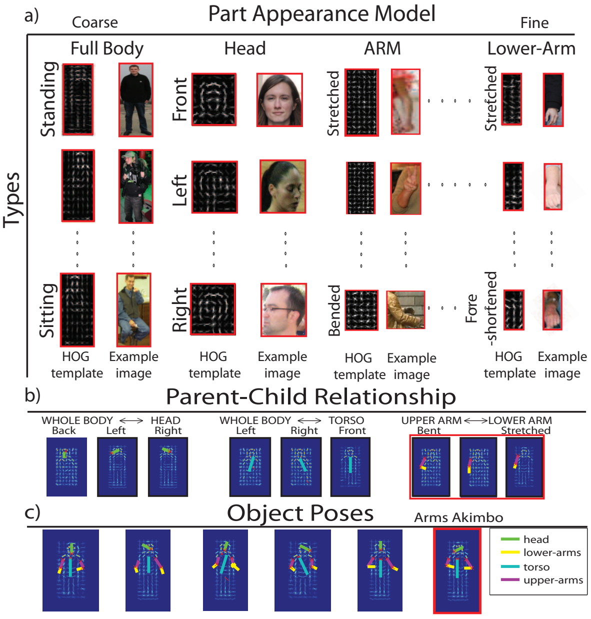


Figure 4. Visualization of a learned APM. Panel (a) shows the learned Histogram of Oriented-Gradient (HOG) templates with the corresponding example images for each part-type. Panel (b) shows the parent-child geometric relationships in our model, where different parts are represented as color coded sticks. Panel (c) shows samples of object poses obtained by selecting different combinations of part-types from the APM.

4. Model Learning

The overall model parameter $w = (A, \ldots, d, \ldots, b \ldots)$ is the collection of appearance parameters As, deformation parameters ds, and biases bs. In this section, we illustrate how to learn the model parameters w. Since all the model parameters are linearly related to the matching score (Eq. 1, 2, 3), the score of a specific set of part hypotheses H can be computed as $w^T\Psi(H; I)$, where $\Psi(H; I)$ contains all the appearance features $\psi_a(.)$, geometric features $\psi_d(.)$. The matching score can be decomposed into

$$w^T\Psi(H; I) = \sum_{i \in \mathcal{V}} A_{(i,s_i)}^T \psi_a(h_i; I) + \sum_{(i,j) \in \varepsilon}\left(b_{i,j}^{(s_i,s_j)} - d_{(j,s_j)}^T \psi_d(h_j, T(h_i, t_{ij}^{(s_i,s_j)}))\right) \quad (4)$$

where $\mathcal{V}$ is the set of part indices, ε is the set of parent-child parts, $A_{(i,s_i)}$ specify the appearance parameter for type s_i of part i, $d_{(i,s_i)}$ specify the deformation parameter for type s_i of part i, and $b_{i,j}^{(s_i,s_j)}$ and $t_{ij}^{(s_i,s_j)}$ specify bias and expected displacement of selecting part j with type s_j as the child of part i with type s_i.

Consider that we are given a set of example images and part annotations $\{I^n, H^n\}_{n=1,2,\ldots,N}$. We can cast the parameter learning problem into the following SSVM [18] problem,

$$
\begin{aligned}
\min_{w,\,\xi^n \geq 0} \quad & w^T w + C \sum_n \xi^n(H) \\
\text{s.t.} \quad & \xi^n(H) = \max_H(\triangle(H; H^n) + \\
& w^T \Psi(H; I^n) - w^T \Psi(H^n; I^n)) \\
& , \forall n \,, \forall H \in \mathcal{H} \qquad (5)
\end{aligned}
$$

where $\triangle(H; H^n)$ is a loss function measuring incorrectness of the estimated part configuration H, while the true part configuration is H^n, and C controls the relative weight of the sum of the violation term with respect to the regularization term. The loss is defined to improve the pose estimation accuracy as follows,

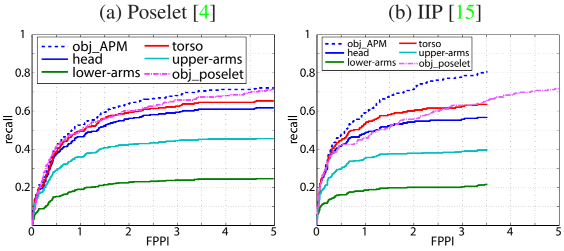
$$
\begin{aligned}
\triangle(H; H^n) &= \frac{1}{M} \sum_{i=1}^{M} \triangle((h_m, s_m); (h_m^n, s_m^n)) \\
&= \frac{1}{M} \sum_{i=1}^{M} (1 - overlap((h_m, s_m); (h_m^n, s_m^n))) \qquad (6)
\end{aligned}
$$

where $overlap((h_m, s_m); (h_m^n, s_m^n))$ is the intersection area divided by union area of two windows specified by the part locations and types. Here we use a stochastic subgradient descent method within the SSVM framework to solve Eq. 5. The subgradient of $\partial_w \xi^n(H)$ can be calculated as $\Psi(H^*; I^n) - \Psi(H^n; I^n)$, where $H^* = \arg\max_H(\triangle(H; H^n) + w^T\Psi(H; I^n))$. Since the loss function can be decomposed into a sum over local losses for each individual part i, H^* can be solved similarly to the recognition problem in Sec. 3.1.

Analysis of our learned model. Fig. 4(a) shows learned part appearance models from a person APM with 3 levels of recursion with typical part-type examples. Since all the part-type appearance models are jointly trained by minimizing the same objective function (Eq. 5), the appearance model captures the shapes of the part-type examples as well as the strength of the HOG weights reflecting the importance of each part-type (See Fig. 4 for learned HOG templates). Fig. 4(b) illustrates a few parent-child geometric relationships in the APM. For example, our model learns that a head appears on the upper-body of a person with different orientations (Fig. 4(b)-Left), and learn the stretched and bent configurations for the left-arm (Fig. 4(b)-Middle). Notice that these parent-child geometric relationships indeed capture common gestures that appear in daily person activities, like ”arms akimbo” (Fig. 4(c) red box). Fig. 4(c) shows more object poses by selecting different combinations of part-types.

5. Implementation Details

Feature representation: We use the projected Histogram of Oriented-Gradient (HOG) feature implemented in [8] to describe part-type appearance. **Manual supervision:** In order to train an APM, a set of articulated part annotations is required. For people, we use the 19 keypoints provided in the poselet dataset [4] as the part supervision. We manually annotated cats and dogs with 24 and 26 keypoints, respectively. **Type discovery:** We use the keypoints configuration and part length to object height ratio to initially group parts into different types. After this initial grouping, each example can be discriminatively assigned into different groups according to the appearance similarity. **Discretized part orientation:** We follow the common convention to divide the part orientation space into 24 discrete values (15° each).

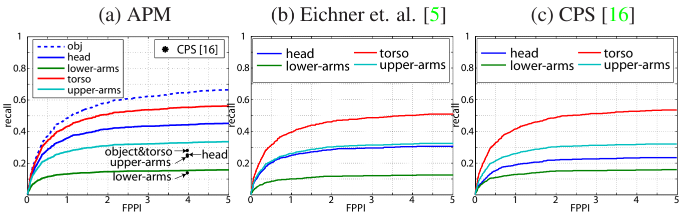

(a) Poselet [4]

(b) IIP [15]

Figure 5. Panel (a) shows that our detector applied on Poselet dataset [4] slightly outperforms the state-of-the-art person detector [3] (dashed curves). Panel (b) shows that APM significantly outperforms [3] on challenging Iterative Image Parsing dataset [15]. Recall-vs-FPPI curves are shown for each human part (with different color codes) by using our method (solid curves).

6. Experiments

We evaluate our method on three main datasets, all of which contain objects in a variety of poses in cluttered scenes. Object detection datasets that contain objects with very restricted poses (e.g., TUD-UprightPeople, TUD-Pedestrians [1]) are not suitable for evaluation here, since we are interested in datasets that make the detection and pose estimation equally challenging. First, we compare our object detection performance on the poselet [4] and Iterative Image Parsing [15] datasets with the state-of-the-art person detector [3] and demonstrate superior performance, especially on [15] which contains challenging sport images with unknown object scale. We introduce a new evaluation metric called recall-vs-False Positive Per Image (FPPI) to show joint object detection and pose estimation performance. More detail about the recall-vs-FPPI can be found in the technical report [17]. Second, on the ETHZ stickmen dataset [5], we show APM outperforms state-of-the-art pose estimators [16, 5] using detection results provided by APM. In order to prove that our method can be used to detect articulated objects other than humans, we test our method on


(a) APM

(b) Eichner et. al. [5]

(c) CPS [16]

Figure 6. Joint object detection and pose estimation performance comparison between our method (a) and [5, 16] (b,c) using recall vs. FPPI for 4 upper-body parts on stickmen dataset. ”obj” indicates the detection performance of our object detector.

PASCAL stickmen		Recall/PCP$_{0.5}$@ 4 FPPI				Recall
Det.	Pose methods	torso	head	upper arm	lower arm	obj
APM	Eichner et al[5]	0.497/77.44	0.311/48.43	0.318/49.52	0.122/19.06	0.642
	CPS[16]	0.525/81.80	0.231/35.92	0.316/49.27	0.155/24.15	
APM (ours)		0.550/85.57	0.439/68.33	0.326/50.73	0.151/23.54	

Figure 7. Comparison with other methods for recall/$PCP_{0.5}$ @ 4 FPPI. Red figures indicate the highest recall for each part. We perform better than the state-of-the-art in term of recalls for every part except lower arms.

the PASCAL 2007 cat and dog categories [6], and obtain convincing joint object detection and pose estimation performance on these extremely difficult categories.

6.1. Comparing with Poselet [3]

The Poselet dataset [4] contains people annotated with 19 types of keypoints, which include joints, eyes, nose, etc. We use the keypoints to define 6 body parts at 3 levels: at the coarsest level, the whole body has 6 types; at the middle level, head has 4 types, torso has 4 types, left&right-arms both has 7 types; at the finest level, left&right-lower-arms both has 2 types. By assuming that body parts and object bounding boxes annotations are available, we train our APM on the same positive images used in [4] and negative images from PASCAL'07 [6]. Fig. 5(a,b) shows that our object detection performance is slightly better than [3] (which achieves the best performance on PASCAL 2010 - human category) on poselet dataset [4] but significantly outperforms [3] on [15], respectively. We observed that [3] tends to fail when the aspect ratios of the object bounding boxes vary due to severe articulated parts deformations. Fig. 5(a,b) also show our joint object detection and pose estimation performance using part recall vs FPPI curves on these challenging datasets. Typical examples are shown in the 1 $\sim$ 2 rows of Fig. 9.

6.2. ETHZ Stickmen dataset

The original ETHZ stickmen dataset [5] contains 549 images, and it is partially annotated with 6 upper-body parts for each person. In order to evaluate the joint object detection and pose estimation performance, we complete the annotation for all 1283 people. Previous algorithms evaluated on this dataset are just pose estimators, which rely on an upper body detector to first localize the person. Because of this, the PCP performance is only evaluated on the 360 detected people that were found by the upper body detector (see [17] for more details). In order to obtain a fair comparison of the joint object detection and pose estimation performance, we use recall/$PCP_{0.5}$ (same as [5]) vs. FPPI curves for all parts. We believe this is a better performance measure than PCP at a specific FPPI. Indeed PCP ignores to what degree the pose estimation performance is affected by the accuracy of object detectors. Notice that PCP at different FPPI can be easily calculated from the part recall v.s. FPPI curves by dividing the recall of each part by the recall of the object. As an example, the latest PCP from [16] is equivalent to the sample points (indicated by dots) at 4 FPPI shown in Fig. 6(a). Notice that our method significantly outperforms [16] for each body part (except for lower arm where [16] and ours are on par).

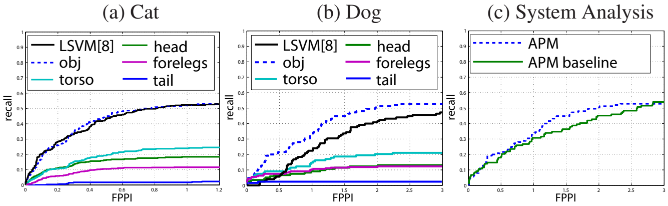


Figure 8. Joint object detection and pose estimation performance shown in recall (following the $PCP_{0.7}$ criteria as defined in [5]) vs. FPPI for cats (a) and dogs (b) on the PASCAL VOC 2007 dataset. Both performances are compared with [8]. Panel (c) compare our dog-APM with a Baseline-APM with no finer parts.

We apply our APM learned from the Poselet dataset [4] to jointly detect objects and estimate their poses on the stickmen dataset (Fig. 6(a)). For a more fair comparison, since APM detects 846 people which is much more than the 360 people detected by the upper body detector [5], we show the performance of [16, 5] by using APM's detection results (Fig. 6(b)(c)) Even though [16, 5] incorporate additional segmentation information and color cues, our method shows superior performance for almost all parts. We believe that the main reason is because that [16, 5] assume accurate person bounding boxes are given both in training and testing. Our method overcomes such limitation by performing joint object detection and pose estimation. A recall/$PCP_{0.5}$@4FPPI table comparison is also shown in Fig. 7 with the winning scores highlighted in red. We also found that our detector detects 92.5% of the 360 people detected by the upper-body detector. Among them, without knowing the object location and scale, our PCPs for torso, head, upper-arm, and lower-arm are $91.9\%, 73.0\%, 60.7\%$, and 31.1%, respectively. Typical examples are shown in the 3 $\sim$ 5 rows of Fig. 9.

6.3. PASCAL 2007 cat and dog

From the PASCAL 2007 dataset, 548 images of cats were annotated with 24 keypoints and 200 images of dogs were annotated with 26 keypoints including ears, joints, tail, etc. Similar to the training procedure of the person model, we train 5 parts at 2 levels[1] APMs for cats and dogs independently on a subset of the data and evaluated on the remaining subset. Fig. 8 shows that APM outperforms the state-of-the art object (LSVM) detector [8] trained on the same set of training data using the voc-release4 code[2]. We further conduct a system analysis on the dog dataset (Fig. 8(c)). By adding articulated parts, the performance increases compared to a baseline model with only a whole object part. Typical examples are shown in the last 2 rows of Fig. 9.

7. Conclusion

We propose the Articulated Part Model (APM) which is a recursive coarse-to-fine and multiple part-type representation for joint object detection and pose estimation of artic-

[1] Whole body at the coarsest level. Head, torso, left-foreleg, right-foreleg, and tail at the finest level.

[2] The code trains a model with 6 root components and 8 latent parts per components.

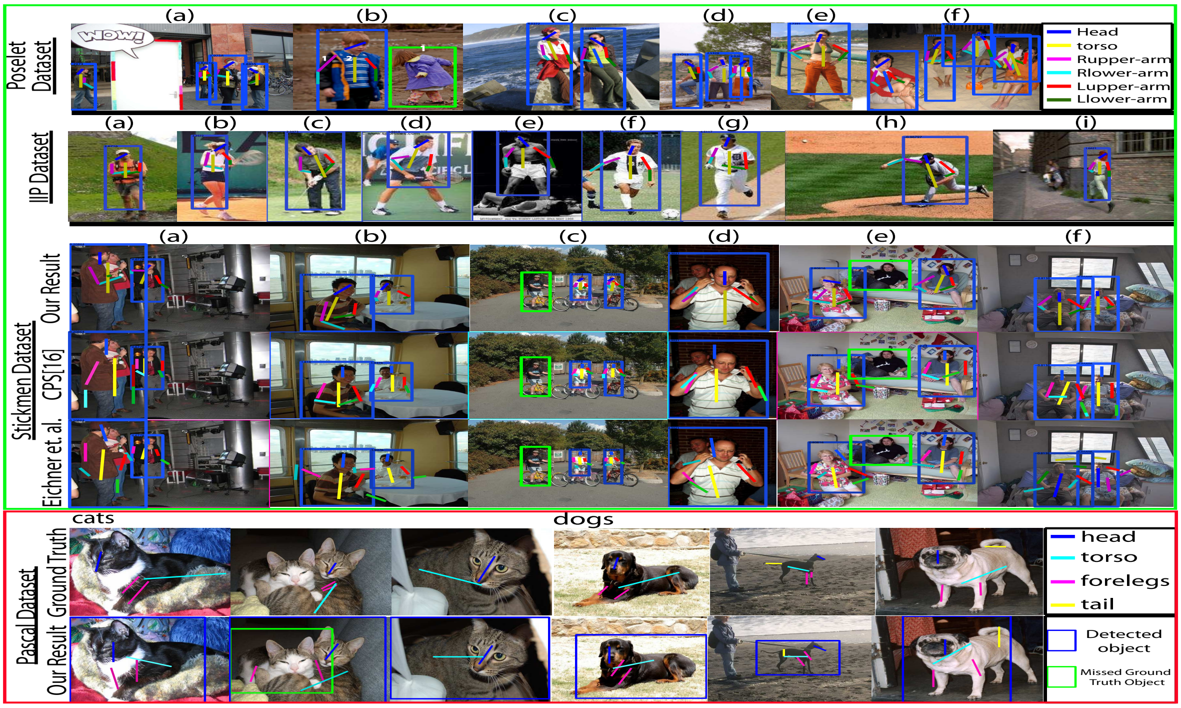


Figure 9. Typical examples of object detection and pose estimation. Sticks with different colors indicate different parts for different object categories. Blue bounding boxes are our prediction and green ones indicate missed ground truth objects. The first 2 rows show the results on Poselet dataset [4] and Iterative Image Parsing dataset [15]. Rows 3 $\sim$ 5 show the comparison between our method and [5, 16] on the stickmen dataset [5]. The last two rows show the ground truth and our results on PASCAL'07 cats and dogs [6], respectively.

ulated objects. We demonstrate on four publicly available datasets that our method obtains superior object detection performances. Using a novel performance measure (the part recall vs. FPPI curve) we show that our part recall at all FPPI are better than the state-of-the-art methods for almost all parts.

Acknowledgments. We acknowledge the support of the ONR grant $N000141110389$. We also thank Murali Telaprolu for his help and support.

References

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

[2] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *CVPR*, 2005. 1

[3] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010. 1, 6, 7

[4] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 1, 6, 7, 8

[5] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *BMVC*, 2009. 1, 6, 7, 8

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL VOC2007 Results. 7, 8

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL VOC2010 Results. 1

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010. 1, 2, 6, 7

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, 2004. 4

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 2005. 1, 2

[11] C. Ionescu, L. Bo, and C. Sminchisescu. Structural svm for visual localization and continuous state estimation. In *CVPR*, 2009. 1

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

[13] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV workshop on statistical learning in computer vision*, 2004. 1

[14] D. Marr. Vision: Acomputational investigation into the human representation and processing of visual information. *W. H. Freedman*, 1982. 1, 2

[15] D. Ramanan. Learning to parse images of articulated bodies. In *NIPS*, 2006. 1, 6, 7, 8

[16] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *ECCV*, 2010. 1, 6, 7, 8

[17] M. Sun. Technical report of articulated part-based model. http://www.eecs.umich.edu/~sunmin/. 6, 7

[18] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004. 2, 5

[19] P. Viola and M. Jones. Robust real-time object detection. *IJCV*, 2002. 1

[20] Y. Wang and G. Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *ECCV*, 2008. 1

[21] Y. Wang, D. Tran, and Z. Liao. Learning hierarchical poselets for human parsing. 2011. 2, 3

[22] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. 2011. 2, 3

[23] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010. 2, 3

[24] L. L. Zhu, Y. Chen, Y. Lu, C. Lin, and A. Yuille. Max margin and/or graph learning for parsing the human body. In *CVPR*, 2008. 2, 3

[25] L. L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010. 2

[26] S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, 2(4), 2006. 2, 3