Training Deformable Models for Localization

Deva Ramanan and Cristian Sminchisescu
Toyota Technological Institute at Chicago, Chicago, IL 60637
ramanan@tti-c.org, crismin@nagoya.uchicago.edu

Abstract

We present a new method for training deformable models. Assume that we have training images where part locations have been labeled. Typically, one fits a model by maximizing the likelihood of the part labels. Alternatively, one could fit a model such that, when the model is run on the training images, it finds the parts. We do this by maximizing the conditional likelihood of the training data.

We formulate model-learning as parameter estimation in a conditional random field (CRF). Initializing parameters with their maximum likelihood estimates, we reach the global optimum by gradient ascent. We present a learning algorithm that searches exhaustively over all part locations in an image without relying on feature detectors. This provides millions of examples of training data, and seems to avoid over-fitting issues known with CRFs.

Results for part localization are relatively scarce in the community. We present results on three established datasets; Caltech motorbikes [8], USC people [19], and Weizmann horses [3]. In the Caltech set we significantly outperform the state-of-the-art [6]. For the challenging people dataset, we present results that are comparable to [19], but are obtained using a significantly more generic model (devoid of a face or skin detector). Our model is general enough to find other articulated objects; we use it to recover poses of horses in the challenging Weizmann database.

1. Introduction

Deformable models have a long-standing history in the vision community beginning with pictorial structures [10] and deformable templates [11], and also include the recent active appearance models [5] and constellation models [4]. These models represent an object as a collection of parts, explicitly encoding both the appearance of a part and its spatial arrangement.

There have been numerous approaches that use these models for **detection**. They address questions of the form: given an image, is there a motorbike or not? Surprisingly, one can obtain state-of-the-art detection performance by ignoring spatial constraints (the so-called "bag of feature" models). We believe this fact suggests that: (1) shape is hard to learn with current methods and (2) one should consider more difficult recognition tasks such as **localization**: where is the motorbike, which way is it facing? Our work focuses on methods for learning and evaluating deformable models, given the task of localization.

A natural method of learning a deformable model is to fit the model to some observed instances. This is often formulated as maximum likelihood (ML). Given a collection of images where part locations have been labeled, one computes sample means and variances (assuming gaussian models). Even non-probabilistic approaches such as exemplar matching implicitly do this; here the mean is encoded by the exemplar.

An alternative is to tune parameters so that the model does well at a task (in our case, localization). Intuitively, we want to tune parameters so that the model, when run on a training image, recovers the labeled part locations. We show that by formulating our model as a conditional random field (CRF), we naturally optimize this criteria.

We demonstrate the resulting models on three datasets; Caltech motorbikes [8], USC people [19], and Weizmann horses [3]. The Caltech set is known to be easy for detection, but we use it to evaluate part *localization*. We surpass the best-reported results in [6]. We present an articulated model for human pose estimation that is comparable to [19] on their challenging set of people images. In contrast to [19], we use a generic articulated model devoid of a face detector or skin model. To demonstrate its generality, we train it to localize deforming horses in the challenging Weizmann dataset.

2. Related Work

Approaches for learning parts-based models can loosely be divided into generative, semi-supervised, and discriminative. Given labeled training data, generative methods follow the ML framework described above [5, 6, 7, 11, 14, 15, 20]. Alternatively, Weber *et al.* and Fergus *et al.* use EM to learn models from partially-labeled data [8, 27]. Their

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

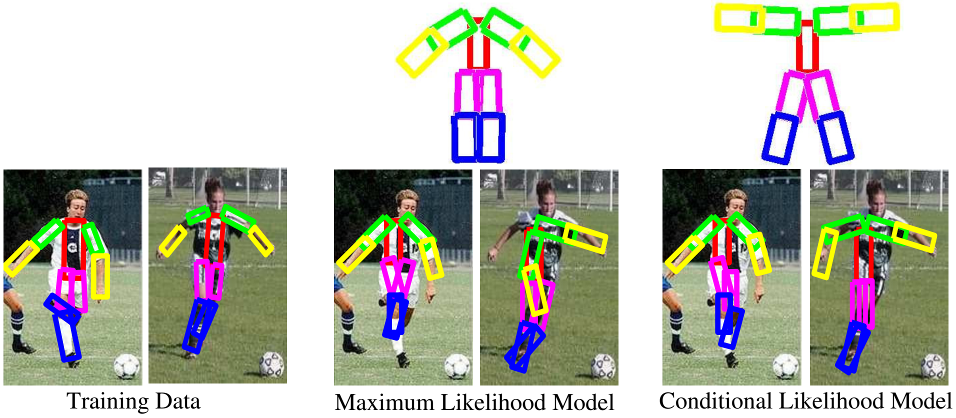

Figure 1. Our approach to model building. Assume we are given a collection of training images Im with labeled part locations L (we show 2 images on the **left**). Classic approaches learn the model Θ that maximizes the joint likelihood $\Pr(Im, L|\Theta)$. Assuming gaussian models, one does this by computing sample means and variances. We show the mean pose in the **top middle**. If we use the model to *infer* the pose in each training image, we often get incorrect localizations; the arm gets confused with the body (**bottom middle**). In this work, we learn a model that is *trained* to infer the correct poses in the training images. We do this by maximizing the conditional $\Pr(L|Im, \Theta)$. We show the new learned mean pose on the **right top**. By pulling the arm away from the body, the resulting model infers poses that are closer to the labeled training set (**bottom right**).

approach required knowing an image contains a motorbike, but not where its parts are located. Although our work here learns in a supervised framework, there is a close connection between our CRF optimization and EM (we look at this in Sec. 4).

Discriminative training of deformable models dates back to at least decision trees [1], convolutional neural nets [18], and include recent approaches such as [12, 13, 22]. In these cases, models are optimized for detection and not localization. Kumar and Hebert introduced CRFs for low-level vision [16]. They infer pixel labels from a loopy grid (and so require approximations for inference and learning), while we infer part locations on a tree-structured model.

There exist relatively few published results for localization of deformable models [6, 23]. One notable exception is human pose estimation. Most work involves tracking in video sequences, although approaches for pose estimation in static images exist [14, 19, 21, 24]. The unconstrained nature of the problem typically requires some limiting assumption such as uncluttered backgrounds, visible faces, and/or visible skin regions.

3. Deformable Part Model

We use a parts and structure model framework common in the previously mentioned approaches. Let us write the location of part i as $l_i = (x_i, y_i)$. We later extend l_i to encode part orientation for articulated models. We denote the configuration of a K part model as $L = (l_1 \dots l_K)$. Given model parameters Θ, the joint probability of a configuration L in an image Im is

$$\Pr(Im, L|\Theta) = \prod_{(i,j)\in E} \Pr(l_i|l_j) \prod_{i=1}^{K} \Pr(Im(l_i)|l_i) \prod_{l \in bg} \Pr(Im(l|bg)) \tag{1}$$

The first term captures part geometry, while the second term models the local image patch at each part. The last term models the image patches in the background. We assume the local probability functions are gaussian:

$$\Pr(l_i|l_j) = N(l_i - l_j; \mu_i, \Sigma_i) \tag{2}$$

The geometric model in Eq.2 has an intuitive interpretation as a "spring" that connects part i to part j. The spring has a rest position of μ_i and a stiffness encoded by Σ_i. We assume E is a known tree (see Sec. 4.2), and so each part i is connected to one parent j. We will often write the relative location of part i simply as $r_i = l_i - l_j$.

$$\Pr(Im(l_i)|l_i) = N(Im(l_i); \alpha_i, \Gamma_i). \tag{3}$$

The appearance model in Eq.3 is defined with a feature vector $Im(l_i)$ describing the image patch centered at (x_i, y_i). We can think of α_i as an image template for part i. We describe our feature vector representation in Sec. 6. Our final model is defined by the parameters of each gaussian $\Theta = \{\mu_i, \Sigma_i, \alpha_i, \Gamma_i, \alpha_{bg}, \Gamma_{bg}\}$.

Inference: In order to use the model to localize an object in an image Im, we need the posterior over part locations L:

$$\Pr(L|Im,\Theta) \propto \prod_{(i,j)\in E} \Pr(l_i|l_j) \prod_{i=1}^{K} \frac{\Pr(Im(l_i)|l_i)}{\Pr(Im(l_i)|bg)} \tag{4}$$

$$L_{MAP} = \text{argmax}_L \Pr(L|Im,\Theta) \tag{5}$$

$$L_{Bayes} = E_\Theta[L] = \sum_L L \Pr(L|Im,\Theta) \tag{6}$$

We use $E_\Theta[\cdot]$ to denote an expectation with respect to the posterior defined by model Θ. Given an image, one can localize an object by either computing the maximum *a posteriori* estimate (L_{MAP}) or the average location with respect to the posterior (the Bayesian estimate L_{Bayes}). If E is tree, both are computable by fast variants of dynamic programming [7]. One can also use the MAP estimate as a detector by thresholding the unnormalized posterior.

Learning: Assume we are given training images Im^t where part locations L^t have been labeled. (As a notation convention, we use subscripts to denote part numbers and superscripts to denote image numbers). The classic criteria for learning a model is to maximize the joint likelihood of the labeled data

$$\Theta_{ML} = \max_\Theta \prod_t \Pr(Im^t, L^t|\Theta) \tag{7}$$

$$= \max_{\mu,\Sigma} \prod_t \Pr(L^t|\mu,\Sigma) \max_{\alpha,\Gamma} \prod_t \Pr(Im^t|L^t,\alpha,\Gamma) \tag{8}$$

In the literature, this is often called maximum likelihood (ML) learning. Since the likelihood factors into Eq. 1, it suffices to find the ML estimates of the individual gaussian terms. This is done by *independently* computing sample means and variances. For example, we set μ_i to be the average position of part i with respect to part j.

In some ways, this independence is unintuitive. Suppose we learn a very accurate appearance template α_i. This suggests we do not need a strong spatial prior Σ_i (since we can find the part simply by matching the template α_i). This inter-dependence implies Θ_{ML} may not be optimal.

What is wrong with Θ_{ML}? Consider Fig.1. The mean pose in a collection of labeled people images tends to have the arms lying alongside the body. This is because, on average, a person tends to keep their arms there. However, such a pose may not be useful for localizing people because the estimated arms will be confused with the body. We do not want the most likely pose, but rather the pose that *produces the best estimates when used for inference*. We argue that ML may be the wrong criteria because it is not directly tied to inference.

The posterior in Eq. 4 is the precise quantity used for inference. We will show that learning a Θ which maximizes $\Pr(L|Im,\Theta)$ produces a model well-suited for localization. We call such a model Θ_{CL} because it maximizes the *conditional* likelihood of labels given the image. Computing Θ_{CL} is difficult because Eq.4 is not normalized. The implicit normalization factor is actually a function of all the parameters Θ. This means that, unlike ML learning, we cannot fit each parameter independently. Our resulting model, however, is equivalent to a tree-structured Conditional Random Field (CRF) [17]. We apply standard algorithms from that literature to learn Θ_{CL}.

4. Maximizing the conditional likelihood

Since are working directly with Eq.4, it will be convenient to simplify our appearance model by assuming both parts and the background have the same covariance Γ_{bg}. In that case we can write:

$$\frac{\Pr(Im(l_i)|l_i)}{\Pr(Im(l_i)|bg)} \propto \exp^{w_i^T \cdot Im(l_i)} \tag{9}$$

where

$$w_i = \Gamma_{bg}^{-1}(\alpha_i - \alpha_{bg}). \tag{10}$$

Given a set of labeled poses $\tilde{L}^t$, let us write the (log) conditional likelihood:

$$\mathcal{L}(\Theta) = \sum_t \log \Pr(\tilde{L}^t|Im^t,\Theta). \tag{11}$$

We find the Θ_{CL} that maximizes Eq.11 by gradient ascent. Decomposing $\Sigma_i^{-1} = C_i^T C_i$, we calculate the gradient as follows:

$$\frac{d\mathcal{L}}{d\mu_i} = C_i^T C_i \{\sum_t \tilde{r}_i^t - \sum_t E_\Theta[r_i^t]\} \tag{12}$$

$$\frac{d\mathcal{L}}{dC_i} = C_i \{\sum (\tilde{r}_i^t - \mu_i)^2 - \sum_t E_\Theta[(r_i^t - \mu_i)^2]\} \tag{13}$$

$$\frac{d\mathcal{L}}{dw_i} = \sum_t Im^t(\tilde{l}_i^t) - \sum_t E_\Theta[Im^t(l_i^t)] \tag{14}$$

where we recall $r_i = l_i - l_j$. These updates are similar to the standard equations found in the CRF literature. The first summation in each term computes "empirical averages" of our sufficient statistics. The second summation computes the expected statistics by averaging over the posterior under the current model Θ (Eq. 6). At the optimal setting Θ_{CL}, the two terms are equal (the gradient is 0). This implies

$$\sum_t \tilde{r}_i^t = \sum_t E_{\Theta_{CL}}[r_i^t]. \tag{15}$$

This captures our initial intuition: we want a model Θ_{CL} that, when used to *infer* the location of part i on a training image, tends to find the labeled location $\tilde{r}_i$.

Optimization: We initialize our model parameters Θ to Θ_{ML}, and then take fixed-size gradient steps until convergence. CRFs are known to be convex, so we are guaranteed to be at the global optimum upon convergence. In practice, we encounter stability issues when C_i is close to 0 (since we must invert it to get Σ_i). We follow this two-step strategy: we first optimize μ_i, w_i while holding C_i fixed at its ML estimate, and then optimize C_i (holding μ_i, w_i fixed) with very small gradient steps. We suspect more sophisticated second-order methods (common in CRF optimization [25]) should work better.

Relationship to EM: Although EM optimizes a very different criteria, algorithmically it is quite similar to our gradient procedure. The expected sufficient statistics in the above equations are the *exact* same quantities computed when learning a part model with EM [8]. This implies that systems which learn part models by EM can also learn CRFs (with a simple extension). During the E step, one computes expected sufficient statistics. Given training images with labeled part locations, one can also compute empirical estimates of those statistics. If the two are equivalent, the learned model is also an optimally trained CRF. If not, one updates the model Θ by taking a gradient step and re-computes the expected statistics.

4.1. Computing the expected sufficient statistics

To compute expectations (for either EM or a CRF update), we need to compute conditional marginals $Pr(l_i|Im)$ and conditional pairwise marginals $Pr(l_i, l_j|Im)$ from Eq. 4. If we assume a tree-structured model, we can compute them exactly in $O(N^2)$ with belief propagation, where N = number of part locations. However, since $N \approx$ number of pixels, this is still too expensive. Most learning approaches search over a small set of image locations returned by a feature detector. However, when training a discriminative model, we would like lots of data to avoid over-fitting. We show we can use the framework of Felzenszwalb and Huttenlocher [7] to compute the expectations over *all* part locations in sub-quadratic time. One can replace all N^2 computations with convolutions, which are $O(N \log N)$. These results also imply that EM can be performed exhaustively without requiring feature detection (as hypothesized in [9]).

To avoid numerical issues, we normalize all messages to sum to 1 as they are computed. The set of "upstream" messages from part i to its parent j are computed as:

$$m_i(l_j) \;\propto\; \sum_{l_i} \Pr(l_i|l_j) a_i(l_i) \qquad (16)$$

$$a_i(l_i) \;\propto\; exp^{w_i^T \cdot Im(l_i)} \prod_{k \in kids_i} m_k(l_i) \qquad (17)$$

For $l_i = (x_i, y_i)$, we can represent messages as 2D images. The image a_i is obtained by multiplying together response images from the children of part i and from the appearance model. Because $\Pr(l_i|l_j)$ is a gaussian (Eq. 2), we can compute message m_i by convolving the image a_i with a gaussian of covariance Σ_i, and shifting the result by μ_i (see [6]). At the root r, the image a_r is the true conditional marginal $\Pr(l_r|Im)$. Starting from the root, we pass messages downstream (from part j to i) to compute the remaining marginals. We also simultaneously compute expectations over pairwise marginals:

$$\Pr(l_i|Im) \propto a_i(l_i) \sum_{l_j} \Pr(l_i|l_j) \Pr(l_j|Im) \qquad (18)$$

$$E_\Theta[r_i] = \sum_{l_j} \Pr(l_j|Im) \sum_{l_i} \Pr(l_i|l_j) r_i a_i(l_i) \qquad (19)$$

$$E_\Theta[r_i^2] = \sum_{l_j} \Pr(l_j|Im) \sum_{l_i} \Pr(l_i|l_j) r_i^2 a_i(l_i) \qquad (20)$$

We compute Eq. 18 by convolving $\Pr(l_j|Im)$ with a gaussian kernel. To compute Eq. 19, note that the product $\Pr(l_i|l_j) r_i$ can be written as a function of the relative position $f(l_i - l_j)$. We compute the inner summation by convolving a_i with f, a weighted gaussian kernel. We average the result over $\Pr(l_j|Im)$ to obtain the final expectation. The same method applies for Eq. 19. Computing $E_\Theta[Im(l_i)]$ is straightforward once we have the conditional marginal $\Pr(l_i|Im)$.

4.2. Learning the tree structure E

Given labeled data, we would like to find the tree E_{CL} that maximizes the conditional $\Pr(L|Im, \Theta)$. Known methods exist for finding the tree E_{ML} that maximizes the joint $\Pr(Im, L|\Theta)$. One fits a spring model (μ, Σ) independently to each possible pair of parts by computing sample estimates. One then computes the spanning tree with the most rigid springs. Recall that model parameters cannot be fit independently in a CRF. Hence finding E_{CL} is difficult; in practice, we use E_{ML}. However, when restricting E to be a star graph, the optimal tree *is* efficiently computable. For a K part model there are K possible star graphs (each part taking its turn as the root). For each graph, we learn a CRF that optimizes $\Pr(L|Im, \Theta)$, and then select the graph with the highest probability.

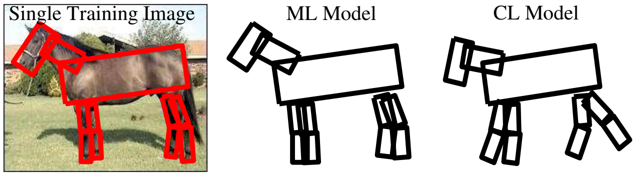

Figure 2. Given a *single* labeled example (**left**), the ML pose is just the labeled pose (**center**). The variance estimates are a user-defined deformation parameter. This is equivalent to building a deformable horse exemplar from a single image. If we use the exemplar to re-estimate the pose in the image, the legs are confused with each other because they are nearby and look similar. By training a pose that maximizes $\Pr(L|Im, \Theta)$, we learn an exemplar with legs that are spread apart (**right**). Using this *caricature* produces better results on image from which it was built.

5. Articulated Models

In this section, we outline the additions needed to learn articulated people models. We model each body part as an oriented rectangle of fixed size. We find people at multiple scales by searching over an image pyramid. We parameterize each oriented rectangle by $l_i = [x_i, y_i, u_i, v_i]$ where x_i, y_i is the location of the top endpoint, and (u_i, v_i) is unit vector that points down into the body. We update our shape model (Eq. 2) to:

$$\Pr(l_i|l_j) = N(t_j(l_i); \mu_i, \Sigma_i), \qquad (21)$$

where t_j = represents the relative part location l_i with respect to the oriented coordinate system of part j. The gaussian distribution on unit vectors is known as a Von Mises distribution [26]. We assume Σ is a block diagonal matrix consisting of Σ^{xy} and Σ^{uv}. Recall we initialize our gradient descent procedure with Θ_{ML}. The ML estimate of μ^{uv} is the renormalized mean of a set of given unit vectors. For a Von Mises distribution Σ^{uv} is a spherical gaussian with variance $\frac{1}{2\kappa}$. The ML estimate for κ is also readily computed from labeled training data [26]. The gradient steps for μ_i^{uv} and κ_i are as follows:

$$\frac{d\mathcal{L}}{d\mu_i^{uv}} = \kappa_i \{ \sum_t \begin{bmatrix} \tilde{u}_i^t \\ \tilde{v}_i^t \end{bmatrix} - \sum_t E_\Theta \begin{bmatrix} u_i^t \\ v_i^t \end{bmatrix} \} \qquad (22)$$

$$\frac{d\mathcal{L}}{d\kappa_i} = \mu_i^{uv} \cdot \{ \sum_t \begin{bmatrix} \tilde{u}_i^t \\ \tilde{v}_i^t \end{bmatrix} - \sum_t E_\Theta \begin{bmatrix} u_i^t \\ v_i^t \end{bmatrix} \} (23)$$

After updating μ_i^{uv} by a gradient step, we re-normalize it to unit length. Intuitively, Eq.23 does not contain any squared terms because the squared difference between two unit vectors simplifies to their dot product [26]. We apply the same techniques from Sec.4.1 by using 3D convolutions to compute the expectations in sub-quadratic time.

Learning from one example: Consider the task of learning deformable models from a single example. This situation is encountered in exemplar-based approaches for recognition. Such approaches seem to be highly successful for object recognition [2]. One can view exemplars as ML estimates fit to one example. The estimated mean is just the sample itself, while the variance is a user-defined deformation parameter. Using the exemplar to re-estimate the pose in the training image might fail if there are ambiguous parts or clutter (Fig 2). Intuitively, a good exemplar should re-estimate the pose it was constructed from. To do this, we might need a *caricature* of the original pose that accentuates discriminative characteristics. Fitting a pose to the conditional likelihood precisely accomplishes this.

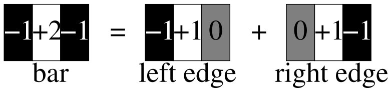

Figure 3. We define the image feature $Im(l_i)$ for an articulated part as the response of an oriented bar detector. A standard bar template can be written as the summation of a left and right edge template. The resulting detector suffers from many false positives, since *either* a strong left or right edge will trigger a detection. A better strategy is to require *both* edges to be strong; such a response can be created by computing the minimum of the edge responses as opposed to the summation.

6. Appearance descriptor $Im(l_i)$

We use two different appearance models in our experimental results; one for 2D models and one for articulated models.

To facilitate comparison of our 2D models with [6], we use an implementation of their part model. Here, a part is represented by 50×50 pixel patch. To compute $Im(l_i)$, we first compute oriented canny edges and separate the result into 4 orientation planes. We dilate each plane with a mask with a 2.5 pixel radius. To reduce the size of the descriptor, we bin each dilated image into an 11×11 grid using soft binning. The final descriptor is $11 \times 11 \times 4 = 484$ dimensional. This implies that our appearance weights w_i are also 484 dimensional. Typically, one might expect over-fitting when training such a high dimensional model. We appear to avoid this problem because of the exhaustive search described in Sec 4.1. We use the training set in [8] which contains 400 images; this means we train w_i with more than 10 *million* image patches.

For our articulated model, we set $Im(l_i)$ to be a scalar representing the response of a bar detector. One might construct a bar filter using a Haar-like template of a light bar flanked by a dark background (Fig. 3). To ensure a zero DC response, one would weight values in white by 2 and values in black by -1. We observe that a bar template can be decomposed into a left and right edge template $f_{bar} = f_{left} + f_{right}$. Denoting an entire image with Im

0-7695-2597-0/06 $20.00 © 2006 IEEE

and convolution by $*$, we write the response as

$$Im * f_{bar} = Im * f_{left} + Im * f_{right}.$$

In practice, using this template results in many false positives since either a single left or right edge triggers a response. We found taking a *minimum* of a left and right edge template resulted in a better response function:

$$\min(Im * f_{left}, Im * f_{right}). \qquad (24)$$

With judicious bookkeeping, we can use the same edge templates to find dark bars on light backgrounds. We compute the feature $Im(l_i)$ at all image locations by taking the log of the response image in Eq.24. We explicitly search over 15 orientations for each fixed-size limb. To find objects at multiple scales, we search over an image pyramid.

7. Results

Experimental results for part localization is scarce in the community. We have performed localization experiments on 3 standard datasets, the Caltech motorbikes [8], USC people [19], and the Weizmann horse set [3]. Given labeled training data from each dataset, we build both maximum likelihood Θ_{ML} and conditional likelihood Θ_{CL} models. We localize parts in a test image by computing the MAP estimate of part locations. We use efficient dynamic programming techniques that compute L_{MAP} in a few seconds per image [7]. We make all of our models translation invariant by setting Σ_{root} to be very large (we do not optimize Σ_{root} during learning).

Caltech motorbikes: The Caltech dataset is known to be relatively easy for detection; we use it as benchmark for *localization*. Crandall *et al* [6] demonstrate quite good performance on the motorbike set by ML training of star-like models. We train a star model using the same labeled training data (kindly provided by the authors). Interestingly, the means μ_i and appearance weights w_i trained by CL are equivalent to their ML estimates. However, the covariances Σ_i are much larger (Fig.4). This results in localization performance that surpasses the state-of-the-art (Table 1). This is because the part models are so strong that they need only a little guidance from a spatial prior. Consider the rear wheel model: by itself, it is an extremely accurate detector but for the fact that it is confused by the front wheel. It requires only a weak spatial prior to resolve this ambiguity. This interdependency between the spatial prior and the part model is lacking in the ML framework, since the model parameters for each are fit *independently* (Eq.8).

USC people: The USC people dataset is challenging set of 20 pictures of people in various poses [19] (kindly provided to us by the authors). We split the data in half into a training and testing set. The ML and CL model learned from the training images (and their mirror-flipped versions) are shown in Fig.1. The CL model learns a rest pose where the arms and legs lie away from the body. This helps during localization because the model will tend to be less confused by edges near the body. We show results for the test set in Fig. 5. We quantitatively evaluate results in Tab. 2. The pose recovery algorithm used by Lee and Cohen is initialized by a face detector and is tuned to find skin pixels; hence it is designed for frontally facing people with uncovered limbs. Our articulated part model from Sec. 6 is quite generic (as we use it to also find horses). We obtain error rates for certain body parts that are comparable to [19] (see Table 2).

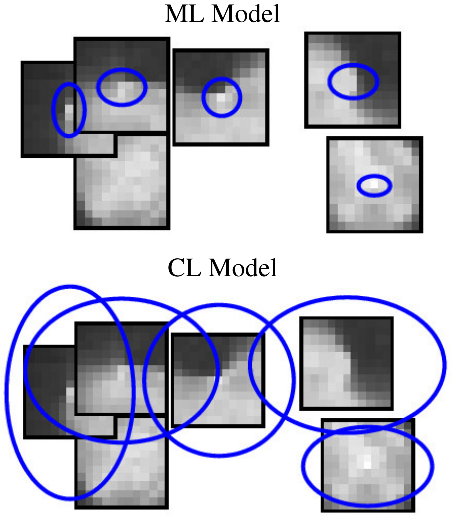

Figure 4. Learning a star model for the Caltech motorbikes. On the **top** is our implementation of the ML model learned by [6] (we assume diagonal Σ_i and plot ellipses at 1 standard deviation). On the **bottom**, the CL model has significantly larger Σ_i. This is because the part appearance models are so strong that only little guidance from a spatial prior is needed. The CL model produces better localizations as shown Table1.

Weizmann horses: The Weizmann horse dataset is a well-known collection of images used to evaluate segmentation. We are not aware of any results presented for part localization. We hand-labeled the first 40 images with ground truth locations, and learned an articulated model from the first 20 images. We show the learned models and test image results in Fig. 6. The CL model almost always localizes the body and most legs correctly, though it often has difficulties with the head. These results are impressive given the variety in appearance and pose for this dataset.

7.1. Discussion

We specifically address the recognition task of **localization**. By focusing on that task, we have developed a new criteria for optimizing part-based models. Instead of learning a model that best matches some labeled poses, we learn the model that best *localizes* those poses. This subtle difference often leads to very different models because the objective

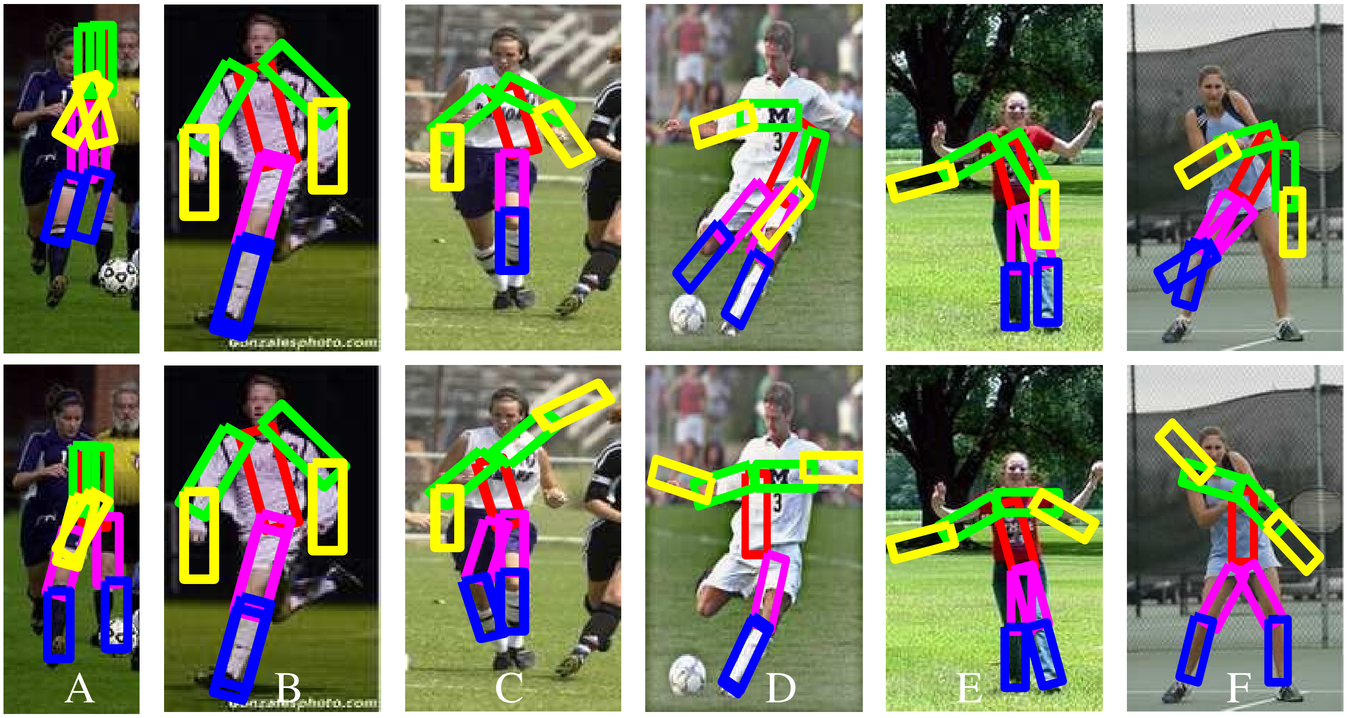

Figure 5. Finding people in the USC dataset. On the **top**, we show poses localized by Θ_{ML}. On the **bottom**, we show poses localized by Θ_{CL}. This data is quite challenging. Many images contain other people in the background (A,C), limb-like clutter (C), and self-occlusion (B,D). The CL model performs better than the ML model because it is less confused by edges close to the body. An exception is (C), where the spread-eagle spatial prior (from Fig. 1) forces the CL model to snap onto limb-like clutter in the background. In general, the CL model does well at finding the torso and legs, but often misses the arms. We show in Table 2 that we localize torsos and legs just as well as specialized approaches that exploit face and skin detection [19].

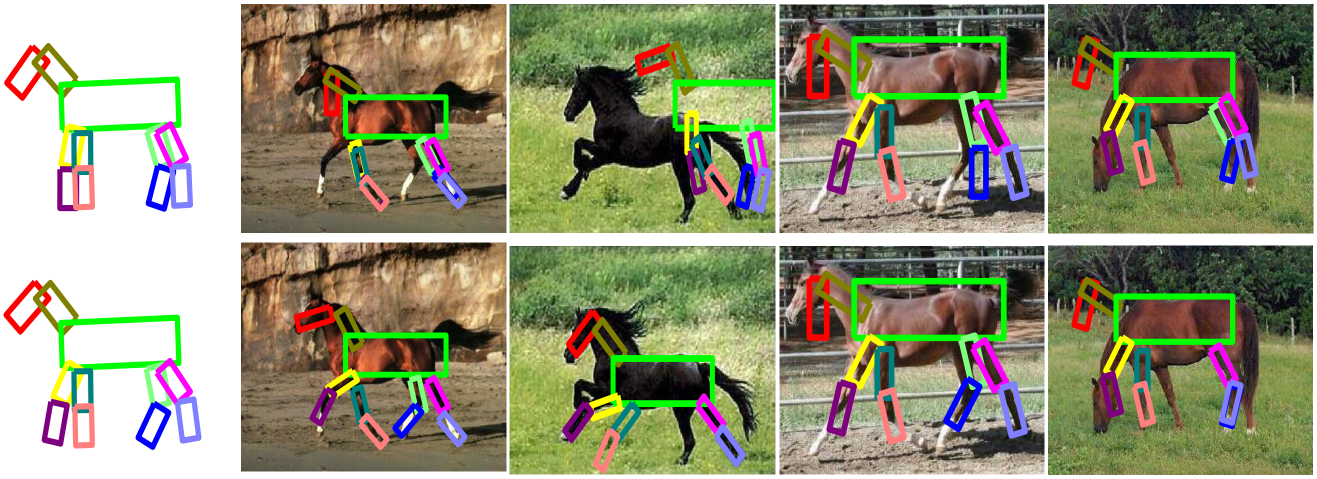

Figure 6. We can localize horses with our articulated model. On the **top**, we show poses localized by Θ_{ML}. On the **bottom**, we show poses localized by Θ_{CL}. Looking at the learned models (**left**), we see the CL model learns a more spread out rest pose (similar to Fig 1). This dataset is known to be challenging because of the variation in appearance and pose. Our CL model consistently achieves good localizations; the body and many of the legs are almost always correctly localized (although the estimates for left/right limbs can be incorrect). We look at quantitative results in Table 3.

is discriminative (rather than generative) and the model parameters are jointly learned (rather than independently). We demonstrate these models on challenging datasets, achieving or surpassing state-of-the-art results.

Acknowledgments: Thanks to David Crandall, Pedro Felzenszwalb, and Mun Wai Lee for helpful discussions

Localization Results for Caltech Motorbikes

	Rear	Front	Head	Tail	Seat	Seat
	wheel	wheel	light	light	Back	Front
ML	4.19	3.22	13.97	11.58	13.17	9.46
CL	2.88	2.44	12.49	7.95	10.39	6.77

Table 1. To evaluate localization, we look at the (90% alpha-trimmed) mean euclidean error of each part, measured with respect to a canonical car width of 200 pixels (as in [6]). Our average error across all parts for the CL model is 7.15. This compares favorable with the best-reported error of 12.9 [6]. This significant reduction seems to stem from the looser spatial model learned by the conditional likelihood model.

Localization Results for USC People

	Sho.	Elbow	Wrist	Hip	Knee	Ankle
ML	21.2	21.4	38.3	11.2	15.3	21.5
CL	17.9	21.9	39.7	7.8	12.3	17.2

Table 2. Our error rates in (pixel) root mean squared error for the USC dataset. Our models struggle to find arms, but the CL model localizes torsos and legs fairly well. Our error rates for those body parts are comparable to the average error of 14.9 reported in [19] (error for individual body parts were not given). Our results are impressive given that [19] uses a face detector and a skin model. Our part appearance models are quite generic; we show they can also be used to find other articulated objects such as horses in Fig. 6.

Localization Results for Weizmann horses

	Nose	Ear	Sho.	Knee	Hoof	Rear
ML	50.9	38.6	24.4	24.7	27.13	25.7
CL	45.9	34.2	19.1	19.8	22.72	20.0

Table 3. Our error rates in (pixel) root mean squared error for the Weizmann dataset. These are computed with respect to a canonical horse width of 300 pixels. The average error for the ML model is 27.9, while the CL model is 23.1. Given the variety in appearance and pose in the dataset, we do quite well at localizing the main body and legs. The head proves difficult; we might do better by learning a specific head model rather than using our generic limb model.

about their work.

References

[1] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE T. Pattern Analysis and Machine Intelligence*, 19(11):1300–1305, 1997.

[2] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, 2005.

[3] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, 2002.

[4] M. Burl, M.Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV*, pages 628–641, 1998.

[5] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *European Conference on Computer Vision*, 1998.

[6] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *CVPR*, 2005.

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *Int. J. Computer Vision*, 61(1), January 2005.

[8] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[9] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, 2005.

[10] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 1(22):67–92, January 1973.

[11] U. Grenander, Y. Chow, and D. Keenan. *Hands: a pattern theoretic study of biological shapes*. Springer-Verlag, 1991.

[12] A. B. Hillel, T. Hertz, and D. Weinshall. Efficient learning of relational object class models. In *ICCV*, 2005.

[13] A. Holub and P. Perona. A discriminative framework for modelling object classes. In *CVPR*, 2005.

[14] S. Ioffe and D. Forsyth. Probabilistic methods for finding people. *Int. J. Computer Vision*, 2001.

[15] M. Kumar, P. Torr, and A. Zisserman. Extending pictorial structures for object recognition. In *BMVC*, 2004.

[16] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *ICCV*, 2003.

[17] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[18] Y. LeCun and Y. Bengio. Pattern recognition and neural networks. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, MA, 1995.

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

[20] T. Leung, M. Burl, and P. Perona. Finding faces in cluttered scenes using random labelled graph matching. In *Int. Conf. on Computer Vision*, 1995.

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

[22] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In *NIPS*, 2004.

[23] D. Ramanan, D. Forsyth, and K. Barnard. Detecting, localizing, and recovering kinematics of textured animals. In *CVPR*, June 2005.

[24] X. Ren, A. C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *ICCV*, 2005.

[25] F. Sha and F. Pereira. Shallow parsing with conditional random feilds. In *HLT/NAACL*, 2003.

[26] H. Shatkay and L. P. Kaelbling. Heading in the right direction. In *ICML*, 1998.

[27] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV (1)*, pages 18–32, 2000.