Learning Weighted Part Models for Object Tracking

Chaoyang Zhao, Jinqiao Wang, Guibo Zhu, Yi Wu and Hanqing Lu

National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. Beijing, China.

Abstract

Despite significant improvement has been made for visual tracking in recent years, tracking target with variant deformable appearance is still a challenging problem. In this paper, we present an approach based on weighted part models that can efficiently handle target appearance change and occlusion during tracking. The object appearance is modeled by mixture of deformable part models with a graph strcture. To adjust the contribution of different parts in the whole tracking process dynamically, we add a weight for each part based on the its Gaussian mixture distribution. The weights of the parts assist to adjust the importance of the discriminative appearance models by sustained appearance temporal distributions. Proper training samples are sampled from the Gaussian mixture distributions of the related parts for model parameter updating. Experimental results show that our approach improves tracking performance by refining part appearance models with part weights involve in.

Keywords: deformable part tracker, GMM model, weighted part models

1. Introduction

Visual tracking is one of the fundamental problems in computer vision and serves as a preprocessing step for many applications including human machine interaction, video surveillance, or higher tasks like scene understanding and action recognition. In recent years, significant progress has been made for visual tracking. However, designing a robust tracker for general object tracking is still a major challenge, especially when occlusion and appearance variation of the target object happens. Many approaches have been developed to address the challenge caused by object occlusion and appearance variations, such as tracking-by-detection approaches proposed by Avidan (2004). This kind of solutions treat the tracking problem as a detection task and transferred a great deal of detection ideas to tracking as proposed by Blaschko and Lampert (2008); Yao et al. (2013); Zhang and van der Maaten (2013). Through online training a classifier, it distinguishes the target from the background. During the tracking process, the classifier is used to locate the object with maximum classification score and collect proper samples for parameter updating. By leverage the detection techniques into tracking, the tracking-by-detection solutions yield great improvements in recent years. However, these kind of approaches can often hardly deal with the problems such as object part occlusions or classifier generalization.

Due to the huge success achieved by the part based object detection approaches proposed by Felzenszwalb et al. (2010), tracking object with parts also shows its advantage as shown by Yao et al. (2013). During tracking, target object together with its related parts and the inherent structures among them are modeled together in a uniform framework. The usage of part based

models shows favorable properties such as robustness to partial occlusion and articulation. The existing part based tracking approaches as proposed by Yao et al. (2013); Zhang and van der Maaten (2013) often adapt same part sample selection strategies and update the part appearance parameters controlled by the appearance scores. However, with time varying, object appearance variation is not a uniform process, different parts may change significantly different from each other due to the 2D view of the frame, the change rates of different parts vary significantly. Therefore, it is not appropriate to treat each part's contribution in the same way as this may cause tracking failure when drastic appearance variation occurs. Take the "lemming" sequence as a example, in the first row of Fig.1 we can see that the appearances change differently for different parts with the target moving. This indicates that in the tracking process, different parts of the target should have different updating strategies with respect to their contribution to the tracked target.

In this paper, we present a weighted part models based tracking approach. A weight is introduced for each part to adjust its contribution to the tracker. Based on the temporal distributions of the part appearance, the weighted part appearance models together with the spatial constrains among them enhance the final tracking result. The dynamic evolution of each part is modeled with a Gaussian mixture distribution that serve as complements to the discriminative part appearance models. The proposed weighted part models exceed normal part models for two reasons: the weights of the parts assist to adjust the importance of the part appearance models by measuring their fitness to the related appearance temporal distributions. They also control the sample selections for model parameter updating by judging whether

the samples are fit for the appearance history distributions. The second and the third row of Fig.1 illustrate the weights and the final tracking results respectively.

The main contribution of this paper is summarized as follows,

1. We propose weighted part models tracker to enhance the deformable part models based tracking approaches.
2. We use the Gaussian mixture distributions to model the temporal distributions of the object part, which serve as the weights of the object part.
3. We propose a new online updating algorithm to update the tracker parameters more efficiently.

2. Related Work

Much of the recent work in model-free tracking focuses on tracking-by-detection methods. Avidan (2004) used off-line SVM to detect the target vehicle. Grabner *et al.* Grabner et al. (2006) used online AdaBoost to update selected features incrementally. Babenko et al. (2009) used online multiple instance boosting to collect training samples for online classifier updating. Kalal et al. (2012) combined tracking and detection results to refine the final tracking decision. Kwon and Lee (2010) decomposed the tracker into multiple basic observation models and improved the tracking results by each of the tracker's contribution. These approaches leverage the advantages of the detection methods, but they require carefully design of the training sample selection strategy for the reason that improper online updating always brings tracking drift.

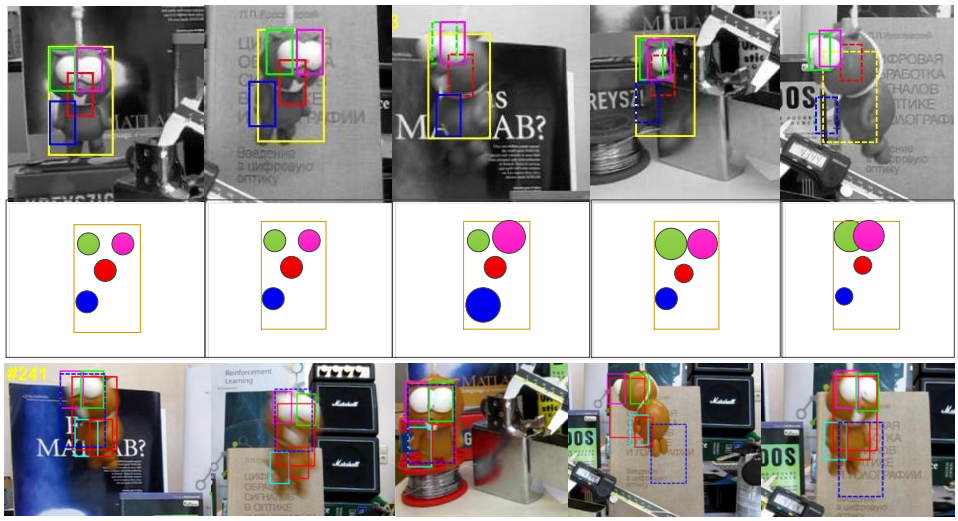

Figure 1: Demonstration of weighted part models tracker. The first shows that object parts suffer varied change rate due to uncontrolled object deformation; the second row illustrates part weights adjusted by our weighted part models; the third row shows the tracking results of weighted part models (red solid box) and the normal part based approach (blue dash box).

Structured learning approaches make predictions over large output spaces that have certain inherent structure. Blaschko and Lampert (2008) detected object locations by structured output predictions. Hare et al. (2011) further used structured SVM to predict object locations by modelling the problem as the prediction of the bounding box change between frames. Yao et al. (2012) adjusted the training sample importance by using weighted reservoir sampling with structural learning framework. These structure based methods achieve good performance in recent years, however, they neglect the different contribution of different local parts, and consider that they have equal importance. Our approach benefits from the structural learning methods and

introduces weighted part models to boost the tracking performance.

Early part based tracking approaches often required manual part initialization as mentioned by Hua and Wu (2006) or prior knowledge proposed by Kwon and Lee (2009) before tracking. Deformable part models proposed by Felzenszwalb et al. (2010) is well known in object detection task for its robust to object appearance changing and easy to extension. The models formulate the part appearances together with the spatial structure between parts. The parameters were learned from latent SVM framework. This spring form spatial constrains were further extended to pose estimation proposed by Yang and Ramanan (2011), face landmark detection proposed by Zhu and Ramanan (2012), and then was introduced to visual tracking. Yao et al. (2013) used part models for visual tracking with a online latent structural learning method. Zhang and van der Maaten (2013) modeled the spatial constrains between parts or multiple targets. Different from these part based approaches for object tracking, we use the dpm based graph models to track the object and also adjust the importance of the part appearance models by introducing a weight for each part. The updating strategy of different parts also refined by the weights of the parts. This adjustment is based on the observation that the part deformations are not always the same, different part models should be weighted and updated differently based both on the traditional part appearance models and on the deformation rates to their temporal evolution distributions.

3. The Proposed Approach

In this paper, we propose a novel approach for online tracking with weighted part models. We first describe the graph structure of the part models used for online tracking, and then propose the formulation of part weights that models the temporal evolution distributions of the part appearances during tracking. After the model inference and learning strategies, we then discuss the sample selection strategies controlled by the weighted part models.

3.1. Graph Construction for Part Models

The deformable part models (DPM) were presented by Felzenszwalb et al. (2010), which uses HOG features proposed by Dalal and Triggs (2005) to describe image patches and formulated the object together with its part appearance models with a graph structure. We first describe the DPM approach that can be used for object tracking, which formulated object parts together with their spatial relationships with a structured graph model. We represent object part i by $B_i = \{\mathbf{x_i}, w_i, h_i\}$ with center location $\mathbf{x_i} = (x_i, y_i)$, width and height. Together with the whole object as the root B_0, we can define a graph $G = (V, E)$ to describe the object. The vertex set $V = \{B_0, B_1, ..., B_n\}$ stands for object parts together with the root. The edges $(i, j) \in E$ between root and different parts represents the spatial relationship. Thus the final score of a object can be calculated as the sum of the root score and the part scores together with their spatial constrains:

$$S(X_t) = \sum_{i=0}^{n} F_i \cdot \phi(\mathbf{x_i}) - \sum_{i=1}^{n} d_i \cdot \phi_d(\delta(\mathbf{x_i})) \tag{1}$$

where $\phi(\mathbf{x_i})$ means the appearance feature of part i, here stands for the HOG feature Dalal and Triggs (2005) extracted from the corresponding part, $\mathbf{x_0}$ stands for the root. As we only consider the part spatial layout to the root, $\delta(\mathbf{x_i})$ means the spatial layout between part i and the root. The compatibility between part i and root $\phi_d(\delta(\mathbf{x_i}))$ is represents by the spatial layout of part i with respect to the root:

$$\phi_d(\delta(\mathbf{x_i})) = (dx, dy, dx^2, dy^2) \tag{2}$$

where $\delta(\mathbf{x_i}) = (dx, dy) = (\mathbf{x_i} - \mathbf{x_0})$ stands for the spatial constraint of the part i to the root. When the detector is used for tracking in a tracking-by-detection manner, the target parts together with their spatial distributions help to assist the final tracking score of the target. The parameter $w = \{F_0, F_1, ..., F_n, d_1, ..., d_n\}$ need to be updated with an online learning strategy. Yao et al. (2013) proposed a two stage training algorithm to update the parameters. Zhang and van der Maaten (2013) slacked the spatial constrains and used the part models for multiple object tracking. In the next subsection, we focus on designing a new strategy to assist the part appearance scores and the part updating process in part weighted manner, further to refine the final tracker's performance.

3.2. Weighted Part Models Tracker

In visual tracking, the object and background appearance varies with time, parts of the object provide different level of information about the current appearance of the target object and background. To better reflect the contributions of the parts to the tracked object, we introduce weights

for different parts to adjust their appearance models due to their different change rate during tracking. We add a weight λ_i for part i with appearance model Sa_i. The part weights also help to sustain a sample pool used for each part's parameter update. Given the part based models in Eq.1, the weighted part models are formulated as follows:

$$S(Y_t; I, \Theta) = \sum_{i=0}^{n} \lambda_i(\mathbf{x_i}) \cdot Sa_i(\mathbf{x_i}) + \sum_{i=1}^{n} D_i(\mathbf{x_i}) \tag{3}$$

where $Y = (B_0, B_1, ..., B_n)$ is the output configuration the object together with its related parts. B_0 is the bounding box of the target (root box) and $B_i,\ i = 1, ..., n$ are the n part boxes. x_i is the relative part feature extracted from the input image I. Θ is the model parameter. $Sa_i(\mathbf{x_i}) = F_i \cdot \phi(\mathbf{x_i})$ and $D_i(x_i) = d_i \cdot \phi_d(\delta(\mathbf{x_i}))$ are the appearance score and the deformation cost separately. $\lambda_i(x_i)$ is the weight for the object part i. In this case, the weights serve two purposes: adjusting the importance of the part appearance models by measuring their fitness to the related temporal distributions of the parts, and selecting the proper training samples for updating the related parameters.

3.2.1. Weight formulation.

Here we use the Gaussian Mixture Distributions (GMM models)proposed by Stauffer and Grimson (1999) to formulate the weight of each part. The GMM models describe the part appearance distributions with time varying and serve as complements for the discriminative models of the part appearances. For each part i, we have:

$$\lambda_i(\mathbf{x_i}) = \sum_{k=1}^{K} \omega_k \cdot \mathcal{N}_i(\mathbf{x_i}|\mu_k, \Sigma_k) \tag{4}$$

where ω_k is the weight parameter of the k-th Gaussian component. $\mathcal{N}(\mathbf{x_i}|\mu_k, \Sigma_k)$ is the Normal distribution of the k-th component represented by:

$$\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) = \frac{1}{2\pi\sqrt{\Sigma_k}} exp(-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1}(\mathbf{x} - \mu_k)) \tag{5}$$

where μ_k is the mean and $\Sigma_k = \sigma_k^2 \mathbf{I}$ is the covariance of the k-th component of the GMM. The GMM models create sample distributions for related parts, which describe the temporal change of the appearance and response the confidence level of the parts in the current frame to their history distributions. Thus the GMM models can be used to assist the confidence level of the SVM based part appearance scores and select proper samples for parameter updating in a weighted manner. The GMM models describe the temporal evolution for each part. they build sample distributions for related parts, which describe the temporal change of the appearance and response the confidence level of the parts in the current frame to the history distributions. When object deformation and occlusion happen, the GMM models assign different weights to different parts in the appearance model. Here we use the raw gray pixel values of the target parts as the input features of the GMM models for the reason that the gray feature is more robust than the rgb feature in our experiments.

GMM Initialization In this paper, we build one Gaussian Mixture model for each part of the target. The initialization of the GMM parameters are relatively sensitive. For K Gaussian mixtures, we set one of the them as the "main Gaussian", the mean of which is initially computed from the mean of the initial training samples. The weight of the "main Gaussian" is set to be higher than Gaussian mixtures. Here we do not use the uniform weight value for initialization for that the "main Gaussian" benefit most from the initial

training samples and represents the initial distribution of the particular part. All the other $K-1$ Gaussian mixtures are initialized with random generated means with a range of $0 \sim 255$ and a length controlled by target part size. The variances define the capacities of the Gaussian mixtures, here we set the value of the "main Gaussian" to be 60 and randomly generate the variances of other mixtures from 50 to 70. During the implementation, we set the number of Gaussian mixtures to be three for each GMM model of the part. The initial weights are set to be $0.4, 0.3, 0.3$ for the "main Gaussian" and all the other Gaussian mixtures of each GMM. Details about GMM initialization parameter settings will be discussed in section 4.

3.2.2. Inference.

Given the parameters of the tracker and a new frame, tracking the target is to find the most likely object configuration to maximize Eq.3:

$$\hat{Y_t} = arg \max_{\hat{Y_t}} S(Y_t; I, \Theta) \tag{6}$$

The root together with the parts form a tree structure (star structure) as suggested by Felzenszwalb et al. (2010), so the problem in Eq.1 can be efficiently solved by the dynamic programming strategy. To find the best output configuration Y in Eq.6, we use a modified distance transform approach used by Felzenszwalb et al. (2010) with GMM function of each part involved in. The outputs of the GMM models serve as the weight of the part appearance scores of the target. Whenever the object appearance changes drastically such as moving with occlusions or deformations, the GMM models help to adjust the importance of the part appearance results by measuring its fitness to the temporal distributions. When finding the most suitable parts for the

tracked target, GMM models help to decide whether the part samples can be used for parameter updating, details are described below.

3.2.3. Learning.

After observing an image I and the inferred object configuration Y in Eq.6, we update the parameters using the sample pool sustained by the GMM models. The parameters in Eq.3 can be denoted by Θ:

$$\Theta = [\lambda_1^T, ..., \lambda_n^T, F_0^T, F_1^T, ..., F_n^T, d_1^T, ..., d_n^T]^T$$

where $\lambda_i = \sum_k (\omega_k, \mu_k, \Sigma_k), k = 1, ..., K$ stands for the GMM model parameter of part i, F_i and d_i are related to appearance and deformation parameters in Eq.1 respectively. For the t-th frame, we learn the parameter Θ_t with training samples from sample pool. In our implementation, we found it more efficient to update the GMM and the part model parameters separately, here we update Θ with a two stage manner. By splitting parameters into $\Theta_t = [\Theta_{gmm}, \Theta_{parts}]$, where $\Theta_{gmm} = [\lambda_1^T, ...\lambda_n^T]^T$ and $\Theta_{parts} = [F_0^T, F_1^T, ..., F_n^T, d_1^T, ..., d_n^T]^T$ are the GMM and part model parameters respectively.

During the first stage, the update of Θ_{gmm} is inspired by the background modeling method proposed by Stauffer and Grimson (1999). For certain part sample x_{t+1}^i of part i that suits the part appearance model, it can be used to update the k-th Gaussian model if satisfying:

$$|x_{t+1}^i - \mu_k^i| < \delta^i \sigma_k^i \tag{7}$$

where δ is the matching threshold for part i. we can then update the Gaussian

model by:

$$
\begin{aligned}
&\omega^i_{k,t+1} = (1-\alpha^i)\omega^i_{k,t} + \alpha^i \\
&\mu^i_{k,t+1} = (1-\rho^i)\mu^i_{k,t} + \rho^i x^i_t \\
&(\sigma^i_{k,t+1})^2 = (1-\rho^i)(\sigma^i_{k,t})^2 + \rho^i (x^i_t - \mu^i_{k,t})^T (x^i_t - \mu^i_{k,t}) \\
&\rho^i = \frac{\alpha^i}{\omega^i_{k,t}}
\end{aligned}
\tag{8}
$$

where α^i and ρ^i are the learning rate of the model and the parameters respectively for the i-th part GMM λ_i. The new GMM score of the sample x^i_t is computed by using Eq.4 and then be used for updating the sample pool of the part i.

The second stage is to update the root and part appearance and deformation parameters Θ_{parts}, which can be updated by minimizing the following structured SVM objective function:

$$
\begin{aligned}
&g(\Theta_{parts}, t) = \frac{1}{2}\|\Theta_{parts}\|^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i \\
&s.t.\ \ S(Y; I, \Theta_{part}) - S(\hat{Y}; I, \Theta_{part}) + \Delta(Y, \hat{Y}) \geq 1 - \xi_i \ \ \ i = 1, ..., N
\end{aligned}
\tag{9}
$$

where $\Delta(Y, \hat{Y})$ measures the cost introduced by the predicted output configuration $\hat{Y}$, for the object with n parts with respective bounding boxes $B_i, i = 0, ..., n$ ($i = 0$ stands for the bounding box of the object). N is the number of training samples sustained by the sample pool. $S(\cdot)$ is the target score function in Eq.3 with GMM based weight score λ_i pre-computed ($\lambda_0 = 1$ for the root):

$$
\begin{aligned}
S(Y; I, \Theta_{part}) &= \sum_{i=0}^{n} \lambda_i F_i \cdot \phi_i(X_i) + \sum_{i=1}^{n} d_i \cdot \phi_{di}(\delta(X_i)) \\
&= \Theta^T_{parts} \cdot \Phi(Y)
\end{aligned}
\tag{10}
$$

where

$$\Phi = [\lambda_0 \phi_0^T, ..., \lambda_n \phi_n^T, \phi_{d1}^T, ..., \phi_{dn}^T]^T.$$

With the configuration cost $\Delta(Y, \hat{Y})$ defined as

$$\Delta(Y, \hat{Y}) = \sum_{i=0}^{n} (1 - \frac{B_i \cap \hat{B}_i}{B_i \cup \hat{B}_i}) \tag{11}$$

we can write the structured SVM loss L for Eq.9 as suggested in Zhang and van der Maaten (2013):

$$\begin{aligned} L(\Theta_{part}; I, Y) &= \max_{\hat{Y}} (S(\hat{Y}; I, \Theta_{part}) - S(Y; I, \Theta_{part}) + \Delta(Y, \hat{Y})) \\ &= \max_{\hat{Y}} (\Theta_{part}^T (\hat{\Phi} - \Phi) + \Delta(Y, \hat{Y})) \end{aligned} \tag{12}$$

the gradient of L with respect to $\theta \in \Theta_{part}$ is given by:

$$\bigtriangledown_\theta L(\Theta_{part}; I, Y) = \bigtriangledown_\theta S(Y^*; I, \Theta_{part}) - \bigtriangledown_\theta S(Y; I, \Theta_{part}) \tag{13}$$

in which the configuration Y^* is acquired by solving the altered inference problem:

$$Y^* = arg \max_{\hat{Y}} (S(\hat{Y}; I, \Theta_{part}) + \Delta(Y, \hat{Y})) \tag{14}$$

Then the parameter can be updated by the passive-aggressive algorithm K. Crammer and Singer (2006); Zhang and van der Maaten (2013):

$$\theta = \theta - \frac{L(\Theta_{part}; I, Y)}{\| \bigtriangledown_\theta L(\cdot) \|^2 + 0.5} \bigtriangledown_\theta L(\cdot) \tag{15}$$

3.2.4. Sample pool update.

Our tracker sustains a pool of training samples of root and all the parts for the parameter updating. We adopt two strategies to update the sample pool: the appearance scores of the root and the parts computed by the

discriminative models, and the temporal distributions of each part indicated by the part weights. Whenever the appearance score $Sa_i(\mathbf{x_i})$ exceeds a certain threshold, the sample can be seen as a candidate of the pool for the root appearance model. Due to the different change and deformation rate for different parts during tracking, we further use the GMM outputs to control the candidate selection process for each part. For one particular part, all the training samples in the part sample pool are sorted by their GMM score computed through Eq.4, the one with the lowest score is replaced by the incoming candidate sample. In this way, we can select the certain samples of the gaussian mixture distributions that best suitable for parameter updating in the current frame. Here the part appearance scores are responsible for the coarse sample selection while the GMM models adjust the selection by checking whether the samples are suitable for the part temporal distributions.

The detailed online algorithm of our weighted part tracker is shown in Alg.1. For the first frame, model initialization includes the initial part selection strategy, initial sample pool construct, and initial model parameter learning. Details about initial part selection strategies are discussed in sec. 4.1. After initial root and parts locations are fixed, we sample positive training samples near the selected parts with 1 to 3 pixels shifted for the root and parts and 50 negative examples have little to no overlap with them to form the initial sample pool. And then we train the initial trackers with the initial sample pool.

4. Experiments

To evaluate the performance of our tracker, we run our tracker on thirteen challenging sequences Wu et al. (2013). These sequences contain varied tracking targets with different challenging situations in object tracking, which includes partial or full occlusion, shape deformation, illumination, and pose/scale variation. We evaluate the performance of the trackers with the following measures: (1) the mean center position error (CLE) per frame and (2) the correct detection rate (CDR), the average percentage of frames with correct detection. A correct detection is defined with the Pascal VOC overlap rate $R_{overlap} \geq 0.5$, where $R_{overlap} = Area(B_T \bigcap B_{GT})/Area(B_T \bigcup B_{GT})$, B_T and B_{GT} are the tracking result and the ground truth bounding box respectively. We first evaluate the part initialization strategy and the effectiveness of the part weight strategy, then we study the performance of our tracker for single object tracking compared with several state-of-the-art approaches.

Our tracker is implemented in MATLAB on a PC with an Intel Core i7 3.4GHz processor and 4G RAM. The average running time is about 10 frames per second. The number of Gaussians for each part weighting model is set to be 3 and the part number is set to be 3 4 for all the sequences.

4.1. Part initialization.

Because only a single bounding box is provided to annotate the object in the first frame of the video, all parts need to be carefully selected for later use. Here we implemented three approaches for initial part selection:

Init 1 – Heuristic part selection approach used in Zhang and van der Maaten (2013). Parts are initialized at the location in which the weights of

the initial global SVM w are large and positive. The overlap between parts cannot be more than 50%. Part size is manually set to be 0.4 of the bounding box size.

Init 2 – Automatic part selection approach. Parts are selected by averagely splitting the bounding box into different sub-boxes.

Init 3 – Automatic heuristic part selection approach. Parts are selected similarly as the method of Init 1, but with a automatic size and shape selection strategy as proposed byFelzenszwalb et al. (2010).

All the features of the parts are extracted on the same scale as the root bounding box features as suggested by Zhang and van der Maaten (2013). To investigate the effect of different part selection methods, We conduct two experiments with the above three strategy. The experiment results are shown in Fig.2. The left figure in Fig.2 shows the CLE performance and the selected parts for different methods in sequence *david* while the right one shows the performance in sequence *sylvester*. The results show that for most frames, the three initialization methods achieve similar performance, which indicate that our tracking algorithm is robust to different the part initializing strategies. We set the rest of our experiments run with *init* 3 selected parts since we don't need to adjust the part size. Due to the recent success achieved by introducing context information such as recent works proposed by Wen et al. (2012); Yang et al. (2009); Zhang et al. (2013), we also allow the part box to cover the area around the target bounding box in the first frame. For the strategies of *init1* and *init2*, the bounding box is enlarged to be 1.2 times of the root bounding box when selecting the object parts.

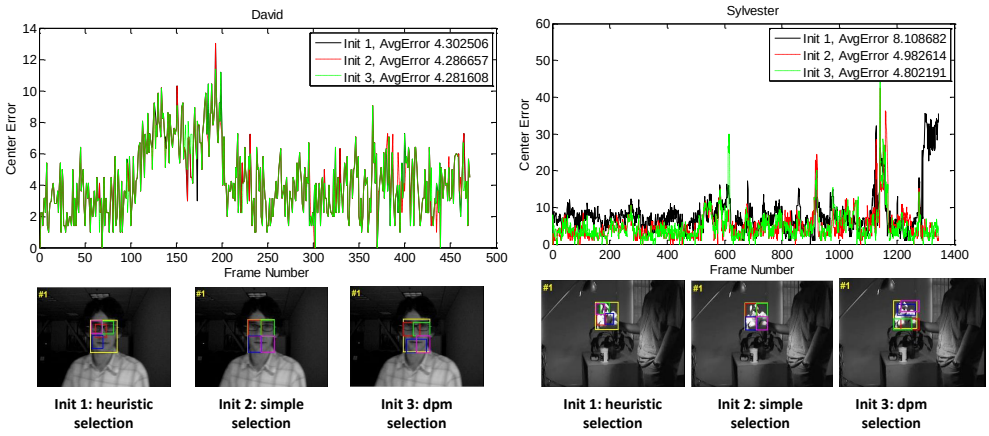

Figure 2: The center location error of our tracker with different part initialization strategies on two sequences "David" and "Sylvester".

4.2. Part weighting model initialization.

Here we discuss initialization strategy for the part weighting models. As shown in Fig.3, the experimental result shows that the tracker's performance can be improved with carefully designed Gaussian mixture numbers for different parts, however, the differences are relatively small. Thus we use three Gaussian mixtures for each part related GMM model.

Table 4.2 shows the results for different GMM parameter initialization strategies. For "mean" approach, we set the mean of each Gaussian of one particular part GMM model to be the mean of the related part samples; for "random" approach, we randomly initialize the Gaussian means of the particular GMM model. For both of these settings, the weights of the Gaussian in each GMM model are set to be equal. Our approach described in section 3.2.1 yields best result. Therefore, we initially set the GMM parameters as described in section 3.2.1.

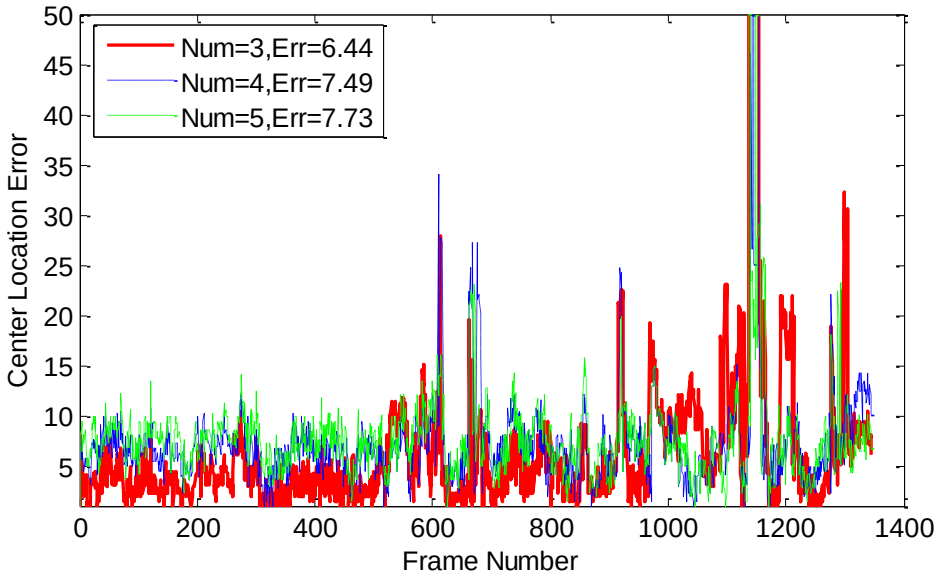

Figure 3: Tracking results comparison with different Gaussian numbers for GMM based part weighting functions on "Sylvester".

4.3. Tracking with weighted parts.

We then evaluate the effectiveness of our weight functions. Experiment results can be found in Fig.4. The results of the first row in Fig.4 show that the weight of each part contributes to the final target score. With the GMM based weights participated in, the tracking result on the left figure in Fig.4 is more robust than the one shows on the left without part weights. The left figure of the first row shows that with GMM weighted part, proper bounding box of the object yields highest score (red bounding box on the left); the right figure shows that without part weights, the highest score indicates a shifted bounding box (red bounding box on the right), the proper target location (blue bounding box) shows lower score due to part appearance score shift-

ing. The second row demonstrates the related location of all the parts with respect to the root location, from which we can see that with weighted part models, the part related locations are relatively steady compared to the part related locations without GMM based part weights. This suggests that our part re-weighting strategy makes the parts more steady during tracing and more robust to the object appearance changes such as occlusion and pos variation. As shown in Fig.5, the first row shows the changes of the part weights (GMM outputs) together with appearance scores of the four parts for first 100 frames. We can see that the part weights computed by GMM models show different distributions other than part appearance scores acquired by the discriminative appearance models. The part weights describe the distributions of part appearances in a different view other than the discriminative appearance models, and serve as complements to the part appearances. The bottom row is the demonstration of the final tracking result with parts located. Part locations are refined by the GMM based part weights to capture more representative object parts. Thus more reliable samples for parameter update can be acquired.

Further results for the comparison can be found in Fig.6, the CLE plots for eight sequences show that the part based models ensures the performance gain for most of the sequences. We believe that certain parts appearance scores may be confused with the background thus to make negative contributions to the tracker during tracking. The GMM based part weights can reduce these phenomenons by sustaining a temporal distributions of parts during tracking, certain false part scores can be depressed by smaller part weights.

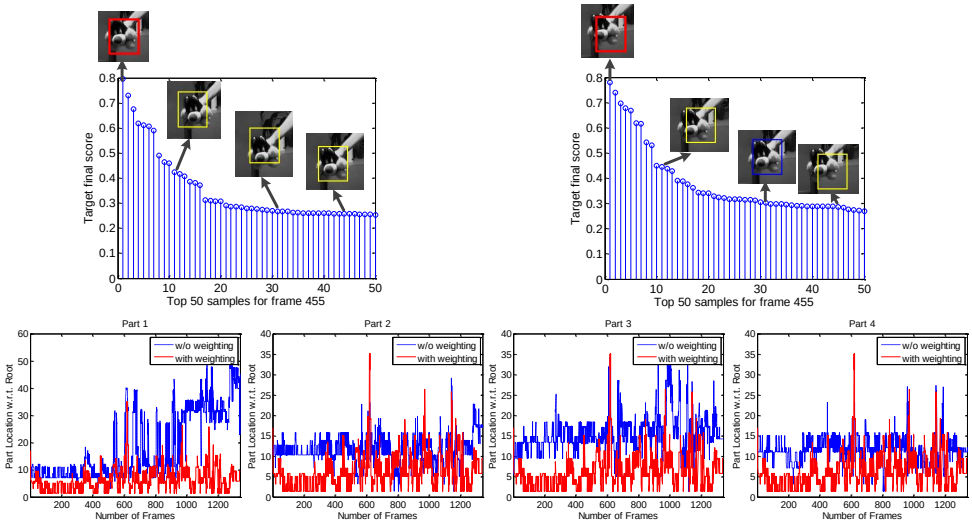

Figure 4: Comparison of trackers with and without GMM based part weight strategy on "sylvester" sequence. The top row shows the top 50 target scores at frame 455. Left: with part weights; right: without part weights. The second row shows the results of parts location changing with respect to the root.

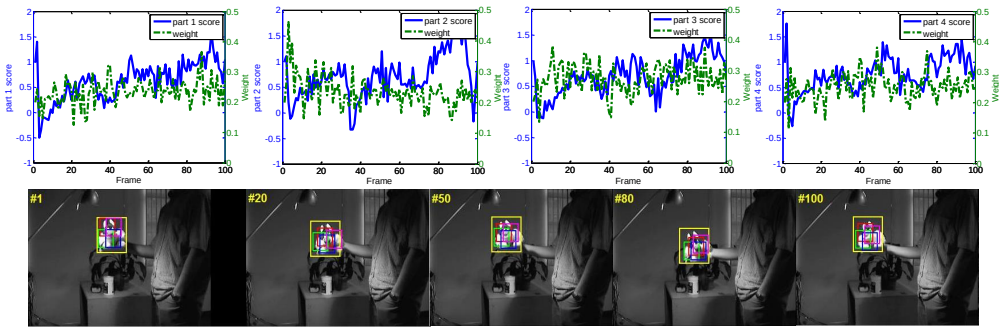

Figure 5: Demonstrations of tracking results with part weighting strategy. The first row shows the changes of the part weights together with the appearance scores of the four parts for first 100 frames. The second row is the demonstration of the final tracking result with parts shown in different colors.

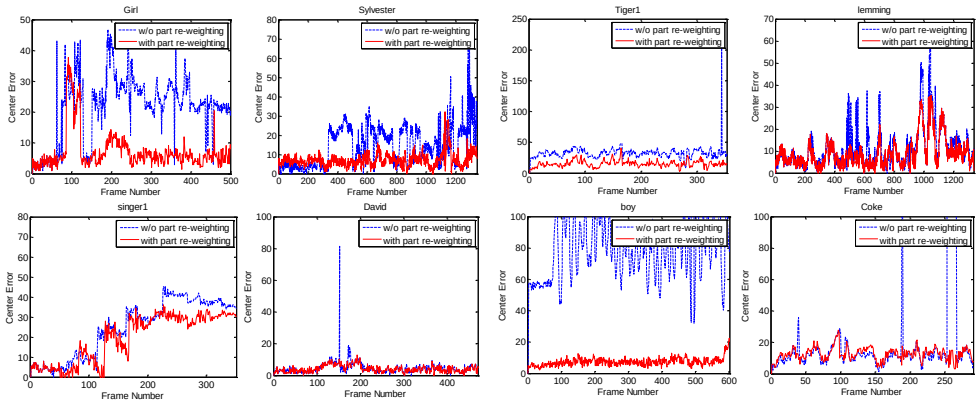

Figure 6: Per frame CLE comparison of with and without GMM based part weights on eight sequences. Here we implemented our own version of dpm based tracker and refer it as "without part re-weighting" method.

4.4. Comparison with state-of-the-arts.

We compare the performance of our tracker with six state-of-the-art trackers: the SPOT tracker (SPOT) by Zhang and van der Maaten (2013), the Struck tracker (struck) by Hare et al. (2011), the MIL tracker (MIL) by Babenko et al. (2009), the OAB tracker (OAB) by Grabner et al. (2006) and the TLD tracker (TLD) by Kalal et al. (2012). All trackers' source code are available online, here we run all tracker with default configuration (all trackers but TLD are with single scale), we run each track for five rounds and report the average results below.

Table.1 and Table.2 show the CLE and CDR results respectively. Our approach outperforms nine of the fourteen sequences with CLE measurement. The results in sequences *girl* and *boy* are mainly caused by the small target size. The usage of HOG feature is not suitable for small targets. further implementation with other features Haar feature may further bring performance

gain. Our tracker achieves good results for the rest of the tested sequences. The CDR result of sequence *singer, david* is mainly caused mainly by the scaled ground truth label, all the trackers except TLD are ran at single scale. Our tracker also achieves reasonable results for the rest of the tested sequences for the tracking successful rate comparing. Visualized tracking results of different trackers can be seen in Fig.7, which reveal the potential benefit of using carefully designed part weights and updating strategy.

5. Conclusion

In this paper, we demonstrated a novel approach for online object tracking with weighted part models. To balance the varied changing rate for different object parts, we introduce GMM model based part weights to assist the part appearance models and guarantee the proper sample selections for parameter updating. GMM based part weights sustain dynamic distributions of the object parts in a temporal evolution view other than the discriminative part appearance models and can be used to refine the part appearance contributions to the final tracker. Experimental results showed that the proposed weighted part models achieved the state-of-the-art performance.

References

Avidan, S., 2004. Support vector tracking. Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 (8), 1064–1072.

Babenko, B., Yang, M.-H., Belongie, S., 2009. Visual tracking with online multiple instance learning. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, pp. 983–990.

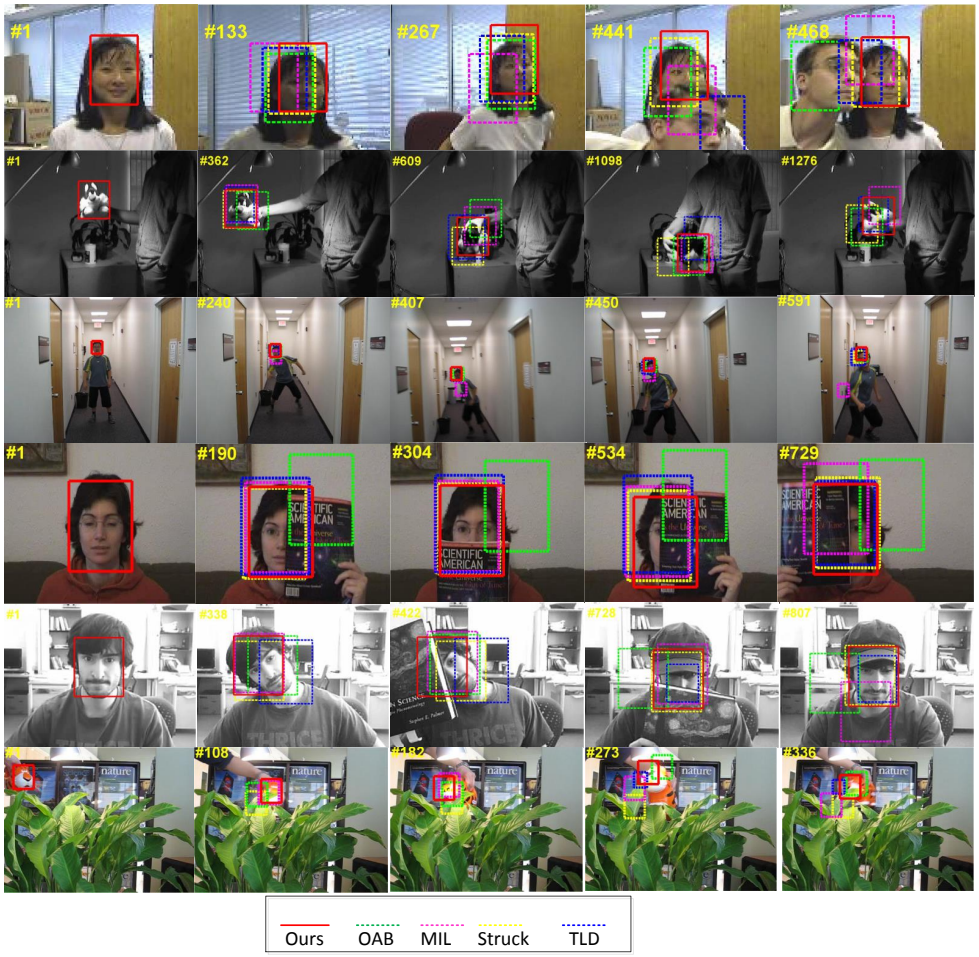

Figure 7: Tracking results on six of the 14 videos(*girl, sylvester, boy, occlusion1, occlusion2, tiger2*) obtained by ours, OAB, MIL, Struck and TLD trackers.

Blaschko, M. B., Lampert, C. H., 2008. Learning to localize objects with structured output regression. In: Computer Vision–ECCV 2008. Springer, pp. 2–15.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human de-

tection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, pp. 886–893.

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

Grabner, H., Grabner, M., Bischof, H., 2006. Real-time tracking via on-line boosting. In: BMVC. Vol. 1. p. 6.

Hare, S., Saffari, A., Torr, P. H. S., Nov 2011. Struck: Structured output tracking with kernels. In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 263–270.

Hua, G., Wu, Y., 2006. Measurement integration under inconsistency for robust tracking. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Vol. 1. IEEE, pp. 650–657.

K. Crammer, O. Dekel, J. K. S. S.-S., Singer, Y., 2006. Online passive-aggressive algorithms. In: Journal of Machine Learning Research. p. 551585.

Kalal, Z., Mikolajczyk, K., Matas, J., 2012. Tracking-learning-detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34 (7), 1409–1422.

Kwon, J., Lee, K. M., 2009. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo

sampling. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, pp. 1208–1215.

Kwon, J., Lee, K. M., 2010. Visual tracking decomposition. In: CVPR. pp. 1269–1276.

Stauffer, C., Grimson, W. E. L., 1999. Adaptive background mixture models for real-time tracking. In: Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. Vol. 2. IEEE.

Wen, L., Cai, Z., Lei, Z., Yi, D., Li, S. Z., 2012. Online spatio-temporal structural context learning for visual tracking. In: Computer Vision–ECCV 2012. Springer, pp. 716–729.

Wu, Y., Lim, J., Yang, M.-H., June 2013. Online object tracking: A benchmark. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. pp. 2411–2418.

Yang, M., Wu, Y., Hua, G., 2009. Context-aware visual tracking. Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 (7), 1195–1209.

Yang, Y., Ramanan, D., 2011. Articulated pose estimation with flexible mixtures-of-parts. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, pp. 1385–1392.

Yao, R., Shi, Q., Shen, C., Zhang, Y., van den Hengel, A., 2012. Robust tracking with weighted online structured learning. In: European Conference on Computer Vision (ECCV), 2012. pp. 158–172.

Yao, R., Shi, Q., Shen, C., Zhang, Y., van den Hengel, A., 2013. Part-based visual tracking with online latent structural learning. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, pp. 2363–2370.

Zhang, K., Zhang, L., Yang, M.-H., Zhang, D., 2013. Fast tracking via spatio-temporal context learning. arXiv preprint arXiv:1311.1939.

Zhang, L., van der Maaten, L., June 2013. Structure preserving object tracking. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. pp. 1838–1845.

Zhu, X., Ramanan, D., 2012. Face detection, pose estimation, and landmark localization in the wild. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, pp. 2879–2886.

Algorithm 1: Weighted Part Models Tracker

Input:

model parameter Θ^t_{gmm} and Θ^t_{parts},

sample pool for root and all the parts $\mathcal{P}^t_i$,

and the low level image feature x_{t+1} for the $(t+1)_{th}$ frame.

Output: $(t+1)_{th}$ frame target tracking result $\hat{Y}_{t+1}$,

updated model Θ^{t+1}_{gmm} and Θ^{t+1}_{parts},

updated sample pool $\mathcal{P}^{t+1}_i$

if $t = 0$ **then**

  $\Theta^t_{gmm} = \phi$, $\Theta^t_{parts} = \phi$, $\mathcal{P}^t_i = \phi$;

  Do model initialization;

for $i = 0, 1, 2, ..., n$ *part* **do**

  Compute root and parts appearance score Sa_i as in Eq.1;

  if $i > 0$ **then**

    Compute part weight λ_i as in Eq.4;

Solve Eq. 6 described in sec. 3.2.2;

for $i = 0, 1, 2, ..., n$ *part* **do**

  Update model parameter $\Theta^{t+1}_{gmm,i}$ and $\Theta^{t+1}_{part,i}$ by using Eq. 8 and Eq. 15;

  Update sample pool $\mathcal{P}^{t+1}_i$ as in described in sec. 3.2.4;

Return $\hat{Y}_{t+1}$, Θ^{t+1}_{gmm}, Θ^{t+1}_{parts} and $\mathcal{P}^{t+1}_i$

Table 1: Compared results for different Gaussian parameter settings of the part GMM models on "Sylvester".

strategies	mean	random	ours
CLE	7.20	8.21	6.44
CDR	0.93	0.92	0.96

Table 2: Compared average center location error (CLE) results on fourteen sequences.

Sequence	Ours	SPOT	Struck	MIL	OAB	TLD
girl	7.20	11.42	**4.25**	15.03	8.84	8.44
Sylvester	**6.44**	7.27	8.00	14.33	16.03	12.46
coke	17.11	22.36	**15.90**	46.39	32.99	31.43
oc. Face1	**17.00**	18.57	21.12	31.04	30.32	32.93
oc. Face2	7.90	**7.73**	9.29	16.96	18.55	15.67
tiger1	**15.03**	15.70	34.21	37.94	99.66	39.80
tiger2	**14.82**	16.42	21.64	44.03	105.35	31.14
david	4.15	**4.00**	8.62	18.69	39.58	4.5
trellis	**3.45**	3.97	16.34	61.80	71.06	27.06
deer	**6.82**	8.24	7.78	72.39	22.15	25.83
boy	7.38	224	**5.17**	13.72	28.74	6.84
singer1	**13.24**	16.82	16.73	22.84	16.51	28.05
fish	**7.20**	7.41	19.46	29.82	37.68	12.56
lemming	**8.97**	11.53	50.97	82.10	61.99	88.33
Avg.	**9.98**	11.65	17.11	36.22	37.24	26.07

Table 3: Compared correct detection rate (CDR) results on fourteen sequences.

Sequence	Ours	SPOT	Struck	MIL	OAB	TLD
girl	0.9	0.87	0.96	0.57	**0.97**	0.78
Sylvester	**0.96**	0.93	0.85	0.73	0.42	0.91
coke	**0.81**	0.75	0.71	0.22	0.47	0.52
oc. Face1	**1**	**1**	0.99	0.78	0.92	0.99
oc. Face2	**1**	**1**	0.98	0.91	0.85	0.77
tiger1	**0.89**	**0.89**	0.83	0.58	0.25	0.13
tiger2	0.87	**0.88**	0.81	0.64	0.44	0.27
david	0.62	0.62	0.62	0.61	0.34	**1**
trellis	**0.84**	0.82	0.66	0.47	0.56	0.66
deer	**1**	**1**	**1**	0.41	0.93	0.97
boy	0.95	0.03	**0.99**	0.52	0.90	0.82
singer1	0.46	0.45	0.41	0.37	0.34	**0.98**
fish	**1**	0.99	**1**	0.81	0.72	**1**
lemming	**0.86**	0.82	0.68	0.52	0.59	0.5
Avg.	**0.87**	0.79	0.82	0.58	0.62	0.74