Data Decomposition and Spatial Mixture Modeling for Part based Model

Junge Zhang, Yongzhen Huang, Kaiqi Huang, Zifeng Wu and Tieniu Tan

National Lab of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
Email: {*jgzhang, yzhuang, kqhuang, zfwu,tnt*}*@nlpr.ia.ac.cn*

Abstract. This paper presents a system of data decomposition and spatial mixture modeling for part based models. Recently, many enhanced part based models (with *e.g.,* multiple features, more components or parts) have been proposed. Nevertheless, those enhanced models bring high computation cost together with the risk of over-fitting. To tackle this problem, we propose a data decomposition method for part based models which not only accelerates training and testing process but also improves the performance on average. Besides, the original part based model uses a strict rigid structural model to describe the distribution of each part location. It is not "deformable" enough, especially for those instances with different viewpoints or poses in the same aspect ratio. To address this problem, we present a novel spatial mixture modeling method. The spatial mixture embedded model is then integrated into the proposed data decomposition framework. We evaluate our system on the challenging PASCAL VOC2007 and PASCAL VOC2010 datasets, demonstrating the state-of-the-art performance compared with other related methods in terms of accuracy and efficiency.

1 Introduction

Part based models have been a successful method for representing object categories [1–6]. It was firstly proposed by Fischler and Elschlager [7] in 1973. Later in [8] Marr and Nishihara introduced articulated limb model. In the past several years, Felzenszwalb *et al*'s work [1, 9] significantly advances the original pictorial structure model [7]. Part based models have been widely used in several important computer vision problems such as object detection [1–4, 6, 10, 11], pose estimation [5], action recognition [12] and scene understanding [13].

Part based models consider that an object can be modeled as a collection of local part templates, together with structural constraints. In the past decade, constellation model proposed by Fergus *et al* [11] and pictorial structure model presented by Felzenszwalb *et al* [1, 9] obtained great success. The latter deformable part based model (DPBM) [1] stands out for its outstanding performance on VOC challenges [14]. The use of moving parts can well adapt the learnt model to target image structure. For detection task, this kind of configuration has good property of robustness to deformation and partial occlusion which

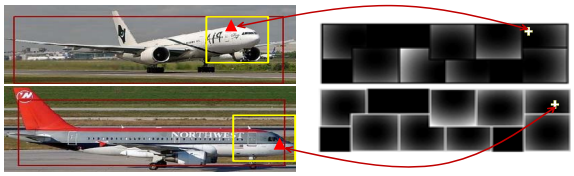

Fig. 1: Examples of different viewpoint but the same aspect ratio. The layout of structural constraints should be different.

provides superior performance than rigid template model [15]. However, to our understanding, part based models have two basic limits: 1) the computational complexity is high. 2) The original DPBM is not "deformable" enough.

Recently, there surge many enhanced models through multiple features [3] such as combining Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP), more components and parts [5, 16]. These methods obtained very promising results on either detection task or pose estimation. Nevertheless, these models suffer from large computational complexity. Besides, when the length of models becomes longer, they face a higher risk of over-fitting.

The original DPBM [1] uses one unique reference anchor point for each part. This results in that the layout of structural constraints or penalty for each part is all the same and rigid when applying the same component model. We think that each anchor associates a layout of structural constraints. As discussed in [1], the size of each component model is initialized by objects' aspect ratio to avoid bad local minimas. But in practice, two objects with the same aspect ratio may have different viewpoints or poses. As seen from Figure 1, the two aeroplanes share the same aspect ratio but have apparently different viewpoints. In this case, it is inappropriate if we encourage the same layout of structural penalty to both of them.

Motivated by those challenges, this paper tries to address these two limits. Firstly, we propose a method of data decomposition for part based model which not only significantly reduces memory usage and computational cost but also outperforms other related systems. Secondly, we propose a spatial mixture modeling method in which part location is described as mixture distribution learnt from weakly labeled data. Thirdly, we integrate the spatial mixture model into the proposed data decomposition framework and to the best of our knowledge, the presented system achieves the state-of-the-art performance compared with all other related methods from both competition and literature.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the proposed method. Section 4 gives the experimental results. Section 5 concludes this paper.

2 Related work

As mentioned previously, many enhanced models with multiple features [3, 17], more components or parts [5, 16] are proposed recently. The drawback or limit of these methods is that their models' complexity is too high in terms of computing

and memory. Hussain *et al* [17] propose applying partial least squares (PLS) on three different types of visual features. The supervised PLS is performed on each separate root and part filter which requires carefully collecting training data for each root and part model. Besides, how to perform data alignment for such separate learning reductions is difficult. Moreover, it is infeasible for more flexible part models with more features, components and parts.

Patrick *et al* [16] adopt sharing parts across intra and inter categories to reduce the model's complexity. Although sharing parts can reduce the number of parameters, it may face the risk of decreased discrimination of parts. For example, we'd like a car's 'wheel' part has discrimination not only between a car and a person but also between a car and a bus. If the 'wheel' part is shared across car and bus category, its discrimination will certainly decrease when classifying a window as a car or a bus.

Pedersoli *et al* [18] show that the dimensionality of filter and window search space dominate most of the computation time. They propose a coarse-to-fine search strategy to speedup the complex hierarchical part based model. In [19], they propose a selective window sampling strategy via segmentation which makes using multiple features based bag-of-words possible. Those methods indeed reduce search space and speedup training and testing, but they all do not improve models' discrimination. The promising way to improve models' discrimination is building more discriminative appearance feature or modeling strategy.

Yang *et al* [5] extends the DPBM into flexible mixture of parts which is "deformable" enough for pose estimation. They use the relative location between parent node and children node to define the part type. Besides, in DPBM, only one large rigid part is used to describe *e.g.,* a 'leg'. But in [5], a 'leg' is represented by many small flexible rigid parts so that matching articulated pose becomes possible. However, this method requires fully annotated training data and carefully predefined part dependence or part order. Therefore, training a generic class model with weakly labeled training samples is difficult. Besides, compared with larger parts, the discriminability of smaller parts may be decreased.

3 Proposed methods

The proposed system is schematically shown in Figure 2. With the input of different features, we need to perform data calibration for further data decomposition. Then the learnt basis is applied to factorize the feature into lower-dimensional space. Besides, we consider each part's configuration as a spatial mixture model. The structural penalty for each part is more flexible. In the following paragraphs, we will describe the details of the proposed method.

3.1 Data decomposition

As mentioned previously, these enhanced models via multiple features, more components or parts always require huge computation resources. And they also

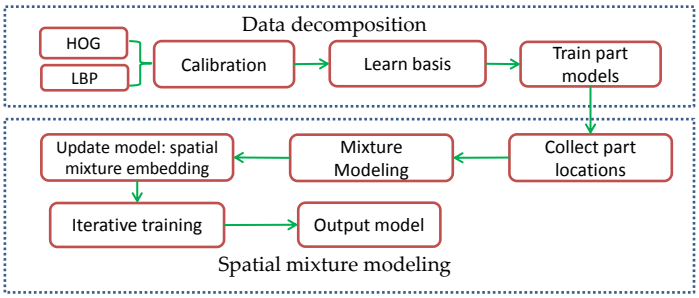

Fig. 2: The pipeline of the proposed system, including two parts: data decomposition and spatial mixture modeling.

face the higher risk of over-fitting. The first contribution of this paper is reducing part based models' complexity via data decomposition which enables efficient and sufficient training and testing.

Before details, the feature map used in DPBM should be introduced. Cell structured feature map [1] is computed at every scale. This kind of processing enables us to use fast convolution routine for matching. As known that cell filter is the basic unit of either root filter or part filter, therefore, we can perform data decomposition on cell filter.

We know that some cell filters refer to background, while others to objects. Besides, the size of cell filter is usually so small that each cell can not hold sufficient appearance information to represent background or objects. Therefore, the decomposition method should be **unsupervised** or **label independent**. The original feature data (background or objects, and this is why we call our method as data decomposition) can be reconstructed without supervision. This is the basic principle we should follow. Another principle is the decomposition should be **efficient** due to large scale application.

For unsupervised methods, there are *e.g.,* principal components analysis (PCA) and non-negative matrix factorization (NMF). Considering efficiency, PCA rather than NMF is an ideal choice because it benefits from its linear projection. Therefore, we propose using PCA to perform data decomposition. This paper mainly focuses on reducing the complexity of those enhanced models. PCA is already a very useful method to address it. It should be mentioned that this paper considers both training and testing model entirely in the context of multiple features which is different from [1, 20]. [20] considers testing procedure only, while in [1] PCA is used to analyze the basic 36-dim HOG feature and finally they adopt the manually reduced 31-dim HOG feature based on their analysis. Thus, the context and usage of PCA is different.

The problem of PCA is that its results heavily rely on the relative scaling of the original data. If two variables have different units (*e.g.,* kilometers and miles.), the results produced by PCA will be different. Therefore, it is necessary to do data calibration before learning the basis which is in fact very essential for the system.

Proposed data calibration

Motivated by the promising performance of [3], this paper adopts HOG from [1]

and LBP feature with uniform patterns as well. We use $X_1(i)$ and $X_2(j)$ (wherein $i \in [1, 31]$ and $j \in [1, 59]$) to denote HOG and LBP feature and η_1 and η_2 for their discriminative ability, respectively. Their relative discriminative ability is defined as: $\lambda = \frac{\eta_1}{\eta_2}$. The average accuracy evaluated on 20 categories from PASCAL VOC2007 reported in [21] is used as their respective discrimination. We don't consider a category dependent discrimination for generalization. The proposed data calibration includes two steps: 1) removing variables with low sample variance; 2) Re-scaling two different sources of data.

As we know, variables with low variance from the same source data indicate low contribution or even damage to discrimination. Therefore, we can remove those variables with very low variance. In [3, 21], the results show that the discrimination of HOG is superior over LBP on average. Motivated by these results, we only remove the variables from LBP. Suppose the variance of $X_2(j)$ is $var_2(j)$ and the 20% lowest variance (it is determined empirically.) is used as threshold $thresh$. Then we remove the variance according to Eq.1.

$$X_2(j) = \begin{cases} 0, & \mathrm{var}_2(j) \leq thresh \\ X_2(j), & if\ else \end{cases} \tag{1}$$

The second step is to re-scale the different source data. After removing the variables with low variance from LBP, the mean value of X_1 and X_2 is computed and denoted as u_1 and u_2, respectively. X_1 is chosen as the reference scale for its better discrimination. Then the relative discriminative ability λ is considered into re-scaling problem according to Eq.2.

$$X_2 = \frac{X_2}{\frac{u_2}{u_1} \cdot \lambda} \tag{2}$$

From the statistics, we find that u_2 is larger than u_1. Thus, we need to re-scale X_2 to smaller scale. Besides, the larger λ is, the less contribution that X_2 gives. Therefore, we divide X_2 by λ together with $\frac{u_2}{u_1}$ to re-scale the original data to an appropriate relative scale considering both relative mean value and discrimination.

Implementation details. Similar to [1], the models are trained horizontal symmetric. The question is how to find the corresponding relationship between mirrored features in decomposed space or subspace which is another key problem in our detector.

Suppose the extracted low-level feature from one side view image is f, then we can get its corresponding symmetric feature $f^{'}$. We use d, $d^{'}$ and g to denote their factorized feature in subspace and decomposition function respectively. Then, we get

$$\begin{aligned} d &= g(f) \\ d^{'} &= g\left(f^{'}\right) \end{aligned} \tag{3}$$

The simplest way of finding their corresponding relationship is regressing $d^{'}$ on d. Then it turns to find the transformation matrix L through $d' = g(f') = dL$. In this paper, we adopt PLS which is suitable for multivariate regression.

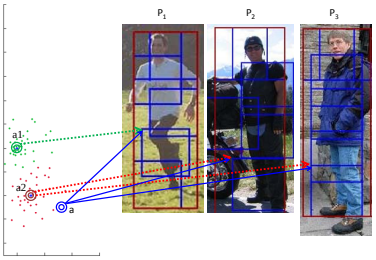

Fig. 3: A toy example for illustrating the motivation of spatial mixture modeling.

3.2 Spatial Mixture Modeling

We now consider working out the flexible part matching scheme. As we know, the response of each part p_i is simultaneously determined by appearance information and structural penalty.

$$h\left(p_i\right) = \max_{s_i \in \zeta} \left(h_a\left(s_i\right) - h_s\left(s_i\right)\right) \tag{4}$$

where h_a and h_s are the appearance and structural response respectively. Part's moving location $s_i : (x_i, y_i)$ (of dimensionality d) belongs to the searching space ζ. In the basic DPBM, the structural penalty is achieved by

$$h_s\left(s_i\right) = w_i{}' d\left(s_i\right), \quad where \;\; d\left(s_i\right) = \left(dx_i \; dx_i^2 \; dy_i \; dy_i^2\right) \tag{5}$$

where $w_i{}'$ and $d\left(s_i\right)$ are the deformation coefficients and structural features respectively. In the basic DPBM, only a single layout of structural constraint is used to describe part's distribution. If the viewpoint or pose within the same aspect ratio is the same, a single layout of structural constraints would be sufficient to model the part's spatial distribution. In practice, two objects with the same aspect ratio usually have apparently different viewpoints or poses (*e.g.*,Figure 1 and Figure 3). Therefore, a single layout of structural constraints can not capture such variation and spatial mixture modeling becomes necessary.

In this paper, we assume that the part spatial distribution follows mixture Gaussian distribution. Then we present the spatial mixture modeling based on Gaussian to capture the variations of parts. In this case, the system can make a more "flexible" decision for the structural penalty for each part. Naturally, we define the score of each part as:

$$h\left(p_i\right) = \max_{s_i \in \zeta} \left(h_a\left(s_i\right) - \sum_{j=1}^{K} \lambda_{i,j} h_{s,j}\left(s_i\right) \right) \tag{6}$$

where K is the number of Gaussian components and λ_j is the weight of structural penalty from the j^{th} component. In this case, the structural penalty is weighted accumulated based on the contribution from each Gaussian component. Therefore, when the relative location between the part current moving location and each Gaussian component changes, the layout of structural constraints for that

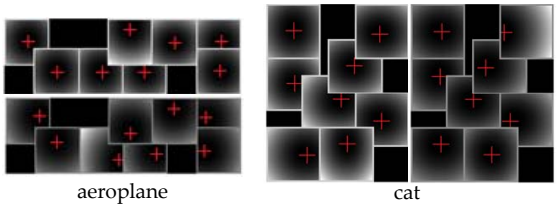

Fig. 4: Here are two examples of deformation model from spatial mixture modeling: the left one is about aeroplane and the right one is cat.

part will change accordingly which makes flexible part matching possible. λ_j is obtained through:

$$\begin{aligned} &\lambda_{i,j} = w_{i,j} g\left(s_i|\mu_j, \Sigma_j\right) \\ &s.t \sum_{j=1}^{K} w_{i,j} = 1 \; and \; w_{i,j} \geq 0 \end{aligned} \tag{7}$$

where $w_{i,j}$ are the mixture weights, and $g\left(s_i|\mu_j, \Sigma_j\right)$ is the component Gaussian density. μ_j, Σ_j are the mean value and covariance matrix of the j^{th} Gaussian component in the mixture. Each Gaussian probability density is

$$g\left(s_i|\mu_j, \Sigma_j\right) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\left\{-\frac{1}{2}(s_i - \mu_j)^T \Sigma_j^{-1}\left(s_i - \mu_j\right)\right\} \tag{8}$$

The vector $\phi = (w, \mu, \Sigma)$ is the unknown parameters of the spatial mixture model which needs to be estimated.

There are various algorithms for estimating the parameters of ϕ. A popular method for maximizing the likelihood of the training data is expectation-maximization (EM). The basic idea of EM algorithm is beginning with initial parameters ϕ, to evaluate the new parameters ϕ', for which we hope the likelihood is larger. The new parameters then become the initial model for the next iteration and the process is repeated until converged. In our system, we use K-means to generate the initial parameters.

Implementation details. Our spatial mixture embedded model starts from training a basic DPBM. Then we collect the part location from those positive samples of 70% overlap with ground truth. After that K-means is applied over each part's location. The mean value and covariance matrix are generated from each cluster. The initial weight for each Gaussian component is determined by the fraction of the number of points in each cluster. With the initial parameters, EM algorithm is executed to find the proper parameters ϕ. The learnt μ is then used as the new reference anchor points in the spatial mixture embedded part based model. We use the model with spatial mixture modeling to mine positive and negative training samples for further iteratively training. The score of each part location is determined by Eq.6. The iterative training process terminates at a certain iterations or when the model changes little. The full algorithm is summarized in Algorithm 1.

Discussions. The spatial mixture model utilizes mixture layouts of structural constraints and provides a flexible part matching scheme which can well address the problem caused by variations in viewpoint or pose. The superiority of

Algorithm 1: Training Spatial Mixture Embedded Model.

Input : Positive/Negative samples
Output: *model*

1: Train basic DPBM. N: the number of components;
for *component* $i \leftarrow 1$ **to** N **do**
  for $k \leftarrow 1$ **to** $iter1$ **do**
    Train root model: $model\{i\}.root$;
  for $k \leftarrow 1$ **to** $iter2$ **do**
    Train part based model: $model\{i\}$

2: Initialize spatial mixture embedded model;
for *component* $i \leftarrow 1$ **to** N **do**
  Apply $model\{i\}$ on positive training samples;
  Collect part locations into $locs_{i,p}$;
  `// p is the subscript for each part`
  Clustering, generate the initial parameter $\phi_{i,p} = (w_{i,p}, \mu_{i,p}, \Sigma_{i,p})$ for each cluster;
  Estimate the parameter $\phi'_{i,p} = \left(w'_{i,p}, \mu'_{i,p}, \Sigma'_{i,p}\right)$ of mixture model with EM and $\phi_{i,p}$;
  Initialize spatial mixture model with $\phi'_{i,p}$;

3: Update model and retrain;
for *component* $i \leftarrow 1$ **to** N **do**
  for $i \leftarrow 1$ **to** $iter3$ **do**
    Apply updated $model\{i\}$ for collecting training data;
    Part response is determined according to Eq. 6;
    Update parameters and retrain $model\{i\}$.

our method can be graphically demonstrated by Figure 3. The locations of the right bottom part (foot part) collected from positive samples are plotted in the left image in Figure 3. The blue point denotes the anchor point in the original DPBM. Apparently, the original DPBM will punish the foot part in P_2 and P_3 slightly, while punishing that part in P_1 heavily which in fact is not desired for that pose. In our system, based on the proposed spatial mixture model, we can match each part more "deformably" or flexibly with mixture layouts of structural penalty (Eq. 6) rather than the original DPBM. Therefore the structural penalty from spatial mixture model can relieve the penalty for the foot part in P_1 while still retain slight punishment for that part in P_2 and P_3. In a word, the proposed method is capable of capturing variations in viewpoint or poses by allowing more flexible part matching. Figure 4 gives an example of learnt deformation model with spatial mixtures. The red cross in each part denotes the learnt anchor for that part. An apparent relative displacement of anchors associated with the same part can be found in Figure 4.

4 Experiments

We evaluate the proposed method on challenging PASCAL VOC dataset [14] which is widely recognized as difficult testbed for object detection and most algorithms report their results on this dataset. We use Average Precision (AP) [14] score as the criterion, which is widely adopted in PASCAL VOC challenge. As mentioned in 3.1, HOG and LBP are used as the low-level features. All the models are trained with six components, and each component associates eight parts of cell size 6×6. The experiments are mainly divided into three subsections: 1) empirical results with data decomposition; 2) experimental results with spatial mixture embedded model; 3) full results on PASCAL VOC2007 and VOC2010 datasets.

4.1 Data Decomposition

In this subsection, the experiments include three parts: 1) determining the factorized lower dimensionality; 2) studying the effect of data calibration and 3) training and testing computational cost and accuracy. These experiments on PASCAL VOC2007 are designed for verifying the effectiveness of the proposed methods, hence we only conduct the experiments on several categories. The final complete results will be given in 4.3.

Determining dimensionality. Table 1 illustrates how changing PCA dimension K affects the performance. As seen from Table 1, when K exceeds 30, the performance tends to be stable. Considering the trade-off between efficiency and effectiveness, we set K to 40 which is found performing well on all 20 categories (Note: Before applying PCA, data calibration is performed).

category	K=15	K=20	K=25	K=30	K=35	K=40	K=50
aeroplane	1.7	14.4	16.7	33.2	34.5	34.4	32.6

Table 1: This table shows how changing PCA dimension K affects the results.

Effect of data calibration. The experimental results are shown in Figure 5. Three groups of experiments are conducted: one is the naïve combination which refers to the method concatenating different types of features into a unified feature vector; The second is that we directly perform data decomposition on original data without any calibration; The third is the proposed method. As seen from Figure 5, the results of directly applying PCA without any calibration are usually bad. Because the dimensionality and mean value of LBP feature are larger than HOG, the learnt basis from PCA often has bias to LBP which in fact shows poorer discriminability on average than HOG. The results from the `complete` experiments on `aeroplane` and `bottle` verify the positive effect of data calibration via improving `noprocess` by 5% and 2.2%, respectively.

Training and testing computation cost and accuracy. Table 2 summarizes the quantitive results. All these experiments are conducted on the same computer with the same configuration. The computer is configured with Intel E5520 CPU of 2.27GHz. The training time for `cow` is all most the same. But the

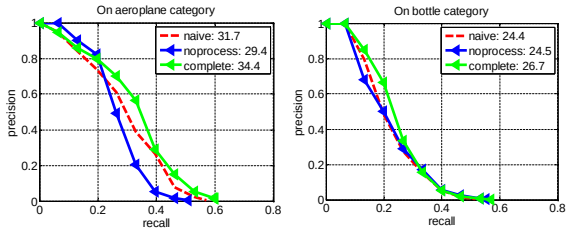

Fig. 5: Experimental results of data calibration. `naive` refers to the naïve combination of HOG and LBP. `noprocess` denotes the method that perform data decomposition without any calibration. `complete` represents the proposed data decomposition with data calibration.

	Class	Training (hour)	Testing (hour)	AP score
Baseline [1, 22]	cow	-	1.4	25.2
	cat	-	1.3	19.3
Naïve	cow	18.1	3.7	28.1
	cat	24.1	3.5	23.3
Boost[3]	cow	17.9	3.5	26.9
	cat	23.2	3.4	24.2
DD	cow	18.0	**1.9**	**30.4**
	cat	18.3	**1.8**	**24.6**

Table 2: Training and testing computation cost and accuracy comparison experiments. `Baseline` refers the result from running the provided models [1, 22] in the same environment. `Naïve` stands for naïve combination method. `Boost` refers the method described in [3]. `DD` denotes the proposed system without spatial mixture modeling.

proposed method achieves a speedup, of more than 1.9 times than naïve combination during evaluation. For the `cat` category, the training time is less than naïve combination, and speedup in evaluation stage is nearly two-fold. The naïve combination requires $O(90)$ (31+59=90) operations at each cell filter while the proposed method requires only $O(40)$ operations. The practical speedup factor is about 1.8 ($1.8 < \frac{90}{40} = 2.25$) which indicates the decomposition costs a bit time but is still very efficient especially for multiple features. The memory consumption is also reduced from original 10G byte to now 4G byte on average during training. We also implement a naïve version of the `Boost` method according to their description in [3]. We find that the training and testing computational cost is almost the same with naïve combination. Moreover, the proposed method achieves better performance on these two categories than [3]. The improvement over baseline method [1] proves that combining texture feature indeed helps discriminability which has been verified in [3] as well. The improvement over the naïve combination and [3] verifies that data decomposition over the original multiple features can still improve models' discriminability. The reason may be the presented data decomposition suppresses undesired noise and the reduced complexity makes the model can be trained more sufficiently.

4.2 Spatial Mixture Modeling

We take a "data-driven" approach to determine the number of mixture components K by analyzing the parts' spatial distribution. Limited by space, we plot only two parts distribution in Figure 6 randomly chosen from `person` and `chair`

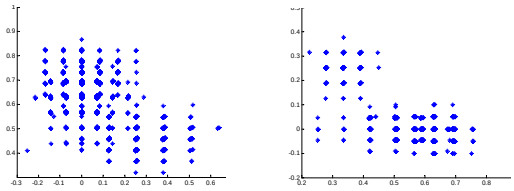

Fig. 6: Part spatial distribution. The left one is from `person`'s middle part, and the right one is from `chair`.

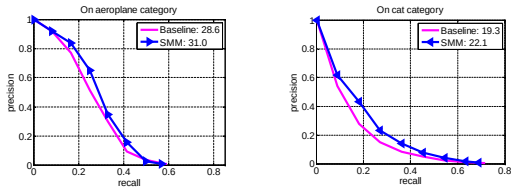

Fig. 7: Experimental results of spatial mixture modeling. Baseline is the result from running the provided model [1, 22] on PASCAL VOC2007.

categories. We can find that the number of peaks is almost 2. Besides, if K is larger, the parameters of the model are more such that it will plague sufficiently training model and efficient testing. Therefore, we set K to 2 generally for all categories on VOC datasets. In this subsection's experiments, we use HOG from [1] without data decomposition. The baseline is the standard DPBM [1, 22]. As seen from Figure 7, the proposed spatial mixture embedded model improves the baseline by 2.4% and 2.8%, respectively. The improvement verifies the effectiveness of the proposed spatial mixture embedded model which provides flexible and more "deformable" part configuration.

4.3 Complete results on PASCAL VOC datasets

Motivated by the above results, we integrate the proposed spatial mixture embedded model into the data decomposition framework and evaluate the whole system on PASCAL VOC2007 and VOC2010.

Results on PASCAL VOC2007. Table 3 gives the results of our detector on PASCAL VOC2007. The results here are without specific context based post-processing. We compare our method with other related representative methods.

As shown in Table 3, `DDSSM` (the proposed complete system) obtains the best AP score in 7 of 20 categories. The mean AP score is 36.0%, which is the best among these compared methods. `DDNoSSM` (the proposed method without spatial mixture modeling.) also obtains best score in 3 categories, and the mean AP is the second best among these methods. The closest approach to us is from [3], our method without spatial mixture modeling exceeds it by 0.9%. The whole system improves [3] by 1.7%. The improvement of `DDSSM` over `DDNoSSM` proves that the proposed spatial mixture modeling improves models' discriminability on average. The improvement is promising because the result from [3] is already very challenging. Besides, the proposed system shows advantage over [3] in terms of efficiency. This is currently the state-of-the-art performance without context

	plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	meanAP
V4 [22, 1]	28.9	59.5	10.0	15.2	25.5	49.6	57.9	19.3	22.4	25.2	23.3	11.1	56.8	48.7	41.9	12.2	17.8	33.6	45.1	41.6	32.3
UCI [23]	28.8	56.2	3.2	14.2	29.4	38.7	48.7	12.4	16.0	17.7	24.0	11.7	45.0	39.4	35.5	**15.2**	16.1	20.1	34.2	35.4	27.1
LHS [4]	29.4	55.8	9.4	14.3	28.6	44.0	51.3	21.3	20.0	19.3	25.2	12.5	50.4	38.4	36.6	15.1	19.7	25.1	36.8	39.3	29.6
MKL [24]	**37.6**	47.8	**15.3**	15.3	21.9	**50.7**	50.6	**30.0**	17.3	33.0	22.5	**21.5**	51.2	45.5	23.3	12.4	23.9	28.5	45.3	**48.5**	32.1
LatentCRF [2]	31.9	57.0	9.1	15.2	26.0	42.7	49.3	14.5	15.2	18.5	24.2	11.8	49.1	41.9	35.7	14.5	18.9	23.3	34.3	41.3	28.7
C2F [18]	27.7	54.0	6.6	15.1	14.8	44.2	47.3	14.6	12.5	22.0	24.2	12.0	52.0	42.0	26.8	10.6	22.9	18.8	35.3	31.1	26.7
SMC [25]	26.0	56.0	10.0	11.0	21.0	47.0	50.0	16.0	19.0	23.0	20.0	12.0	51.0	45.0	37.0	12.0	17.0	29.0	41.0	38.0	29.1
PLS [17]	18.0	41.1	9.2	9.8	24.9	34.9	39.6	11.0	15.5	16.5	11.0	6.2	30.1	33.7	26.7	14.0	14.1	15.6	20.6	33.6	21.3
ParAttr [26]	25.6	33.0	6.8	3.2	16.3	47.7	37.9	14.0	0.9	9.6	17.0	11.5	23.3	32.5	19.8	5.3	**29.9**	18.0	16.7	32.1	20.1
HStruct [27]	31.7	56.3	1.7	15.1	27.6	41.3	48.0	15.2	9.5	18.3	26.1	11.3	48.5	38.9	35.8	14.8	17.7	18.8	34.1	39.8	27.5
ExModel [28]	20.8	48.0	7.7	14.3	13.1	39.7	41.1	5.2	11.6	18.6	11.1	3.1	44.7	39.4	16.9	11.2	22.6	17.0	36.9	30.0	22.7
Boosted[3]	36.7	59.8	11.8	**17.5**	26.3	49.8	58.2	24.0	**22.9**	27.0	24.3	15.2	58.2	49.2	44.6	13.5	21.4	34.9	47.5	42.3	34.3
DDNoSSM	34.4	59.4	11.1	16.8	26.7	50.0	60.2	24.6	22.5	30.4	30.8	16.0	**61.3**	**51.3**	44.0	13.5	20.8	**39.2**	48.5	42.6	35.2
DDSSM	35.8	**60.4**	10.9	17.3	**29.9**	50.1	**62.6**	25.5	22.8	**38.2**	**32.1**	16.1	59.9	51.1	**44.8**	13.2	19.8	38.5	**49.5**	42.6	**36.0**

Table 3: Full results on PASCAL VOC 2007. DDNoSSM denotes the proposed method without spatial mixture modeling, while DDSSM with spatial mixture modeling. V4 is the popular DPBM proposed by Felzenszwalb *et al*[1, 22]. UCI refers to the method with multi-class layout [23] which wins Marr prize at ICCV2009. LHS stands for the method [4] which shows very competitive performance in recent years. MKL [24] is the winner method at PASCAL VOC2009 challenge, which uses four kinds of multi-level features. LatentCRF is from [2], in which a latent CRF based on a flexible assembly of parts is proposed for object detection. C2F represents the method described in [18]. This paper proposes a coarse-to-fine framework for deformable object detection. SMC denotes the scalable multi-class object detection [25], in which a shared codebook is jointly trained over all classes. PLS refers the method present in [17]. In ParAttr method [26], objects are described using a spatial model based on its constituent parts. HStruct represents the discriminative hierarchical structure model based on multiple features which is described in [27]. ExModel is an interesting method introduced recently in [28]. Boosted refers the winner method [3] in PASCAL VOC2010 challenge. Whether from the number of single class's best score or the mean AP, the proposed method performs best. It should be noted here the proposed method has not be filtered by context information.

	plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	meanAP
V4 [22, 1]	45.4	49.8	9.5	11.9	27.5	49.0	42.4	27.7	16.8	20.3	10.2	19.2	40.0	47.0	41.8	8.7	26.1	12.2	41.4	34.3	29.1
sharepart [16]	24.7	38.2	0.0	1.2	0.2	33.3	37.7	7.3	1.4	4.6	8.1	8.1	21.5	31.8	11.5	6.3	17.0	5.1	9.6	23.9	14.6
Boosted [3]	49.6	51.0	12.7	15.1	26.1	50.9	44.4	30.6	17.3	25.3	15.2	22.4	42.3	51.3	43.5	8.5	28.8	20.6	43.8	36.4	31.8
SegAs [19]	**58.2**	41.9	**19.2**	14.0	14.3	44.8	36.7	**48.8**	12.9	28.1	**28.7**	**39.4**	44.1	52.5	25.8	**14.1**	**38.8**	**34.2**	43.1	**42.6**	34.1
DDNoSSM	45.5	53.3	14.7	16.8	33.2	53.0	48.3	35.0	17.4	31.0	22.4	24.5	45.3	52.0	44.8	12.0	37.7	24.7	45.9	36.5	34.7
DDSSM	49.9	**54.9**	14.9	**17.0**	**33.6**	**53.6**	**50.6**	35.4	**18.1**	**31.4**	21.7	24.5	**45.8**	**52.6**	**49.2**	11.6	38.2	25.5	**47.4**	38.3	**35.7**

Table 4: Full results on PASCAL VOC2010. V4 is from [22] without context rescoring. sharepart refers the method described in [16]. In [16], certain parts are shared across mutli-class for multi-class object detection. Boosted [3] represents the winner method of PASCAL VOC2010. We run the detector implemented by ourself on PASCAL VOC2010. SegAs is the latest result from [19] which mainly focuses on selecting windows with high "objectness" via segmentation. Their object appearance model is based on bag-of-words. DDNoSSM refers the proposed method without spatial mixture modeling, while DDSSM with spatial mixture modeling.

rescoring and selective window search (e.g., [19]). MKL method with four different features also provides very competitive results, and our system gets better results by nearly 4%. And it is reported [24] that the MKL method takes roughly 67 seconds per image, therefore, the time for evaluating the whole VOC2007 is about 92 hours. Our method not only outperforms it in accuracy but also is very computational efficient. The proposed method takes about 2 hours for evaluation (as we run in different environment, the time here are only for rough comparison). We also noted that the additional experiment of [17] in his thesis [29] shows that they achieved 36.0% which is comparable to us. But our method does not need careful part calibration.

Results on PASCAL VOC2010. The complete results on PASCAL VOC2010 are given in Table 4. We compare with other four methods which published their results on PASCAL VOC2010. As seen from Table 4, DDSSM obtains the best score in 11 out of 20 categories and the best mean AP of 35.7% among all these methods. DDNoSSM also obtains the second best mean AP. SegAs [19] focuses on selective window sampling via segmentation. The proposed method exceeds [19] by 1.6% without any selective search. Also the selective search via segmentation is always time consuming. Compared with [3] in which low-level features used are similar to us, the proposed method without spatial mixture modeling improves [3] by nearly 3%, and over 5% improvement which is very challenging on PASCAL VOC datasets is observed on bottle, cow, diningtable and sheep categories. These results indicate that appropriate data decomposition over different sources of data for part based model not only reduces model's training and testing computational cost but also improves the accuracy on average. Besides, spatial mixture modeling improves DDNoSSM by about 1% on average, which further indicates the proposed spatial mixture modeling helps discriminability. In a

word, our system obtains the state-of-the-art performance compared with those methods without context rescoring on challenging PASCAL VOC datasets for detection task. Moreover, compared with other related challenging systems, the proposed algorithm requires less memory and computation time both in training and testing phase.

5 Conclusion

This paper has presented an enhanced part based model by means of data decomposition and spatial mixture modeling. We have made three major contributions: 1) We have studied the problem of complexity of those enhanced models and address this problem with data decomposition. In practice, we propose the methods for data calibration and finding transformation matrix which are very essential for the whole system. 2) We firstly build a more "deformable" and flexible part based model via spatial mixture modeling without fully annotated training samples. 3) The proposed data decomposition over multiple features for part based model not only reduces the computation requirement but also improves the accuracy and exceeds the previous state-of-the-art algorithms. The integrated system with data decomposition and spatial mixture modeling finally obtains the state-of-the-art performance on PASCAL VOC datasets compared with other methods without context.

Currently, the proposed system is still far from real time. On one hand, we can adopt the strategy such as [18, 19] to reduce the search space. On the other hand, we will continue to study the data decomposition for part based model following the principals described in this paper to reduce the model to a much lower dimensionality. Besides, our future work will also include learning part mixtures as well as the proposed spatial mixtures from weakly labeled data.

Acknowledgements This work is funded by National Natural Science Foundation of China (Grant No. 61175007), the National Key Technology R&D Program (Grant No. 2012BAH07B01), the National Basic Research Program of China (Grant No. 2012CB316302).

References

1. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI **32** (2010) 1627 –1645
2. Schnitzspan, P., Roth, S., Schiele, B.: Automatic discovery of meaningful object parts with latent crfs. In: CVPR. (2010) 121 –128
3. Zhang, J., Yu, Y., Huang, K., Tan, T.: Boosted Local Structured HOG-LBP for Object Localization. In: CVPR. (2011) 1393–1400
4. Zhu, L., Chen, Y., Yuille, A.L., Freeman, W.T.: Latent hierarchical structural learning for object detection. In: CVPR. (2010) 1062–1069
5. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. (In: CVPR) 1385 – 1392
6. Schnitzspan, P., Fritz, M., Roth, S., Schiele, B.: Discriminative structure learning of hierarchical representations for object detection. In: CVPR. (2009) 2238 –2245

7. Fischler, M., Elschlager, R.: The representation and matching of pictorial structures. Computers, IEEE Transactions on **C-22** (1973) 67 – 92
8. Marr, D., Nishihara, H.K.: Representation and recognition of the spatial organization of three-dimensional shapes. In: Proceedings of the Royal Society of London. Series B, Biological Sciences. (1978) 269–294
9. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. Int. J. Comput. Vision **61** (2005) 55–79
10. Girshick, R., Felzenszwalb, P., McAllester, D.: Object Detection with Grammar Models. In: NIPS. (2011)
11. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR. (2003) 264–271
12. Wang, Y., Mori, G.: Hidden part models for human action recognition: Probabilistic versus max margin. TPAMI **33** (2011) 1310 –1323
13. Pandey, M., Lazebnik, S.: Scene recognition and weakly supervised object localization with deformable part-based models. In: ICCV. (2011) 1307 – 1314
14. Mark, E., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal Visual Object Classes (VOC) Challenge. (In IJCV) 303–338
15. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005) 886–893
16. Ott, P., Everingham, M.: Shared parts for deformable part-based models. In: CVPR. (2011) 1513 – 1520
17. Hussain, S.u., Triggs, B.: Feature sets and dimensionality reduction for visual object detection, BMVA Press (2010) 112.1–112.10
18. Pedersoli, M., Vedaldi, A., Gonzalez, J.: A coarse-to-fine approach for fast deformable object detection. In: CVPR. (2011) 1353 – 1360
19. van de Sande, K.E.A., Uijlings, J.R.R., Gevers, T., Smeulders, A.W.M.: Segmentation as selective search for object recognition. In: ICCV. (2011) 1879 – 1886
20. Felzenszwalb, P.F., Girshick, R.B., Mcallester, D.: Cascade object detection with deformable part models. In: CVPR. (2010) 2241 – 2248
21. Zhang, J., Yu, Y., Zheng, S., Huang, K.: An empirical study of visual features for part based model. In: ACPR. (2011) 219–223
22. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.: Discriminatively Trained Deformable Part Models, Release 4. (2010)
23. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV. (2009) 229 –236
24. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV. (2009) 606 –613
25. Razavi, N., Gall, J., van Gool, L.: Scalable multi-class object detection. In: CVPR. (2011) 1505 – 1512
26. Divvala, S.K., Zitnick, C., Kapoor, A., Baker, S.: Detecting objects using unsupervised parts-based attributes. Technical Report CMU-RI-TR-11-10, Robotics Institute, Pittsburgh, PA (2010)
27. Schnitzspan, P., Fritz, M., Roth, S., Schiele, B.: Discriminative structure learning of hierarchical representations for object detection. In: CVPR. (2009) 2238 –2245
28. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-svms for object detection and beyond. In: ICCV. (2011) 89 – 96
29. ul Hussain, S.: Machine Learning Methods for Visual Object Detection. PhD thesis, University of Caen (2011)