Mixture of Parts Revisited: Expressive Part Interactions for Pose Estimation

Anoop R Katti
IIT Madras
Chennai, India
akatti@cse.iitm.ac.in

Anurag Mittal
IIT Madras
Chennai, India
amittal@cse.iitm.ac.in

Abstract

Part-based models with restrictive tree-structured interactions for the Human Pose Estimation problem, leave many part interactions unhandled. Two of the most common and strong manifestations of such unhandled interactions are self-occlusion among the parts and the confusion in the localization of the non-adjacent symmetric parts. By handling the self-occlusion in a data efficient manner, we improve the performance of the basic Mixture of Parts model by a large margin, especially on difficult poses. We address the confusion in the symmetric limb localization using a combination of two complementing trees, showing an improvement in the performance on all the parts with a very small trade-off in the running time. Finally, we show that the combination of the two solutions improves the results. We compare our HOG-based method with other methods using similar features and report results equivalent to the best method on two standard datasets with a large reduction in the running time.

1. Introduction

Human Pose Estimation in a 2D image is the task of detecting the presence of humans in the image and localizing their body parts. This problem is motivated by its potentially enormous applicability in high-level vision tasks such as Action Detection, Human Computer Interaction, Gesture Recognition, automatic analysis of videos of people etc.

A challenge unique to the human pose estimation problem is the large articulation that characterizes the human body. The most successful approaches are based on part-based models [5, 4, 3]. Here, the human body is modeled as an articulation of deformable body parts, flexibly connected to each other via spring-like connections. The appearance of each part is modeled independently. Due to dividing the entire body as an articulation of smaller parts, part-based model can handle a combinatorially large number of articulations.

Estimating pose while modeling all the interactions among the parts amounts to inferencing on a loopy-graph, which is intractable [12]. Yet, algorithms performing approximate inference have been proposed [8, 17, 18, 23], but at the cost of a high running time. Therefore, in order to perform efficient pose estimation using Dynamic Programming, the interactions between the parts are typically restricted to a tree structure [4, 5, 27] (Fig 4(a)). However, the downside of such a restriction is that it fails to capture the interactions between the non-adjacent parts. Two strong and frequently occurring manifestations of such unhandled interactions are (i) self-occlusion among the body parts producing large variations in part appearances (*e.g.* a fully vis-

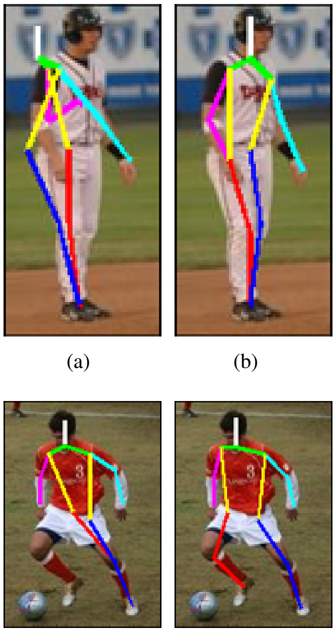


Figure 1. Example pose estimation by Yang *et al*. [27](a, c) and our model (b, d). Erroneous pose estimation in (a) is due to the unhandled self-occlusion (here, of the upper-arm by the torso) and in (c) is due to the absence of a kinematic constraint between the left and the right lower limbs.

ible upper arm vs an upper arm occluded by the torso in the side-view, Fig 1(a)) and (ii) a confusion in the localization of the non-adjacent symmetric limbs due to the absence of a kinematic constraint between them (*e.g.* in the tree shown in Fig 4(a), the left and the right legs are not kinematically constrained. Therefore, predictions of both the legs may localize on one of the legs as shown in Fig 1(c)).

In order to perform inference tractably while still capturing higher order interactions among parts, more recent works model combinations of physically close body parts, instead of a single part. For example, [19, 24] use latent nodes for the combined parts to capture the higher order spatial relations. Each type of any latent node prefers certain locations and types of its smaller constituent observed nodes. [7, 13, 14, 26] use many larger rigid templates spanning multiple parts called Poselets. Poselets handle part interactions by capturing the variations in the appearance caused by them. Depending on which of the poselets are activated, images are further processed to localize the parts. Other methods to handle the higher order interactions perform clustering in the pose space at semi-global [16] or global [10, 11] level and learn cluster-specific deformable models.

The problem with the above methods is that the variations in such "superparts" increase combinatorially as a function of the number of constituent parts. Therefore, when compared to part-level modeling, many more latent-node-types/templates/clusters are required to handle such large variations, consequently demanding a larger training data and a higher test time.

Different from the above methods, some of the latest works [9, 21, 20] improve the underlying features and the learning algorithm using the deep networks obtaining excellent part detectors followed by imposing cursory spatial constraints to eliminate the false detections.

Mixture of Parts (MoP) model, proposed by Yang and Ramanan [27], has been shown to be versatile through its successful application in various problems of computer vision [2, 6, 27, 28]. This is mainly due to the flexibility of modeling the multi-modal appearances of parts by learning a mixture of templates per part, instead of a single template. Particularly, Desai and Ramanan's [2] method of handling the self-occlusion within the MoP framework, while maintaining the tree structured interactions with part-level modeling, is promising.

In our paper, we focus on furthering the theory of Human Pose Estimation by proposing two modular and efficient improvements over Desai and Ramanan [2] to handle (i) the self-occlusion in a more data efficient manner and (ii) the confusion in localization of the non-adjacent symmetric parts. The improvements are modular because each solution can be applied independently, giving different strengths to the base MoP model or in combination, giving the best performance; they are efficient because they maintain the tree structured interactions and part-level modeling.

We demonstrate our ideas using the traditional HOG features and hence compare against the most recent methods using similar features. On two standard datasets, namely the LSP [10] and the IP [15], we report results that are on par with the best HOG-based method [14] with a large reduction in the running time.

2. Part-Based Model

The part-based model has been shown to be very powerful since it can handle large arbitrary articulations compared to a full-object model [3, 4] . One popular implementation of the part-based model is the Mixture of Parts model [27]. In this model, the appearance of every part is modeled using a mixture of templates, rather than a single template. This makes the method more robust to variations in the appearance of the parts. In this paper, we use the Mixture of Parts model as our baseline upon which we build our ideas.

In this section, we first review the Mixture of Parts model. This is followed by a review of the approximations made by the part-based model for efficient estimation of the pose and two issues arising from it, namely the self-occlusions and the confusion in the localization of the symmetric parts due to insufficient constraints. We also review Desai and Ramanan's Phraselets [2] approach for deriving the part mixtures to address the issue of self-occlusion.

2.1. Mixture of Parts (MoP) Model

Let I be the given image and let $G = (V, E)$ be the MoP model, where V is the set of parts and E is the set of pairwise constraints between the connected parts. Each part i is parameterized by (p_i, t_i), where $p_i = (x_i, y_i)$ is the pixel location and t_i is the mixture type in the mixture of templates for part i. Let $(\mathbf{p}, \mathbf{t})$ represent a pose configuration, where $\mathbf{p} = [p_1 \dots p_{|V|}]^T$ and $\mathbf{t} = [t_1 \dots t_{|V|}]^T$. Then the MoP model, parameterized by $(\mathbf{w}, \mathbf{b})$, scores a pose configuration $(\mathbf{p}, \mathbf{t})$ on an image I as:

$$S(I, \mathbf{p}, \mathbf{t}; \mathbf{w}, \mathbf{b}) = \sum_{i \in V} \left[w_i^{t_i} \cdot \phi(I, p_i) \right] + \sum_{(i,j) \in E} \left[w_{i,j}^{t_i} \cdot \psi(p_i - p_j) + b_{i,j}^{t_i, t_j} \right] \quad (1)$$

The first term in (1) scores the matching of part-type specific template $w_i^{t_i}$ to the HOG [1] features $\phi(I, p_i)$ extracted from the image I at p_i.

The second term in (1) enforces (a) part-type specific kinematic constraints: $\psi(p_i - p_j) = [-dx, -dx^2, -dy, -dy^2]$, where $dx = (x_i - x_j - \mu x_{i,j}^{t_i})$, $\mu x_{i,j}^{t_i}$ is the average difference in the $x-$values between part i and its parent j in the training images with type t_i

for part i and dy is similarly defined; (b) type compatibility constraints between a part and its parent: $b_{ij}^{t_i t_j}$ scores the compatibility of type t_i of part i and type t_j of its parent j.

Given a test image I, the objective is to infer the max-scoring pose configuration, i.e. $(\mathbf{p}^*, \mathbf{t}^*) = argmax\ S(I, \mathbf{p}, \mathbf{t}; \mathbf{w}, \mathbf{b})$ or the set of pose configurations above some threshold in the case of multiple persons in an image.

Learning: The model parameters, $(\mathbf{w}, \mathbf{b})$, are learned using structured SVM. Let $(\mathbf{p}^n, \mathbf{t}^n)$ be the ground-truth pose configuration of the n^{th} positive training image I^n. Then, $\beta = (\mathbf{w}, \mathbf{b})$ is obtained by solving:

$$\arg\min_{\beta,\xi\geq 0} \frac{1}{2}\|\beta\|^2 + C\sum_n \xi_n \tag{2}$$
$$s.t.\ \forall n \in pos\ \ S(I^n, \mathbf{p}^n, \mathbf{t}^n; \beta) \geq 1 - \xi_n$$
$$\forall n \in neg,\ \forall p, t\ \ S(I^n, \mathbf{p}, \mathbf{t}; \beta) \leq -1 + \xi_n$$

The above quadratic program solves for the lowest norm β that scores the ground-truth pose configurations in positive images above 1 and negative images below -1. This is solved using the dual coordinate descent solver of [27].

2.2. Part Interactions

The space of all possible poses, $(\mathbf{p}, \mathbf{t})$, is combinatorially large. Therefore, to search for a pose that maximizes the score, $S(I, \mathbf{p}, \mathbf{t}; \mathbf{w}, \mathbf{b})$, is very hard. However, if the connections between the parts are restricted to a tree-structure, the maximization of $S(I, \mathbf{p}, \mathbf{t}; \mathbf{w}, \mathbf{b})$ over this space can be performed efficiently using Dynamic Programming [4, 27]. The structure of the tree that is generally used is as shown in the Fig 4(a) where strongest interactions are taken to be the kinematic constraints between the adjacent parts in the human body. The downside of such a restriction is that many other interactions among the parts remain unhandled, limiting the expressive power of the model. Two commonly occurring, yet significant manifestations of such unhandled part interactions are (a) self-occlusion and (b) a confusion in the localization of unconnected symmetric parts.

Self-occlusion occurs when one part of the body partially or fully occludes another part. Due to heavy articulations in the human body, almost any part can potentially occlude any other part. This leads to a large variation in the appearance of a part caused by other parts that may not be directly connected to it in the tree. For example, a fully visible upper arm appears very different than an upper arm partially occluded by the torso 1(a). Similarly, a fully visible head appears very different than a head occluded by a lifted arm.

Another important consequence of limiting the part-interactions to a tree structure is the confusion in the localization of the unconnected symmetric limbs. For example, in the tree shown in Fig 4(a), no kinematic constraint exists between the left and the right legs. This often leads to an overlap of their predictions with one of the legs (Fig 1(c)). On the other hand, since the left and the right arms are constrained through the shoulder connections, their locations are much more accurately predicted in this tree structure.

2.3. Phraselet Clustering

Self-occlusion leads to an overlap of the parts in an image, thus creating a change in the observed appearance of the parts. Desai and Ramanan [2] handle these variations by learning a mixture of templates for every part. These templates capture the appearances of different clusters of overlap patterns, called Phraselets. An overlap pattern for part i is represented by the relative placements of parts that are in close vicinity of i. For example, Fig 2 shows three overlap patterns around the upper-arm. Fig 2(a) and Fig 2(b) are differentiated since the parts close to the upper-arm are different, while Fig 2(a) and Fig 2(c) are differentiated since the the same parts that are close to the upper-arm are differently placed around it.

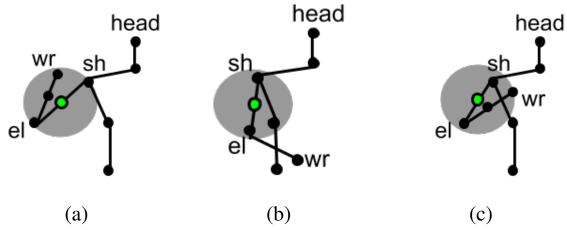


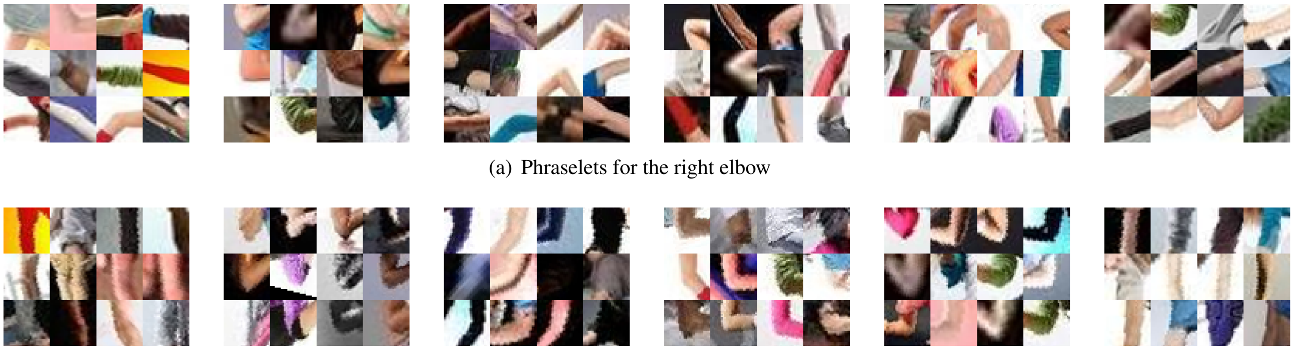
Figure 2. Overlap patterns for the upper-arm. sh: shoulder, el: elbow, wr: wrist.

Formally, an overlap pattern for a part i is expressed as a vector of weighted relative placements of the other parts, where the weighting is based on the distance of the parts from the part i; smaller the distance, larger the weight and vice versa. Mathematically, it can be written as, $\boldsymbol{\Delta} = [\boldsymbol{\Delta}_1^T \ldots \boldsymbol{\Delta}_{|V|}^T]^T$, where $\boldsymbol{\Delta}_j = exp(-\|\boldsymbol{\delta}_j\|) \cdot (\boldsymbol{\delta}_j)$, $\boldsymbol{\delta}_j = [x_j - x_i,\ y_j - y_i]^T$ and $|V|$ is the number of parts. For all the training images, $\boldsymbol{\Delta}$'s are formed and similar overlap patterns are clustered together using k-means clustering. These clusters are the Phraselets for the part i. A mixture of k templates are learned for the k Phraselets of the part i. Phraselets for the right elbow are shown in Fig 3(a).

In the following sections, we present our improvements on Desai and Ramanan's Phraselets [2] in handling self-occlusion, followed by our proposed solution for the confusion in the localization of non-adjacent symmetric part.

3. Improvements in Handling Self-Occlusion

Consider the overlap patterns around the upper-arm in Fig 5(a) and Fig 5(b). It can be seen that the patterns are



(a) Phraselets for the right elbow

(b) Rotation Normalized Phraselets for the right elbow

Figure 3. Clusters of overlap patterns around the right elbow by Phraselets and Rotation Normalized Phraselets.

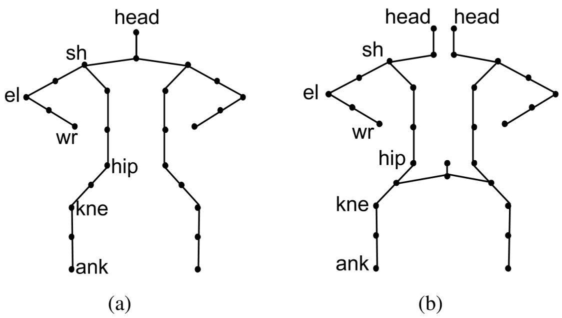


Figure 4. (a) Tree used by [4, 27], the *Upper-Constrained Tree*. (b) the *Lower-Constrained Tree*.

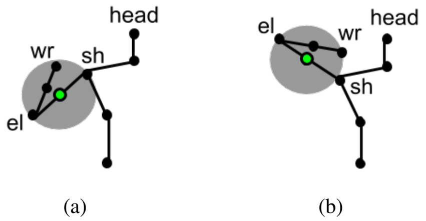


Figure 5. Similar overlap patterns at different orientations for the upper-arm. sh: shoulder, el: elbow, wr: wrist.

very similar, except that they are differently oriented. However, due to the rotation, the relative placements of the parts are different. Therefore, these patterns are placed in different clusters and different templates are learned for them. This method works well if the patterns repeat enough number of times at many different orientations in the training set such that templates can be learned for each orientation. However, this would require very large training sets which may not always be available.

We note that the data efficiency of Phraselets can be significantly improved by representing the overlap patterns in an orientation agnostic manner. Towards this, we specify an overlap pattern for part i by the relative placements of parts that are in close vicinity, normalized according to the orientation of i. More precisely, we modify the weighted relative placement, $\mathbf{\Delta}_j$, to weighted rotation-normalized relative placement, $\mathbf{\Delta}_j = exp(-\|\boldsymbol{\delta}_j\|) \cdot (R_{-\theta_i}\boldsymbol{\delta}_j)$, where θ_i is the orientation of part i and R_θ is the rotation matrix for angle θ.

Further, we note that appearance variation of a part is generally caused by only a few other parts that often happen to come physically close to it. These are determined by defining a set of occluding parts, $\mathcal{O}_i$, as the set of parts that overlap with i in at least m (=100) training images. For every positive training image, $\mathbf{\Delta} = [\mathbf{\Delta}_1^T \dots \mathbf{\Delta}_{|\mathcal{O}_i|}^T]^T$ is formed, where $|\mathcal{O}_i|$ is the number of occluding parts of part i. As before, the set of $\mathbf{\Delta}$'s obtained from the training images is clustered using k-means and a template is trained per cluster. Now, each cluster represents a Rotation Normalized Phraselet. Occluding parts facilitate formation of cleaner clusters. Fig 6 schematically shows the difference in clustering for the upper-arm between Phraselets [2] and the Rotation Normalized Phraselets. Fig 3(b) shows our Rotation Normalized Phraselets of the right elbow compared with the Phraselets in Fig 3(a). It can be observed that, due to rotation normalization, a wider variety of overlap patterns are captured with much less repetition of patterns across the clusters.

Since the templates are now rotation normalized, they are matched at all orientations. Therefore, p_i is updated to: $p_i = (x_i, y_i, \theta_i)$, where θ_i is the orientation of part i. Also, $\psi(p_i - p_j) = [-dx \;\; -dx^2 \;\; -dy \;\; -dy^2 \; cos(d\theta)]$, with $dx = (x_i - x_j - \mu x_{i,j}^{t_i,\theta_i})$. $\mu x_{i,j}^{t_i,\theta_i}$ is calculated as $\mu x_{i,j}^{t_i,\theta_i} = \mu r_{i,j}^{t_i} cos(\theta_i)$, where $\mu r_{i,j}^{t_i}$ is the average distance between the part i and its parent j in the training images with type t_i for part i. During inference, $(\mathbf{p}^*, \mathbf{t}^*) = argmax \; S(I, \mathbf{p}, \mathbf{t}; \mathbf{w}, \mathbf{b})$ is calculated. Learning is performed using (2).

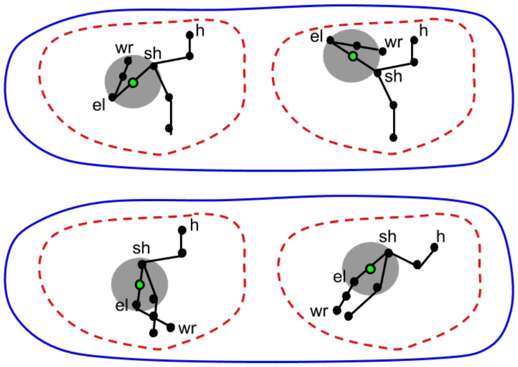


Figure 6. The red dashed lines depict the Phraselet clustering [2] for the upper-arm, while the blue solid lines depict the Rotation Normalized Phraselet clustering.

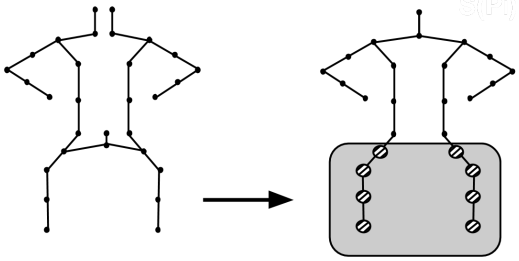


Figure 7. Two Trees pose estimation.

4. Localization of Non-adjacent Symmetric Parts

Often, predicted locations of the symmetric parts overlap with one of the parts if they are not kinematically constrained in the part-based model. For example, in Fig 1(c), the leg predictions from the tree of Fig 4(a) overlap with the same leg due to the absence of a constraint between them. However, enforcing the constraints between both the shoulders as well as the hips would introduce a cycle in the tree, inhibiting efficient maximization using Dynamic Programming [4].

Towards unambiguous localization of the symmetric lower limbs while still exploiting the advantage of tree structured interactions, we define a new tree as shown in Fig 4(b). In this tree, the connection between the left and right side happens in the lower body and not in the upper body. Specifically, two nodes are added around the pelvis region. This connection constrains the left and the right lower limbs kinematically and avoids the overlap of leg predictions. Furthermore, it strengthens the localization of the lower body parts by collecting additional image evidence around the pelvis region. We refer to this tree as the *Lower-constrained Tree* and the traditionally used tree (shown in Fig 4(a)) as the *Upper-constrained Tree*.

A missing constraint in the upper part of the *Lower-constrained Tree* may produce erroneous upper body pose estimates, consequently affecting the lower body pose estimation as well. We observe that the head has a distinctive appearance and is localized reliably. Therefore, in order to mimic the constraints in the upper body, we introduce two head nodes in the *Lower-constrained Tree* as shown in Fig 4(b). This is especially effective when there is less ambiguity in the head location. In such cases, the *Lower-constrained Tree* outperforms the *Upper-constrained Tree*, even for the upper parts; the correct localization of the lower parts along with unambiguous head location helps in the localization of the upper parts as well. However, in general, the *Lower-constrained Tree* still underperforms on the upper parts. Due to this, we combine the two complementing trees in order to obtain superior pose estimates for both the upper and the lower parts.

Our method of combining the two trees is based on the following idea: suppose we know the true locations of the lower body parts in an image. Now, if we infer the upper body pose using the *Upper-constrained Tree* with the lower parts fixed to their true locations, the estimates of the upper body parts would improve. However, in practice, we do not know the true locations. Instead, if we use a different model that performs better lower body pose estimation, the overall accuracy increases.

Two Trees Pose Estimation (Fig 7): First, the pose is estimated using the *Lower-constrained Tree*. Treating the predicted pose of just the lower-body parts as the input evidence, the upper body pose is again estimated using the *Upper-constrained Tree*.

Similar combination can be performed by embedding the upper body pose predicted using the *Upper-constrained Tree* into the *Lower-constrained Tree*. However, due to the constraint in the lower part and the two heads nodes mimicking the constrains in the upper part, the *Lower-constrained Tree* generally localizes the parts more accurately. Therefore, starting with the *Upper-constrained Tree* and re-estimating with the *Lower-constrained Tree* does not generally perform better than the opposite way. The *Upper-constrained Tree* and the *Lower-constrained Tree* are trained independently using (2).

Note that Wang and Mori [25] also use multiple complementing trees to handle the non-adjacent part interactions. However, the difference is mainly in combining the trees. While they combine the distribution over poses returned by the trees using boosting, we combine the max-scoring poses returned by the trees using our Two Trees Pose Estimation.

5. Experimental Evaluation

5.1. Setup

5.1.1 Datasets

We use three datasets, namely the Leeds Sports (LSP) dataset [10], the Image Parse (IP) dataset [15] and a Dance dataset created by us. All these datasets have challenging poses with complex part interactions. Hence they are suitable to evaluate our ideas.

The LSP has 1000 images for training and 1000 images for testing. It contains images of people involved in sports like football, gymnastics, tennis etc and is particularly challenging in terms of pose variations. The annotations contain pixel locations of 14 keypoints (head, neck, shoulders, elbows, wrists, hips, ankles and knees). The IP has 100 images for training and 205 images for testing. The nature of images and the annotated keypoints are similar to that of LSP.

The Dance dataset has 150 images for training and 72 images for testing. The images are selected from the top searches returned by Google for the keywords ”Dance Solo“, ”Hip hop poses“ and ”Modern dance poses“. Similar to LSP, 14 keypoints are annotated for each image.

5.1.2 Evaluation Measure

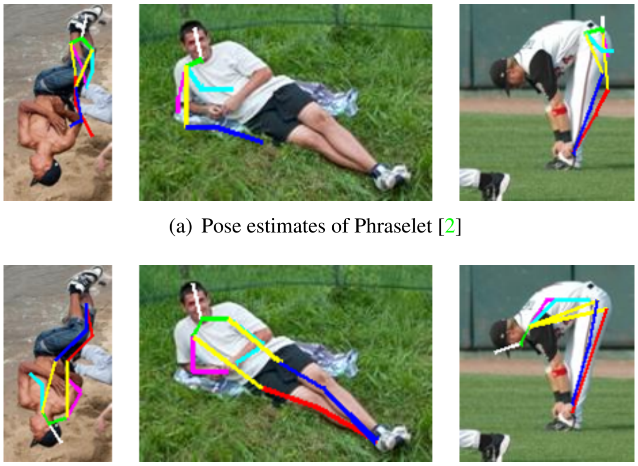
We use the standard Percentage of Detected Joints (PDJ) [16, 22] as our evaluation measure. According to PDJ, a joint is correctly detected if the distance between the predicted joint location and the groundtruth location is within some fraction of the torso diameter (distance between the left shoulder and the right hip). In a PDJ curve, this fraction is varied between 0 and 0.5 and the percentage of detected joints are plotted for each value of this fraction [22]. PDJ_{avg} is used to represent the average percentage of detected joints over the whole curve.

5.1.3 Implementation Details

For all our experiments, we use 7 part-types and 36 orientations. For fairness in comparison, we retrain the MoP [27] and the Phraselets [2] with 7 part-types as well. With most part of the code in matlab, our full model (Rotation Normalized Phraselets + Two Trees Pose Estimation) takes about 15 seconds using 8 parallel threads and about 60 seconds using a single thread on a typical image in the LSP dataset.

5.2. Results and Discussion

In this section, we first analyze the two orthogonal solutions that we propose. Then, we combine both the orthogonal solutions and compare our results with the state-of-the art.



(a) Pose estimates of Phraselet [2]

(b) Pose estimates of Rotation Normalized Phraselet

Figure 8. Qualitative analysis of Rotation Normalized Phraselet.

Category		Elb	Wri	Kne	Ank
Sports	MoP [27]	38.4	30.8	46.2	43.2
	Phraselets [2]	40.2	30.6	47.8	45.2
	RotNorm Phr (Ours)	**48.3**	**37.5**	**51.8**	**47**
Gym	MoP [27]	15.2	12.1	14.6	15.8
	Phraselets [2]	16	10.2	14.9	16.4
	RotNorm Phr (Ours)	**22.3**	**18.4**	**21.6**	**18.4**
Dance	MoP [27]	52.9	43.9	48.3	39.7
	Phraselets [2]	52.4	44.8	45.6	39.1
	RotNorm Phr (Ours)	**67.4**	**61.8**	**57.8**	**49.6**

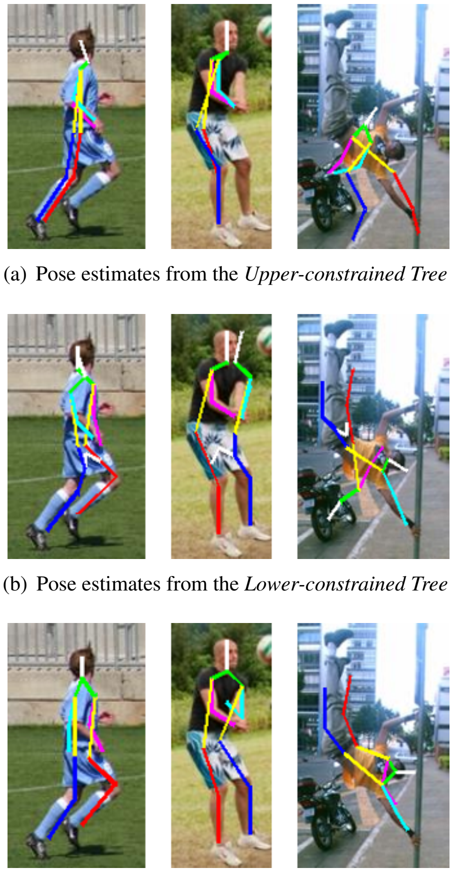
Table 1. PDJ_{avg} values for various categories of images.

5.2.1 Rotation Normalized Phraselets

Recall that Rotation Normalized Phraselets capture the appearances of overlap patterns around a part normalized according to the part's orientation while Phraselets [2] capture the appearances of the unnormalized overlap patterns. Due to limited training data, the overlap patterns generally do not repeat enough number of times at multiple orientations such that a template can be learned for each orientation. Fig 8 shows some cases where Phraselets fail to handle the non-upright self-occlusion patterns while the Rotation Normalized Phraselets are able to estimate the pose correctly.

For quantitative evaluation of Rotation Normalized Phraselets, we form three categories of images, namely Sports, Gym and Dance. The Sports category contains all the images from the LSP test set. The Gym category is formed by manually selecting the images from the LSP test set if the activity in the image is gymnastics or aerobics. There are 129 images in the Gym category. All the images in the Dance test set are placed in the Dance category.

We evaluate three algorithms, namely the basic Mixture of Parts [27] (MoP) model, Phraselets [2] and our Rotation Normalized Phraselets on all the image categories. For evaluation on Sports and Gym categories, models are trained on the LSP training set, while for evaluation on Dance category, models are trained on the Dance training set. The PDJ_{avg} values of elbow, wrist, knee and ankle are reported in Table 1. It can be seen that Phraselets modestly improves over MoP while Rotation Normalized Phraselets obtain large improvement on all the parts. The gain in performance is especially significant on more challenging categories such as Gym and Dance.


(a) Pose estimates from the *Upper-constrained Tree*

(b) Pose estimates from the *Lower-constrained Tree*

(c) Two Trees Pose Estimation

Figure 9. Qualitative analysis of Rotation Normalized Phraselet.

5.2.2 Two trees Pose Estimation

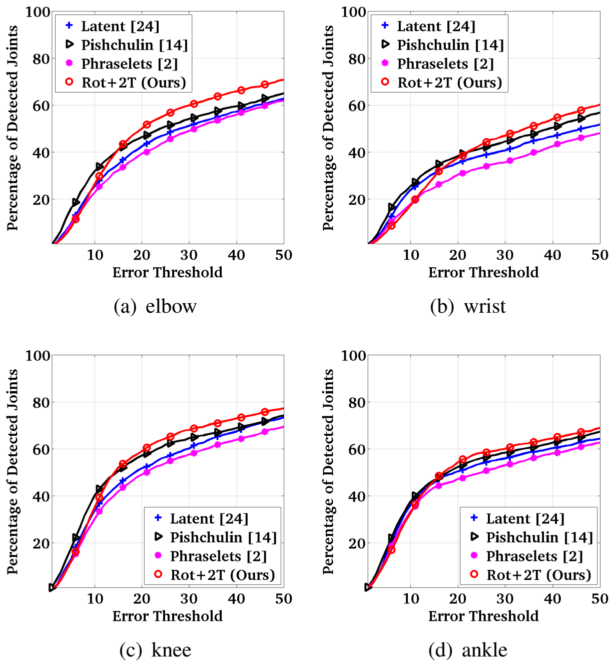
Fig 9 shows some pose estimates from the *Upper-constrained Tree*, the *Lower-constrained Tree* and the combination of the two trees. In the first and the second images, head appearance is distinctive. Therefore, the two head nodes in the *Lower-constrained Tree* (Fig 4(b)) localize accurately, effectively mimicking the constrains in the upper body. This, in addition to the constraints in the lower body causes the *Lower-constrained Tree* to localize both the upper and lower parts correctly by itself. The second image also shows how an erroneous lower body pose estimation in the *Upper-constrained Tree* (Fig 4(a)) can cause an error in the upper body pose estimation as well. The third image shows an example where the *Upper-constrained Tree* predicts an inaccurate pose, the *Lower-constrained Tree* predicts a partially accurate pose but the combination predicts a fully accurate pose.

Meth.		Sho	Elb	Wri	Kne	Ank
MoP [27]	Up T	51.8	38.4	30.7	46.2	43.1
	Lo T	**54**	**40.3**	**32**	49.7	46.5
	2T	53.3	39.3	31.5	**49.7**	**46.5**
Phr [2]	Up T	56.2	40.2	30.6	47.8	45.2
	Lo T	58	**42.2**	**33.5**	52.1	48.7
	2T	**58.3**	41.7	31.7	**52.1**	**48.7**
RotNorm Phr (Ours)	Up T	**60.3**	**48.3**	37.5	51.8	47
	Lo T	55	44.9	36	55.4	50
	2T	60.1	48.1	**37.9**	**55.4**	**50**

Table 2. Two Trees Pose Estimation Analysis on the LSP dataset [10]. Up T: *Upper-constrained Tree*, Lo T: *Lower-constrained Tree*, 2T: Two Trees Pose Estimation.

Table 2 reports the PDJ_{avg} values of various joints for the *Upper-constrained Tree*, the *Lower-constrained Tree* and the two trees for the three algorithms on the LSP dataset. First, we point out that the *Lower-Constrained Tree* consistently improves the localization accuracy of the lower parts. Further, the Two Trees Pose Estimation shows superior localization accuracies for both the upper and the lower body parts. Note that this is achieved by only doubling the running time.

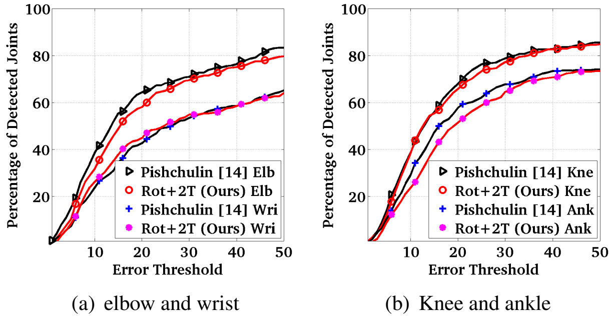
An interesting observation is that in algorithms where there is no rotation normalization, such as MoP [27] and Phraselets [2], the *Lower-Constrained Tree* performs better on all the parts in comparison with the *Upper-Constrained Tree*. This is because, when there is no rotation normalization, the templates are searched only over translations and scales. Due to the reduced search space, the ambiguity in head location reduces. Therefore, both the head nodes in the *Lower-constrained Tree* localize accurately just based on the appearance, leading to a better localization of the upper body parts as well. This observation is very useful for the algorithms without rotation normalization, since pose estimation on the *Lower-constrained Tree* takes the same time as the traditionally used tree (the *Upper-Constrained Tree*), yet being more accurate.



(a) elbow (b) wrist

(c) knee (d) ankle

Figure 10. PDJ curves on the LSP dataset [10] (best viewed in color).



(a) elbow and wrist (b) Knee and ankle

Figure 11. PDJ curves on the Image Parse dataset [15] (best viewed in color).

5.2.3 Comparisons with the Other Methods

We note that our best performing model is a combination of the Rotation Normalized Phraselets and the Two Trees Pose Estimation reported in the last row of Table 2. We compare this model on two datasets, namely the LSP [10] and the IP [15], with the most recent methods that use HOG features. Comparative PDJ curves on the LSP dataset are shown in Fig 10. It can be seen that we beat the baseline (Desai and Ramanan [2]) by a large margin. Also, no method out-performs all the methods and we perform equivalently to the best performing methods.

Our performance is very similar to that of the model of Pishchulin *et al*. [14]. They combine the best practices in the Human Pose Estimation problem and show that such a combination can produce very good results. However, their method uses many complex DPM templates at various granularities to perform accurate localization. Due to this, the running time of their released code on a typical image in the LSP test set increases to about 7 minutes. On the other hand, we use simple part-level HOG templates and achieve a similar performance with a running time of about 15 seconds on the same images. Moreover, the specialized detectors of Pishchulin *et al*. [14] can be used with our model as well to boost our performance, albeit at the cost of increased running time.

We also compare our method on the IP dataset. The PDJ curves are shown in Fig 11. It can be seen that our performance is similar to that of Pishchulin *et al*. [14], albeit again at a much lower computational cost.

6. Conclusion

Tree structured pair-wise constraints are restrictive in terms of encoding all the possible part interactions. Two strong manifestations of such unhandled part interactions are self-occlusion among the parts and a confusion in the localization of the non-adjacent symmetric limbs. We propose two modular and efficient improvements to Desai and Ramanan's [2] method to address the above problems. First, we propose Rotation Normalized Phraselets for handling self-occlusion in a more data efficient manner. We show especially large improvements on uncommon poses such as sports, gymnastics and dance. Secondly, we propose a solution for handling the confusion in the localization of non-adjacent symmetric limbs using a combination of two complementing trees and report a boost in performance taking only twice the time. We also show that a combination of the above two solutions improves the results compared to either used alone. We evaluate our method on two standard datasets and achieve equivalent results to the best performing methods in much less time when compared to the state-of-the-art part based model.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 2

[2] C. Desai and D. Ramanan. Detecting actions, poses, and objects with relational phraselets. In *Computer Vision–ECCV 2012*, pages 158–172. Springer, 2012. 2, 3, 4, 5, 6, 7, 8

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010. 1, 2

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005. 1, 2, 3, 4, 5

[5] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Comput.*, 22(1):67–92, Jan. 1973. 1

[6] G. Ghiasi, Y. Yang, D. Ramanan, and C. C. Fowlkes. Parsing occluded people. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2401–2408. IEEE, 2014. 2

[7] G. Gkioxari, P. Arbeláez, L. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3342–3349. IEEE, 2013. 2

[8] A. Gupta, A. Mittal, and L. S. Davis. Constraint integration for efficient multiview pose estimation with self-occlusions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(3):493–506, 2008. 1

[9] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler. Learning human pose estimation features with convolutional networks. *arXiv preprint arXiv:1312.7302*, 2013. 2

[10] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12. 2, 6, 7, 8

[11] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1465–1472. IEEE, 2011. 2

[12] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 1

[13] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 588–595. IEEE, 2013. 2

[14] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3487–3494. IEEE, 2013. 2, 8

[15] D. Ramanan. Learning to parse images of articulated bodies. In *Advances in neural information processing systems*, pages 1129–1136, 2006. 2, 6, 8

[16] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3674–3681. IEEE, 2013. 2, 6

[17] L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2041–2048. IEEE, 2006. 1

[18] T.-P. Tian and S. Sclaroff. Fast globally optimal 2d human detection with loopy graph models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 81–88. IEEE, 2010. 1

[19] Y. Tian, C. L. Zitnick, and S. G. Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In *Computer Vision–ECCV 2012*, pages 256–269. Springer, 2012. 2

[20] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. *arXiv preprint arXiv:1411.4280*, 2014. 2

[21] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2014. 2

[22] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. *arXiv preprint arXiv:1312.4659*, 2013. 6

[23] D. Tran and D. Forsyth. Improved human parsing with a full relational model. In *Computer Vision–ECCV 2010*, pages 227–240. Springer, 2010. 1

[24] F. Wang and Y. Li. Beyond physical connections: Tree models in human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 596–603. IEEE, 2013. 2

[25] Y. Wang and G. Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *Computer Vision–ECCV 2008*, pages 710–724. Springer, 2008. 5

[26] Y. Wang, D. Tran, and Z. Liao. Learning hierarchical poselets for human parsing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1705–1712. IEEE, 2011. 2

[27] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011. 1, 2, 3, 4, 6, 7

[28] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012. 2