Part-based models for finding people and estimating their pose

Deva Ramanan

Abstract This chapter will survey approaches to person detection and pose estimation with the use of part-based models. After a brief introduction/motivation for the need for parts, the bulk of the chapter will be split into three core sections on Representation, Inference, and Learning. We begin by describing various gradient-based and color descriptors for parts. We will next focus on Representations for encoding structural relations between parts, describing extensions of classic pictorial structures models to capture occlusion and appearance relations. We will use the formalism of probabilistic models to unify such representations and introduce the issues of inference and learning. We describe various efficient algorithms designed for tree-structures, as well as focusing on discriminative formalisms for learning model parameters. We finally end with applications of pedestrian detection, human pose estimation, and people tracking.

1 Introduction

Part models date back to the generalized cylinder models of Binford [3] and Marr and Nishihara [40] and the pictorial structures of Fischler and Elschlager [24] and Felzenszwalb and Huttenlocher [19]. The basic premise is that objects can be modeled as a collection of local templates that deform and articulate with respect to one another.

Contemporary work: Part-based models have appeared in recent history under various formalisms. Felzenszwalb and Huttenlocher [19] directly use the pictorial structure moniker, but also notably develop efficient inference algorithms for matching them to images. Constellation models [20, 7, 63] take the same approach, but use a sparse set of parts defined at keypoint locations. Body plans [25] are another rep-

Deva Ramanan, Department of Computer Science, University of California at Irvine e-mail: dramanan@ics.uci.edu

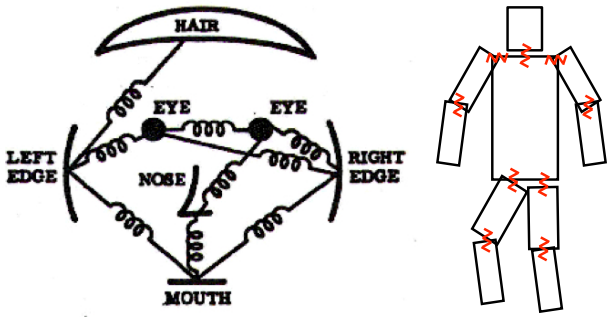

Fig. 1 One the **left**, we show a pictorial structure model [24, 19] which models objects using a collection of local part templates together with geometric constraints, often visualized as springs. One the **right**, we show a pictorial structure for capturing an articulated human "puppet" of rectangular limbs, where springs have been drawn in red for clarity.

resentation that encodes particular geometric rules for defining valid deformations of local templates.

Star models: A particularly common form of geometric constraint is known as a "star model", which states that part placements are independent within some root coordinate frame. Visually speaking, one think of springs connecting each part to some root bounding box. This geometric model can be implicitly encoded in an implicit shape model [38]. One advantage of the implicit encoding is that one can typically deal with a large vocabulary of parts, sometimes known as a codebook of visual words [57]. Oftentimes such codebooks are generated by clustering candidate patches typically found in images of people. Poselets [4] are recent successful extension of such a model, where part models are trained discriminatively using fully supervised data, eliminating the need for codebook generation through clustering. K-fan models generalize star models [9] by modelling part placements as independant given the location of K reference parts.

Tree models: Tree models are a generalization of star model that still allow for efficient inference techniques [19, 28, 45, 51]. Here, the independence assumptions correspond to child parts being independently placed in a coordinate system defined by their parent. One common limitation of such models is the so-called "double-counting" phenomena, where two estimated limbs cover the same image region because their positions are estimated independently. We will discuss various improvements designed to compensate for this limitation.

Related approaches: Active appearance models [8, 41] are a similar object representation that also decomposes an object into local appearance models, together with geometric constraints on their deformation. Notably, they are defined over continuous domains rather than a discretized state space, and so rely on continuous optimization algorithms for matching. Alternatively, part-based representations have also been used for video analysis be requiring similar optical flow for pixels on the same limb [32, 5].

2 Part models

In this section, we will overview techniques for building localized part models. Given an image I and a pixel location $l_i = (x_i, y_i)$, we write $\phi(I, l_i)$ for the local descriptor for part i extracted from a fixed size image patch centered at l_i. It is helpful to think of part models as fixed-size templates that will be used to generate part detections by scanning over the image and finding high-scoring patches. We will discuss linearly-parameterized models where the local score for part i is computed with a dot product $w_i \cdot \phi(I, l_i)$. This allows one to use efficient convolution routines to generate scores at all locations in an image. To generate detections at multiple scales, one can search over an image pyramid. We will discuss more detailed parameterizations that include orientation and foreshortening effects in Section 3.2.

2.1 Color models

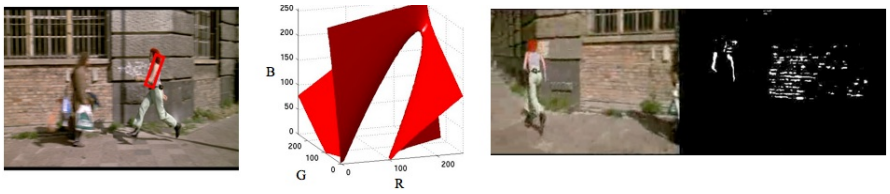

Fig. 2 One the **left**, show pixels used to train a color-based model for an arm. Pixels inside the red rectangle are treated as positive examples, while pixels outside are treated as negatives. On the **left-center**, we show the discriminant boundary learned by a classifier (specifically, logistic regression defined on quadratic RGB features). On the **right** two images, we show a test image and arm-pixel classification results using the given discriminant boundary.

The simplest part model is one directly based on pixel color. A head part should, for example, contain many skin pixels. This suggests that augmenting a a head part template with a skin detector will be beneficial. In general, such color-based models will not work well for limbs because of intra-class variation; people can appear in a variety of clothes with various colors and textures. Indeed, this is one of the reasons why human pose estimation and detection is challenging. In some scenarios, one may know the appearance of clothing *a priori*; for example, consider processing sports footage with known team uniforms. We show in Section 4.2 and Section 6.3 that one can learn such color models automatically from a single image or a video sequence. Color models can be encoded non-parametrically with a histogram (e.g., 8 bins per RGB axis resulting in a $8^3 = 512$ descriptor), or a parametric model which is typically either a gaussian or a mixture of gaussians. In the case of a simple gaussian, the corresponding color descriptor $\phi_{RGB}(I, l_i)$ encodes standard sufficient

statistics computed over a local patch; the mean ($\mu \in R^3$) and covariance ($\Sigma \in R^{3\times 3}$) of the color distribution.

2.2 Oriented gradient descriptors

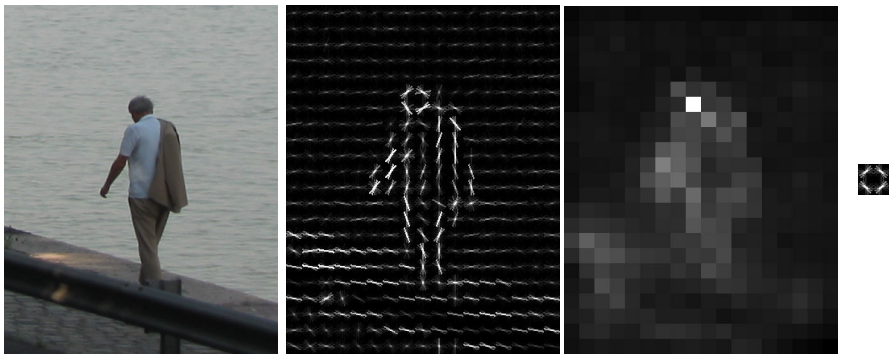

Fig. 3 On the **left**, we show an image. On the **center left**, we show its representation under a HOG descriptor [10]. A common visualization technique is to render an oriented edge with intensity equal to its histogram count, where the histogram is computed over a 8×8 pixel neighborhood. We can use the same technique to visualize linearly-parameterized part models; we show a "head" part model on the **right**, and its associated response map for all candidate head location on the **center right**. We see a high response for the true head location. Such invariant representations are useful for defining part models when part colors are not known *a priori* or not discriminative.

Most recognition approaches do not work directly with pixel data, but rather some feature representation designed to be more invariant to small changes in illumination, viewpoint, local deformation, etc. One of the most successful recent developments in object recognition is the development of engineered, invariant descriptors, such as the scale-invariant feature transform (SIFT) [39] and the histogram of oriented gradient (HOG) descriptor [10]. The basic approach is to work with normalized gradient orientation histograms rather than pixel values. We will go over HOG, as that is a particular common representation. Image gradients are computed at each pixel by finite differencing. Gradients are then binned into one of (typically) 9 orientations over local neighborhoods of 8×8 pixel. A particularly simple implementation of this is obtained by computing histograms over non-overlapping neighborhoods. Finally, these orientation histograms are normalized by aggregating orientation statistics from a local window of 16×16 pixels. Notably, in the original definition of [10], each orientation histogram is normalized with respect to multiple (4, to be exact) local windows, resulting in vector of 36 numbers to encoding the local orientation statistics of a 8×8 neighborhood "cell". Felzenszwalb et al [18] demonstrate that one can reduce the dimensionality of this descriptor to 13 num-

bers by looking at marginal statistics. The final histogram descriptor for a patch of $n_x \times n_y$ neighborhood cells is $\phi(I,l_i) \in R^{13n_xn_y}$.

3 Structural constraints

In this section, we describe approaches for composing the part models defined in the previous section into full body models.

3.1 Linearly-parameterized spring models

Assume we have a K-part model, and let us write the location of the k^{th} part as l_k. Let us write $z = \{l_1,\ldots,l_K\}$ for a particular *configuration* of all K parts. Given an image I, we wish to score each possible configuration z:

$$S(I,z) = \sum_{i=1}^{K} w_i \cdot \phi(I,l_i) + \sum_{i,j\in E} w_{ij} \cdot \psi(I,l_i,l_j) \tag{1}$$

We would like to maximize the above equation over z, so that for a given image, our model can report the best-scoring configuration of parts.

Appearance term: We write $\phi(I,l_i)$ for the image descriptor extracted from location l_i in image x, and w_i for the HOG filter for part i. This local score is akin to the linear template classifier described in the previous section.

Deformation term: Writing $dx = x_j - x_i$ and $dy = y_j - y_i$, we can now define:

$$\psi(I,l_i,l_j) = \left[dx\ dx^2\ dy\ dy^2\right]^T \tag{2}$$

which can be interpreted as the negative spring energy associated with pulling part j from a canonical relative location with respect to part i. The parameters w_{ij} specify the rest location of the spring and its rigidity; some parts may be easier to shift horiontally versus veritically. In Section 3.3, we derive these linear parameters from a Gaussian assumption on relative location, where the rest position of the spring is the mean of the Gaussian, and rigidity is specified by the covariance of the Gaussian.

We define E to be the (undirected) edge set for a K-vertex relational graph $G = (V,E)$ that denotes which parts are constrained to have particular relative locations. Intuitively, one can think of G as the graph obtained from Figure 1 by replacing parts with vertices and springs with edges. Felzenszwalb and Huttenlocher [19] show that this deformation model admits particularly efficient inference algorithms when G is a tree (as is the case for the body model in the right of Figure 1).

For greater flexibility, one could also make the deformation term depend on the image I. For example, one might desire consistency in appearance between left and

right body parts, and so one could augment $\psi(I, l_i, l_j)$ with squared difference between color histograms extracted at locations l_i and l_j [61]. Finally, we note that the score can be written function of the part appearance and spatial parameters:

$$S(I,z) = w \cdot \Phi(I,z)$$

3.2 Articulation

The classic approach to modeling articulated parts is to augment part location l_i with pixel position, orientation, and foreshortening

$$l_i = (x_i, y_i, \theta_i, s_i).$$

This requires augmenting the spatial relational model (2) with model relative orientation and relative foreshortening, as well as relative location. Notably, this enhanced parameterization increases the computational burden of scoring the local model, since one must convolve an image with a family of rotated and foreshortened part templates.

While [19] advocate explicitly modeling foreshortening, recent work[49, 45, 48, 1] appear to obtain good results without it, relying on the ability of the local detectors to be invariant to small changes in foreshortening. [48] also demonstrate that by formulating the above scoring function in probabilistic terms and extracting the *uncertainty* in estimates of body pose (done by computing marginals), one can estimate foreshortening. In general, parts may also differ in appearance due to other factors such as out-of-plane rotations (e.g., frontal versus profile faces) and semantic part states (e.g., an open versus a closed hand).

In recent work, [64] foregoe an explicit modeling of articulation, and instead model oriented limbs with mixtures of non-articulated part models - see Figure 10. This has the computational advantage of sharing computation between articulations (typically resulting in orders of magnitude speedups), while allowing mixture models to capture other appearance phenomena such as out-of-plane orientation, semantic part states, etc.

3.3 Gaussian tree models

In this section, we will develop a probabilistic graphical model over part locations and image features. We will show that the log posterior of part locations given image features can be written in the form of (1). This provides an explicit probabilistic motivation for our scoring function, and also allows for the direct application of various probabilistic inference algorithms (such as sampling or belief propagation). We will also make the simplifying assumption that the relational graph $G = (V, E)$

is a tree that is (without loss of generality) rooted at part/vertex $i = 1$. This means we can model G as a directed graph, further simplyifying our exposition.

Spatial prior: Let us first define a prior over a configuration of parts z. We assume this prior factors into a product of local terms

$$P(z) = P(l_1) \prod_{ij \in E} P(l_j|l_i) \tag{3}$$

The first term is a prior over locations of the root part, which is typically the torso. To maintain a translation invariant model, we will set $P(z_1)$ is to be uninformative. The next terms specify spatial priors over the location of a part given its parent in the directed graph G. We model them as diagonal-covariance Gaussian density defined the relative location of part i and j:

$$P(z_j|z_i) = N(z_j - z_i; \mu_j, \Sigma_j) \quad \text{where} \quad \Sigma_j = \begin{bmatrix} \sigma_{j,x} & 0 \\ 0 & \sigma_{j,y} \end{bmatrix} \tag{4}$$

The ideal rest position of part j with respect to its parent is given by μ_j. If part j is more likely to deform horizontally rather an vertically, one would expect $\sigma_{j,x} > \sigma_{j,y}$.

Feature likelihood: We would like a probabilistic model that explains all features observed at all locations in an image, including those generated by parts and those generated by a background model. We write L for the set of all possible locations in an image. We denote the full set of observed features as

$$\{\phi(I, l') | l' \in L\}$$

If we imagine a pre-processing step that first finds a set of candidate part detections (e.g., candidate torsos, heads, etc.), we can intuitively think of L as the set of locations associated with all candidates. Image features at a subset of locations $l_i \in L$ are generated from an appearance model for part i, while all other locations from L (not in z) generate features from a background model:

$$\begin{aligned} P(I|z) &= \prod_i P_i(\phi(I, l_i)) \prod_{l' \in L \setminus z} P_{bg}(\phi(I, l')) \\ &= Z \prod_i r(\phi(I, l_i)) \end{aligned} \tag{5}$$

$$\text{where} \quad r(\phi(I, l_i)) = \frac{P_i(\phi(I, l_i))}{P_{bg}(\phi(I, l_i))} \quad \text{and} \quad Z = \prod_{l' \in L} P_{bg}(\phi(I, l'))$$

We write $P_i(\phi(I, l_i))$ for the likelihood of observing feature $\phi(I, l_i)$ given an appearance model for part i. We write $P_{bg}(\phi(I, l')$ for the likelihood of observing feature $\phi(I, l')$ given a background appearance model. The overall likelihood is, up to a constant, only dependent on features observed at part locations. Specifically, it depends on the *likelihood ratio* of observing the features given a part model versus a background model. Let us assume the image feature likelihood in (5) are Gaussian densities with a part or background-specific mean α and a single covariance Σ:

$$P_i(\phi(I,l_i)) = N(\phi(I,l_i);\alpha_i,\Sigma) \quad \text{and} \quad P_{bg}(\phi(I,l_i)) = N(\phi(I,l_i);\alpha_{bg},\Sigma) \tag{6}$$

Log linear posterior: The relevant quantity for inference, the posterior, can now be written as a log-linear model:

$$P(z|I) \propto P(I|z)P(z) \tag{7}$$

$$\propto \exp^{w\cdot\Phi(I,z)} \tag{8}$$

where w and $\Phi(I,z)$ are equivalent to their definitions in Section 3.1. Specifically, one can map Gaussian mean and variances to linear parameters as below, providing a probabilistic motivation for the scoring function from (1).

$$w_i = \Sigma^{-1}(\alpha_i - \alpha_{bg}), \qquad w_{ij} = -\left[\frac{\mu_{j,x}}{\sigma^2_{j,x}} \ \frac{1}{2\sigma^2_{j,x}} \ \frac{\mu_{j,y}}{\sigma^2_{j,y}} \ \frac{1}{2\sigma^2_{j,y}}\right]^T \tag{9}$$

Note that one can relax the diagonal covariance assumption in (4) and part-independant covariance assumption in (6) and still obtain a log-linear posterior, but this requires augmenting $\Phi(I,z)$ to include quadratic terms.

3.4 Inference

Fig. 4 Felzenszwalb and Huttenlocher [19] describe efficient dynamic programming algorithms for computing the MAP body configuration, as well as efficient algorithms for sampling from the posterior over body configurations. Given the image and foreground silhoette (used to construct part models) on the **left**, we show two sampled body configurations on the **right** two images.

MAP estimation: Inference corresponds to maximizing $S(x,z)$ from (1) over z. When the relational graph $G = (V,E)$ is a tree, this can be done efficiently with dynamic programming (DP). Let kids(j) be the set of children of j in E. We compute the message part j passes to its parent i by the following:

$$\text{score}_j(z_j) = w_j \cdot \phi(x,z_j) + \sum_{k\in\text{kids}(j)} m_k(z_j) \tag{10}$$

$$m_j(z_i) = \max_{z_j} \ \text{score}_j(z_j) + w_{ij} \cdot \psi(x,z_i,z_j) \tag{11}$$

Eq. (10) computes the local score of part j, at all pixel locations z_j, by collecting messages from the children of j. Eq. (11) computes for every location of part i, the best scoring location of its child part j. Once messages are passed to the root part ($j = 1$), $\text{score}_1(z_1)$ represents the best scoring configuration for each root position. One can use these root scores to generate multiple detections in image x by thresholding them and applying non-maximum suppression (NMS). By keeping track of the argmax indices, one can backtrack to find the location and type of each part in each maximal configuration.

Computation: The computationally taxing portion of DP is (11). Assume that there are $|L|$ possible discrete pixel locations in an image. One has to loop over $|L|$ possible parent locations, and compute a max over $|L|$ possible child locations and types, making the computation $O(|L|^2)$ for each part. When $\phi(p_i - p_j)$ is a quadratic function and L is a set of locations on a pixel grid (as is the case for us), the inner maximization in (11) can be efficiently computed for each combination of t_i and t_j in $O(|L|)$ with a max-convolution or distance transform [19]. Message passing reduces to $O(|L|)$ per part, making the overall maximization $O(|L|K)$ for a K-part model.

Sampling: Felzenszwalb and Huttenlocher [19] also point out that tree models allow for efficient sampling. As opposed to traditional approaches to sampling, such as Gibbs sampling or Markov Chain Monte Carlo (MCMC) methods, sampling from a tree-structured model requires *zero burn-in time*. This is because one can directly compute the root marginal $P(l_1|I)$ and pairwise conditional marginals $P(l_j|l_i,I)$ for all edges $ij \in E$ with the sum-product algorithm (analogous to the forward-backward algorithm for inference on discrete Hidden Markov Models). The forward pass corresponds to "upstream" messages, passed from part j to its parent i:

$$P(l_j|l_i,I) \propto P(l_j|l_i)a_j(l_j) \tag{12}$$

$$a_j(l_j) \propto \exp^{w_j \cdot \phi(I,l_j)} \prod_{k \in \text{kids(j)}} \sum_{l_k} P(l_k|l_j,I) \tag{13}$$

When part location l_i is parameterized by an (x,y) pixel position, one can represent the above terms as 2D images. The image a_j is obtained by multiplying together response images from the children of part j and from the local template w_j. When $P(l_j|l_i) = f(l_j - l_i)$, the summation in (13) can be computed by convolving image a_k with filter f. When using a Gaussian spatial model (4), the filter is a standard Gaussian smoothing filter, for which many efficient implementations exist. At the root, the image $a_1(l_1)$ is the true conditional marginal $P(l_1|x)$. Given cached tables of $P(l_1|I)$ and $P(l_j|l_i,I)$, one can efficiently generate samples by the following: Generate a sample from the root $z'_1 \sim P(l_1|I)$, and then generate a sample from the next ordered part given its sampled parent: $l'_j \sim P(I_j|l'_i,I)$. Each involves a table lookup, making the overall sampling process very fast.

Marginals: It will also be convenient to directly compute singleton and pairwise marginals $P(l_i|I)$ and $P(l_j,l_i|I)$ for parts and part-parent pairs. This can be done by first computing the upstream messages in (13), where the root marginal is given by $P(l_1|I) = a_1(l_1)$. and then computing downstream messages from part i to its child part j:

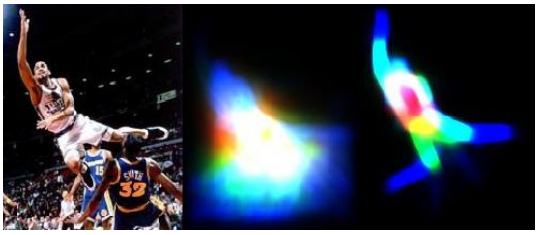

Fig. 5 One can compute part marginals using the sum-product algorithm [45]. Given part marginals, one can render a weighted rectangular mask at all image locations, where weights are given by the marginal probability. Lower limbs are rendered in blue, upper limbs and the head are rendered in green, and the torso is rendered in red. Regions of strong color correspond to pixels that likely to belong to a body part, according to the model. In the **center**, part models are defined using edge-based templates. On the **right**, part models are defined using color models.

$$\begin{aligned} P(l_j, l_i|I) &= P(l_j|l_i, I)P(l_i|I) \\ P(l_j|I) &= \sum_{l_i} P(l_j, l_i|I) \end{aligned} \tag{14}$$

4 Non-tree models

In this section, we describe constraints and associated inference algorithms for non-tree relational models.

4.1 Occlusion constraints

Tree-based models imply that left and right body limbs are localized independently given a root torso. Since left and right limb templates look similar, they may be attracted to the same image region. This often produces pose estimates whose left and right arms (or legs) overlap, or the so-called "double-counting" phenomena. Though such configurations are physically plausible, we would like to assign them a lower score than a configuration that explains more of the image. One can do this by introducing a constraint that an image region an only be claimed by a single part. There has been a body of work [58, 34, 55] developing layered occlusion models for part-based representations. Most do so by adding an additional visibility flag $v_i \in \{0,1\}$ for part i:

$$P(I|z,v) = \prod_i P_i(\phi(I,l_i))^{v_i} \prod_{l' \in L \setminus z} P_{bg}(\phi(I,l')) \tag{15}$$

$$P(v|z) \propto \prod_C \text{vis}(v_C, z_C) \tag{16}$$

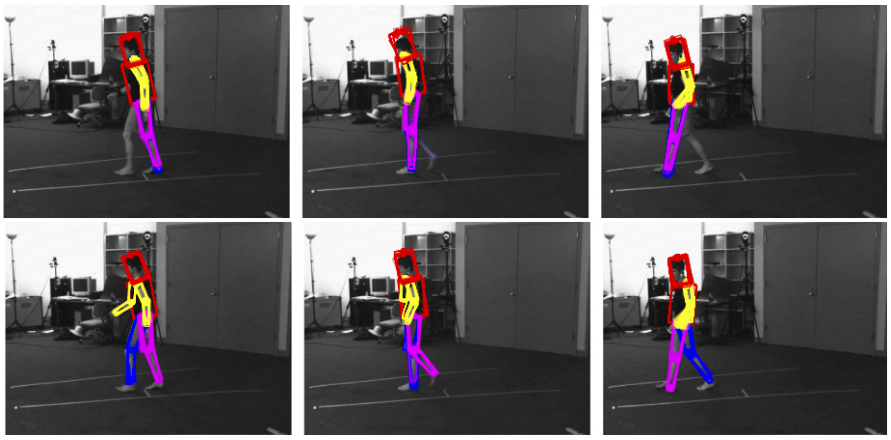

Fig. 6 Sigal and Black [55] demonstrate that the "double-counting" in tree models (**top row**) can be eliminated with an occlusion-aware likelihood model (**bottom row**).

where C is a collection of cliques of potentially overlapping parts, and vis is a binary visibility function that assigns 1 to valid configurations and visibility states (and 0 otherwise). One common approach is to only consider pairwise cliques of potentially overlapping parts (e.g., left/right limbs). Other extensions include modeling visibility at the pixel-level rather than the part-level, allowing for parts to be partially visible [55]. During inference, one may marginalize out the visibility state z and simply estimate part locations z, or one simultaneously estimate both. In either case, probabilistic dependancies between left and right limbs violate classic tree independence assumptions - e.g., left and right limbs are no longer independently localized for a fixed root torso.

4.2 Appearance constraints

People, and objects in general, tend to be consistent in appearance. For example, left and right limbs often look similar in appearance because clothes tend to be mirror symmetric [42, 46]. Upper and lower limbs often look similar in appearance, depending on the particular types of clothing worn (shorts versus pants, long-sleeves versus short sleeves) [61]. Constraints can even be long-scale, as the hands and face of a person tend to have similar skin tones. Finally, an additional cue is that of *background* consistency; consider an image of a person standing on a green field. By enforcing the constraint that body parts are not green, one can essentially subtract out the background [45, 21].

Pairwise consistency: One approach to enforcing appearance constraints is to break them down into pairwise constraints on pairs of parts. One can do this by

defining an augmented pairwise potential

$$\psi(I,l_i,l_j) = ||\phi_{RBG}(I,l_i) - \phi_{RGB}(I,l_j)||^2 \tag{17}$$

where $\phi_{RBG}(I,l_i)$ are color models extracted from a window centered at location l_i. One would need to augment the relational graph G with connections between pairs of parts with potential appearance constraints. The associated linear parameters would learn to what degree certain parts look consistent. Tran and Forsyth show such cues are useful [61]. Ideally, this consistency should depend on additional latent factors; if the person is wearing pants, that both the upper,lower,left, and right leg should look consistent in appearance. We see such encodings as a worthwhile avenue of future research. Additionally, one can augment the above potentials with additional image-specific cues. For example, the lack of a strong intervening contour between a putative upper and lower arm location may be further evidence of a correct localization. Sapp et al. explore such cues in [52, 53].

Global consistency: Some appearance constraints, such as a background model, are non-local. To capture them, we can augment the entire model with latent appearance variables a.

$$\phi(I,l_i,a) = \begin{bmatrix} \phi(I,l_i) \\ f(\phi_{RGB}(I,l_i),a_i,a_{bg}) \end{bmatrix} \tag{18}$$

where we define a_i to be appearance of part i and a_{BG} is the appearance of the background. Ramanan [45] treats these variables as latent variables that are estimated simultaneously with part locations l_i. This is done with an iterative inference algorithm whose steps are visualized in Figure 5. Ferrari et al. [21] learn such variables by applying a foreground-background segmentation engine on the output of a upright person detector.

4.3 Inference with non-tree models

As we have seen, tree models allow for a number of efficient inference procedures. But we have also argued that there are many cues that do not decompose into tree constraints. We briefly discuss a number of extensions for non-tree models. Many of them originated in the tracking literature, in which (even tree-structured) part-based models necessarily contain loops once one imposes a motion constraint on each part - e.g., an arm most not only lie near its parent torso, but must also lie near the arm position in the previous frame.

Mixtures of trees: One straightforward manner of introducing complexity into a tree model is to add a global, latent mixture model $z = \{l_1,\ldots,l_K,z_{global}\}$. For example, the latent variable could specify the viewpoint of the person; one may expect different spatial locations of parts given this latent variable. Given this latent variable, the overall model reduces to a tree. This suggests the following inference procedure:

$$\max_{z} S(I,z) = \max_{z_{global}} \max_{\{l_1,\ldots l_K\}} S(I,z) \tag{19}$$

where the inner maximization can exploit standard tree-based DP inference algorithms. Alternatively, one can compute a posterior by averaging the marginals produced by inference on each tree. Ioffe and Forsyth use such models to capture occlusion constraints [27]. Lan and Huttenlocher use mixture models to capture phases of a walking cycle [36], while Wang and Mori [62] use additive mixtures, trained discriminatively in a boosted framework, to model occlusion contraints between left/right limbs. Tian and Sclaroff point out that, if spring covariances are shared across different mixture components, one can reuse distance transform computations across mixtures [60]. Johnson and Everingham [31] demonstrate that part appearances may also depend on the mixture component (e.g., faces may appear frontally or in profile), and define a resulting mixture tree-model that is state-of-the-art

Generating tree-based configurations: One approach is to use tree-models as a mechanism for generating candidate body configurations, and scoring the configurations using more complex non-tree constraints. Such an approach is similar to N-best lists common in speech decoding. However, in our case, the N-best configurations would tend to be near-duplicates - e.g., one-pixel shifts of the best-scoring pose estimate. Felzenszwalb and Huttenlocher [19] advocate the use of sampling to generate multiple configurations. These samples can be re-scored to obtain an estimate of the posterior over the full model, an inference technique known as importance sampling. Buehler et al. [6] argues that one obtains better samples by sampling from max-marginals. One promising area of research is the use of branch-and-bound algorithms for optimal matching. Tian and Sclaroff [60] point out that one can use tree-structures to generate lower-bounds which can be used to guide search over the space of part configurations.

Loopy belief propagation: A successful strategy for dealing with "loopy" models is to apply standard tree-based belief propagation (for computing probabilistic or max-marginals) in an iterative fashion. Such a procedure is not guaranteed to converge, but often does. In such situations it can be shown to minimize a variational approximation to the original probabilistic model. One can reconstruct full joint configurations from the max-marginals, even in loopy models [65].

Continuous state-spaces: There has also been a family of techniques that directly operate on a continuous state space of l_i rather than discretizing to the pixel grid. It is difficult to define probabilistic models on continuous state spaces. Because posteriors are multi-model, simple Gaussian parameterizations will not suffice. In the tracking literature, one common approach to adaptively discretize the search space using a set of samples or particles. Particle filters have the capability to capture non-Gaussian, multi-modal distributions. Sudderth et al. [59], Isard [29], and Sigal et al. [56] develop extensions for general graphical models, demonstrating results for the task of tracking articulated models in videos. In such approaches, samples for a part are obtained by a combination of sampling from the spatial prior $P(l_j|l_i)$ and the likelihood $P_i(\phi(I,l_i))$. Techniques which focus on the latter are known as data-driven sampling techniques [37, 26].

5 Learning

The scoring functions and probabilistic models defined previously contain parameters specifying the appearance of each part w_i and parameters specifying the contextual relationships between parts w_{ij}. We would like to set these parameters so that they reflect the statistics of the visual world. To do so, we assume are given training data with images and annotated part locations $\{I_n, z_n\}$. We also assume that the edge structure E is fixed and known (e.g., as shown in Figure 1). We will describe a variety of methods for learning parameters given this data.

5.1 Generative models

The simplest method for learning is to learn parameters that maximize the joint likelihood of the data:

$$w_{ML} = \operatorname*{argmax}_w \prod_n P(I_n, z_n | w) \tag{20}$$

$$= \operatorname*{argmax}_w \prod_n \prod_i P(I_n | l_{i,n}, w_i) \prod_{ij \in E} P(l_{j,n} | l_{i,n}, w_{ij}) \tag{21}$$

Recall that the weights w are a function of Gaussian parameters $\{\mu, \sigma, \alpha, \Sigma\}$ as in (9). We can learn each parameter by standard Gaussian maximum likelihood estimation (MLE), which requires computing sample estimates of means and variances. For example, the rest position for part i is given by the average relative location of part i with respect to its parent from the labeled data. The appearance template for part i is given by computing its average appearance, computing the average appearance of the background, and taking the difference weighted by a sample covariance.

5.2 Conditional Random Fields

One of the limitations of a probabilistic generative approach is that assumptions of independence and Gaussian parameterizations (typically made to ensure tractability) are not likely to be true. Another difficulty with generative models is that they are not tied directly to a pose estimation task. While generative models allow us to sample and generate images and configuration, we want a model that produces accurate pose estimates when used for *inference*.

Discriminative models are an attempt to accomplish the latter. One approach to doing this, advocated by [49], is to estimate parameters that maximize the posterior probability $P(z_n | I_n)$ over the training set:

$$\operatorname*{argmax}_w \prod_n P(z_n | I_n, w) \tag{22}$$

This in turn can be written as

$$\begin{aligned} \underset{w}{\text{argmin}}\ L_{CRF}(w) \quad \text{where} \quad & L_{CRF}(w) = \lambda \frac{1}{2}||w||^2 + \sum_n Z_w(I_n) - w \cdot \Phi(I_n, z_n) \\ \text{and} \quad & Z_w(I_n) = \sum_{z'} exp^{w \cdot \Phi(I_n, z')} \end{aligned} \tag{23}$$

where we have taken logs to simplify the expression (while preserving the argmax) and added an optional but common regularization term (to reduce the tendency to overfit parmeters to training data). The second derivative of $L_{CRF}(w)$ is non-negative, meaning that it is a convex function whose optimum can be found with simple gradient descent: $w := w + \text{stepsize}\frac{\partial L_{CRF}(w)}{\partial w}$. Ramanan and Sminchisescu [49] point out this such a model is an instance of a conditional random feild (CRF) [35], and show that the gradient is obtained by computing expected sufficient statistics, requiring access to posterior marginals $P(z_i|x_n, w)$ and $P(z_i, z_j|x_n, w)$. This means that each iteration of gradient descent will require the two-pass "sum-product" inference algorithm (14) to compute the gradient for each training image.

5.3 Structured Max-Margin Models

One can generalize the objective function from (23) to other types of losses. Assume that in addition to training images of people with annotated poses $\{I_n, z_n\}$, we are also given a negative set of images of backgrounds. One can use this training data to to define a structured prediction objective function, similar to those proposed in [16, 33]. To do so, we note that because the scoring function $S_w(x,z)$ is linear in model parameters w, it can be written as $S(x,z) = w \cdot \Phi(x,z)$.

$$\begin{aligned} \arg\min_{w, \xi_n \geq 0} \quad & \lambda \frac{1}{2}||w||^2 + \sum_n \xi_n \\ \text{s.t.} \quad \forall n \in \text{pos} \quad & w \cdot \Phi(I_n, z_n) \geq 1 - \xi_n \\ \forall n \in \text{neg}, \forall z \quad & w \cdot \Phi(I_n, z) \leq -1 + \xi_n \end{aligned} \tag{24}$$

The above constraint states that positive examples should score better than 1 (the margin), while negative examples, for all configurations of parts, should score less than -1. The objective function penalizes violations of these constraints using slack variables ξ_n. Traditional structured prediction tasks do not require an explicit negative training set, and instead generate negative constraints from positive examples with mis-estimated labels z. This corresponds to training a model that tends to score a ground-truth pose highly and alternate poses poorly. While this translates directly to a pose estimation task, the above formulation also includes a "detection" component: it trains a model that scores highly on ground-truth poses, but generates low

scores on images without people. Recent work has shown the above to work well for *both* pose estimation and person detection [64, 33].

The above optimization is a quadratic program (QP) with an exponential number of constraints, since the space of z is $|L|^K$. Fortunately, only a small minority of the constraints will be active on typical problems (e.g., the support vectors), making them solvable in practice. This form of learning problem is known as a structural support vector machine (SVM), and there exists many well-tuned solvers such as the cutting plane solver of SVMStruct [23] and the stochastic gradient descent (SGD) solver in [18], and the dual decomposition method of [33].

5.4 Latent-variable structural models

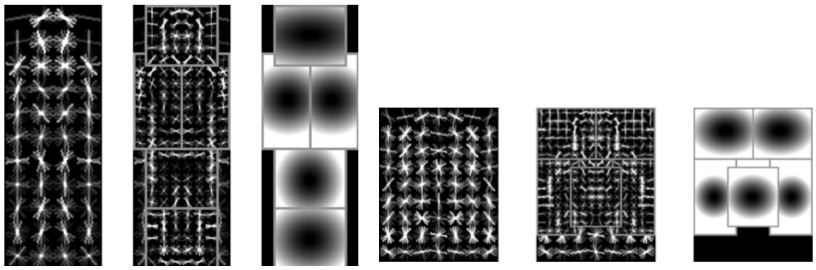

Fig. 7 We show the discriminative part models of Felzenszwalb et al. [18] trained to find people. The authors augment their latent model to include part locations and a discrete mixture component that, in this case, finds full (**left**) upper versus upper-body people (**right**). On benchmark datasets with occluded people, such as the well-known PASCAL Visual Object Challenge [15], such occlusion aware models are crucial for obtaining good performance. Notably, these models are trained using weakly-supervised benchmark training data that consists bounding boxes encompassing the entire object. The part representation is learned automatically using the coordinate descent algorithm described in Section [**?**]

In many cases, it maybe difficult to obtain "reliable" estimates of part labels. Instead, assume every positive example comes with a domain Z_n of possible latent values. For example, limb parts are often occluded by each other or the torso, making their precise location unknown. Because part models are defined in 2D rather than 3D, it is difficult for them to represent out-of-plane rotations of the body. Because of this, left/right limb assignments are defined with respect to the image, and not the coordinate system of the body (which maybe more natural when obtaining annotated data). For this reason, it also may be advantageous to encode left/right limb labels as latent.

Coordinate descent: In such cases, there is a natural algorithm to learn structured models with latent part locations. One begins with a guess for the part locations on positive examples. Given this guess, one can learn a w that minimizes (24) by solving a QP using a structured SVM solver. Given the learned model w, one

can re-estimate the labels on the positive examples by running the current model: $\text{argmax}_{z \in Z_n} w \cdot \Phi(I_n, z_n)$. Felzenszwalb et al. [16] show that both these steps can be seen as coordinate descent on an auxiliary loss function that depends on both w and the latent values on positive examples $Z_{pos} = \{z_n : n \in pos\}$.

$$L_{SVM}(w, Z_{pos}) = \tag{25}$$
$$\lambda \frac{1}{2}||w||^2 + \sum_{n \in pos} \max(0, 1 - w \cdot \Phi(I_n, z_n)) + \sum_{n \in neg} \max_z (0, 1 + w \cdot \Phi(I_n, z))$$

6 Applications

In this section, we briefly describe the application of part-based models for pedestrian detection, human pose estimation, and tracking.

6.1 Pedestrian detection

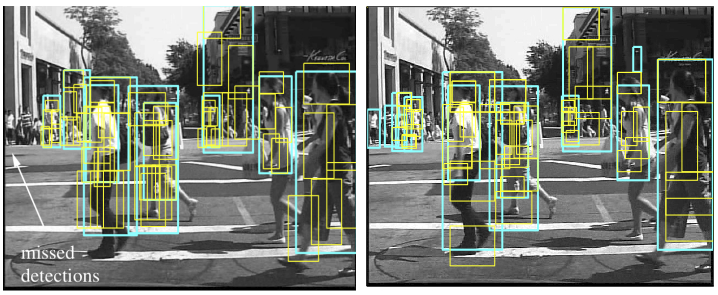

Fig. 8 On the **left**, we show the discriminative part model of [18] (shown in Fig. 7) applied to the Caltech Pedestrian Benchmark [11]. The model performs well for instances with sufficient resolution to discern parts (roughly 80 pixels or higher), but does not detect small pedestrians accurately. We show the multiresolution part model of [43] (**right**) which behaves as a part-model for large instances and a rigid template for small instances. By tailoring models to specific resolutions, one can tune part templates for larger base resolutions, allowing for superior performance in finding both large and small people.

One important consideration with part-based representations is that object instances must be large enough to resolve and distinguish parts - it is, for example, hard to discern individual body parts on a 10 pixel-tall person. [43] describe an extension of part-based models that allow them to behave as rigid templates when evaluated on small instances.

6.2 Pose estimation

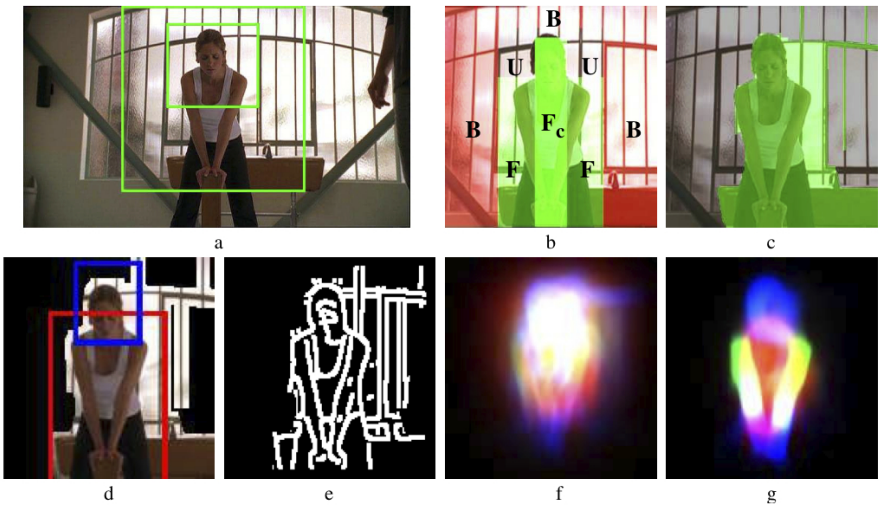

Fig. 9 The pose estimation algorithm of [22] begins by detecting upper bodies (using the discriminative part-model shown in Figure 7), performing a local foreground/background segmentation, and using the learned foreground/background appearance models to produce the final posterior marginal over poses shown in (**g**).

Popular benchmarks for pose estimation in unconstrained images include the parse dataset of [45] and the Buffy stickman dataset [21]. The dominant approach in the community is to use articulated models, where part locations $l_i = (x_i, y_i, \theta_i)$ include both pixel position and orientation. State-of-the-art methods with such an approach include [52, 31]. The former uses a large set of heterogenous image features, while the latter uses the HOG descriptor described here.

Appearance constraints: Part templates by construction must be invariant to clothing appearance. But ideally, one would like to use templates tuned for a particular person in a given image, and furthermore, tuned to discriminate that person from the particular background. [45] describe an iterative approach that begins with invariant edge-based detectors and sequentially learns color-based part models tuned to the particular image. Specifically, one can compute posterior marginals $P(z_i|x,w)$ given clothing-invariant templates w. These posteriors provide weights for image windows as to how likely they belong to particular body parts. One can update templates w to include color information by taking a weighted average of features computed from these image windows, and repeat the procedure. Ferrari et al. [22] describe an alternate approach to learning color models by performing foreground/background segmentations on windows found by upper-body detectors (Figure 9).

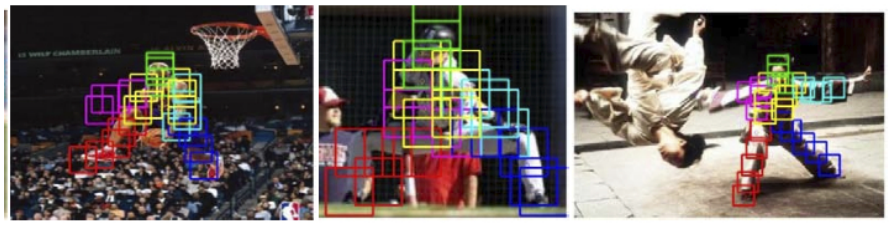

Fig. 10 We show pose estimation results from the flexible mixtures-of-part models from [64]. Rather than modeling parts as articulated rectangles, the authors use local mixtures of non-oriented part models to capture rotations and foreshortening effects.

Mixtures of parts: [64] point out that one can model small rotations and foreshortenings of a limb template with a "local" part-based model parameterized solely by pixel position. To model large rotations, one can use a mixture of such part models. Combining such models for different limbs, one can obtain a final part model where each part appearance can be represented with a mixture of templates. Importantly, the pairwise relational spring model must be extended to now model a collection of springs for each mixture combination, together with a co-occurence constraint on particular mixture combinations. For example, two parts on the same limb should be constrained to always have consistent mixtures, while parts across different limbs may have different mixtures because limbs can flex. Inference now corresponds to estimating both part locations l_i and mixture labels c_i. Inference on such models is fast, typically taking a second per image on standard benchmarks, while surpassing the performance of past work.

6.3 Tracking

To obtain a model for tracking, one can replicate a K-part model for T frames, yielding a spatiotemporal part model with KT parts. However, the relational model E must be augmented to encode dynamic as well as kinematic constraints - an arm part must lie near its parent torso part *and* must lie near the arm part estimated in the previous frame. One can arrive at such a model by assuming a first-order Markovian model of object state:

$$P(z_{1:T}, I_{1:T}) = \prod_t P(z_t|z_{t-1})P(I_t|z_t) \tag{26}$$

By introducing high-order dependancies, the motion model $P(z_t|z_{t-1})$ can be augmented to incorporate physical dynamics (e.g., minimizing acceleration). If we restrict ourselves to first-order models and redefine $I = I_{1:T}$, we can use the same scoring function as (1):

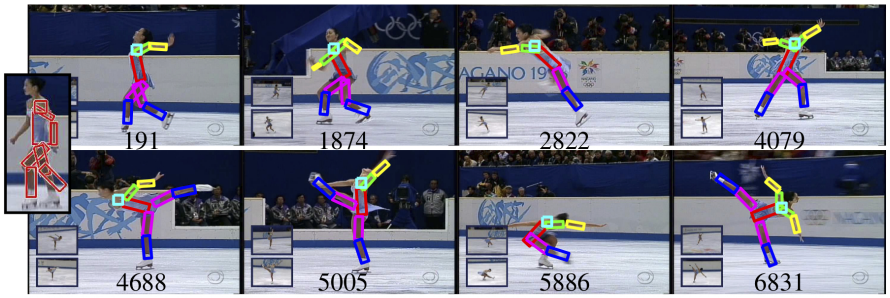

Fig. 11 We show tracking results from the appearance-model-building tracker of [48]. The styled pose detection (using edge-based part models invariant to clothing) is shown on the **left inset**. From this detection, the algorithm learns color appearance models for individual body parts. These models are used in a tracking-by-detection framework that tends to be robust and track for long sequences (as evidenced by the overlaid frame numbers).

$$S(I, z_{1:T}) = \sum_{i=1}^{KT} w_i \cdot \phi(I, l_i) + \sum_{i,j \in E} w_{ij} \cdot \psi(I, l_i, l_j) \tag{27}$$

where the relational graph $G = (V, E)$ consists of KT vertices with edges capturing both spatial and temporal constraints. Temporal constraints add loops to the model, making global inference difficult. An estimated arm must lie near its parent torso and the estimated arm in the previous frame.

A popular approach to inference in such tracking models is the use of particle filters [30, 54, 12]. Here, the distribution over the state of the object z_t is represented by a set of particles. These particles are propagated through the dynamic model, and are then re-weighted by evaluating the likelihood. However, the likelihood can be highly multi-model in cluttered scenes. For example, there maybe many image regions that locally look like a limb, which can result in drifting particles latching onto the wrong mode. A similar, but related difficulty is that such trackers need to be hand-initialized in the first frame. Note that drifting and the requirement for hand initialization seem to be related, as one way to build a robust tracker is to continually re-initialize it. Nevertheless, particle filters have proved effective for scenarios in which manually initialization is possible, there exist strong likelihood models (e.g., background-subtracted image features), or one can assume strong dynamic models (e.g., known motion such as walking).

Tracking by detection: One surprisingly effective strategy for inference is to remove the temporal links from (27), in which case inference reduces to an *independent* pose estimation task for each frame. Though computationally demanding, such "tracking by detection" approaches tend to be robust because an implicit tracker is re-initialized every frame. The resulting pose estimates will necessarily by temporally noisy, but one can apply low-pass filtering algorithms as a post-processing step to remove such noise [48].

Tracking by model-building: Model-based tracking should be easier with a better model. Ramanan and Forsyth [50] argue that this observation links together tracking and object detection; namely one should be able to track with a more accurate detector. This can be accomplished with a latent variable tracking model where object location *and* appearance are treated as unknown variables to be estimated. This is analagous to the appearance constraints described in Section 4.2, where an gradient-based part model was augmented with the latent RGB appearance.

One can apply this observation to tracking people: given an arbitrary video, part appearance models must be initially be clothing-invariant. But when using part model in a tracking-as-detection framework, one ideally would like part models tuned to the appearance of particular people in the video. Furthermore, if there exist multiple people interacting with each other, one can use such appearance-specific models to disambiguate different people. One approach to doing this is first detect people with a rough, but usable part model built on invariant edge-based part templates w_i. By averaging together the appearance of detected body parts, one can learn instance specific appearance models w_i'. One can exploit the fact that the initial part detection can operate at high-precision and low-recall; one can learn appearance from a sparse set of high-scoring detections, and then later use the known appearance to produce a dense track. This initial high-precision detection can be done *opportunistically* by tuning the detector for stylized poses such as lateral walking poses, where legs occupy a distinctive scissor profile [47].

7 Discussion and open questions

We have discussed part-based models for the task of detecting people, estimating their pose, and tracking them in video sequences. Part-based models have a rich history in vision, and currently produce state-of-the-art methods for general object recognition (as evidenced by the popular annual PASCAL Visual Object challenge [15]). A large part of their success is due to engineered feature representations (such as [10]) and structured, discriminative algorithms for tuning parameters. Various open-source codebases for part-based models include [17, 44, 14].

While detection and pose-estimation are most naturally cast as classification (does this window contain a person or not?) and regression (predict a vector of part locations), one would ideally like recognition systems to generate much more complex reports. Complexity may arise from more detailed description of the person's state, as well as contextual summaries that describe the relationship of a person to their surroundings. For example, one may wish to understand the visual attributes of people, including body shape [2], as well as the colors and articles of clothing being worn [37]. One may also wish to understand interactions with nearby objects and/or nearby people [66, 13].

Such reports are also desireable because they allow us to reason about non-local appearance constraints, which may in turn lead to better pose estimates and detection rates. For example, it is still difficult to estimate the articulation of lower arms

in unconstrained images. Given the attribute that a person of interest is wearing a full-hand shirt, one can learn a clothing appearance model from the torso to help aid in localizing arms. Likewise, it is easier to parse an image of two people hugging when one reasons jointly about the body pose of both people.

Such reasoning may require new representations. Perhaps part models provide one framework, but to capture the rich space of such visual phenomena, one will need a vocabulary of hundreds or even thousands of local part templates. This poses new difficulties in learning and inference. Relational models must also be extended beyond simple springs to include combinatorial constraints between visual attributes (one should not instance both a tie and skirt part) and flexible relations between people and their surroundings. To better understand clothing and body pose, inference may require the use of bottom-up grouping constraints to estimate the spatial layout of body parts, as well as novel appearance models for capturing material properties beyond pixel color.

References

1. M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. CVPR*, volume 1, page 4, 2009.
2. A. Balan and M.J. Black. The naked truth: Estimating body shape under clothing. In *European Conf. on Computer Vision*, pages 15–29. Citeseer, 2008.
3. T.O. Binford. Visual perception by computer. In *IEEE conference on Systems and Control*, volume 313, 1971.
4. L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *CVPR*, pages 1365–1372. IEEE, 2010.
5. C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 8–15. IEEE, 1997.
6. P. Buehler, M. Everingham, DP Huttenlocher, and A. Zisserman. Long term arm and hand tracking for continuous sign language TV broadcasts. In *Proc. BMVC*. Citeseer, 2008.
7. M. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. *Computer VisionECCV98*, pages 628–641, 1998.
8. T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *Computer VisionECCV98*, page 484, 1998.
9. D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. 2005.
10. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages I: 886–893, 2005.
11. P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009.
12. J. Duetscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *cvpr*, page 2126. Published by the IEEE Computer Society, 2000.
13. M. Eichner and V. Ferrari. We are family: joint pose estimation of multiple persons. *Computer Vision–ECCV 2010*, pages 228–242, 2010.
14. M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2d articulated human pose estimation software. http://www.vision.ee.ethz.ch/~calvin/articulated_human_pose_estimation_code/.
15. M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010.

16. P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *Computer Vision and Pattern Recognition, Anchorage, USA, June*, 2008.
17. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Discriminatively trained deformable part models. http://people.cs.uchicago.edu/~pff/latent/.
18. Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *IEEE PAMI*, 99(1), 5555.
19. P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.
20. R. Fergus, P. Perona, A. Zisserman, et al. Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2. Citeseer, 2003.
21. V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, June 2008.
22. V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: Retrieving people using their pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
23. T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *Proceedings of the 25th international conference on Machine learning*, pages 304–311. ACM New York, NY, USA, 2008.
24. MA Fischler and RA Elschlager. The representation and matching of pictorial structures. *Computers, IEEE Transactions on*, 100(1):67–92, 1973.
25. DA Forsyth and MM Fleck. Body plans. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 678–683. IEEE, 2002.
26. G. Hua, M.H. Yang, and Y. Wu. Learning to estimate human pose with data driven belief propagation. 2005.
27. S. Ioffe and D. Forsyth. Human tracking with mixtures of trees. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 690–695. IEEE, 2002.
28. S. Ioffe and D.A. Forsyth. Probabilistic methods for finding people. *International Journal of Computer Vision*, 43(1):45–68, 2001.
29. M. Isard. Pampas: Real-valued graphical models for computer vision. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1. IEEE, 2003.
30. M. Isard and A. Blake. Condensationconditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28, 1998.
31. S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference (BMVC)*, 2010.
32. S.X. Ju, M.J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. *fg*, page 38, 1996.
33. M.P. Kumar, A. Zisserman, and P.H.S. Torr. Efficient discriminative learning of parts-based models. In *CVPR*, pages 552–559. IEEE, 2010.
34. P. Kumar, P. Torr, and A. Zisserman. Learning layered pictorial structures from video. 2004.
35. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289. Citeseer, 2001.
36. X. Lan and D.P. Huttenlocher. Beyond trees: Common-factor models for 2d human pose recovery. In *CVPR*, volume 1, pages 470–477. IEEE, 2005.
37. M.W. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *CVPR*, volume 2. IEEE, 2004.
38. B. Leibe, A. Leonardis, and B. Schiele. An implicit shape model for combined object categorization and segmentation. *Toward Category-Level Object Recognition*, pages 508–524, 2006.
39. D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

40. D. Marr and H.K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 200(1140):269–294, 1978.
41. I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.
42. G. Mori, X. Ren, A.A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, 2004.
43. D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. *Computer Vision–ECCV 2010*, pages 241–254, 2010.
44. D. Ramanan. Learning to parse images of articulated bodies. http://www.ics.uci.edu/~dramanan/papers/parse/index.html.
45. D. Ramanan. Learning to parse images of articulated bodies. *NIPS*, 19:1129, 2007.
46. D. Ramanan and DA Forsyth. Finding and tracking people from the bottom up. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2. IEEE, 2003.
47. D. Ramanan, D.A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 271–278. IEEE, 2005.
48. D. Ramanan, DA Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):65–81, 2007.
49. D. Ramanan and C. Sminchisescu. Training deformable models for localization. In *CVPR*, volume 1, pages 206–213. IEEE, 2006.
50. Deva Ramanan and D. A. Forsyth. Using temporal coherence to build models of animals. *Computer Vision, IEEE International Conference on*, 1:338, 2003.
51. R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 700–714. Springer-Verlag, 2002.
52. B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *CVPR*, pages 422–429. IEEE, 2010.
53. B. Sapp, A. Toshev, and B. Taskar. Cascaded Models for Articulated Pose Estimation. *ECCV 2010*, pages 406–420, 2010.
54. H. Sidenbladh, M. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. *Computer VisionECCV 2002*, pages 784–800, 2002.
55. L. Sigal and M.J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *CVPR*, volume 2, pages 2041–2048. IEEE, 2006.
56. L. Sigal, M. Isard, B.H. Sigelman, and M.J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. *Advances in Neural Information Processing System*, 16, 2004.
57. J. Sivic and A. Zisserman. Video Google: Efficient visual search of videos. *Toward Category-Level Object Recognition*, pages 127–144, 2006.
58. E. Sudderth, M. Mandel, W. Freeman, and A. Willsky. Distributed occlusion reasoning for tracking with nonparametric belief propagation. *Advances in Neural Information Processing Systems*, 17:1369–1376, 2004.
59. E.B. Sudderth, A.T. Ihler, M. Isard, W.T. Freeman, and A.S. Willsky. Nonparametric belief propagation. *Communications of the ACM*, 53(10):95–103, 2010.
60. T.P. Tian and S. Sclaroff. Fast Multi-Aspect 2D Human Detection. *Computer Vision–ECCV 2010*, pages 453–466, 2010.
61. D. Tran and D. Forsyth. Improved Human Parsing with a Full Relational Model. *ECCV*, pages 227–240, 2010.
62. Y. Wang and G. Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. *ECCV*, pages 710–724, 2008.
63. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. *Computer Vision-ECCV 2000*, pages 18–32, 2000.
64. Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures of parts. In *CVPR*. IEEE, 2011.

65. C. Yanover and Y. Weiss. Finding the AI Most Probable Configurations Using Loopy Belief Propagation. In *Advances in neural information processing systems 16: proceedings of the 2003 conference*, page 289. The MIT Press, 2004.
66. B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. 2010.