MULTI-OBJECT MODEL FREE TRACKER

[1]Sweta R.H.,[2]AnuradhaV.K.[3]Madhu V.H. and [4]Laxmi K.,[5] Manjunathraddi Bentur,
[1]UG Student,[2]UG Student, [3]UG Student, [4]UG Student, [5]Asst.Professor
Electronics and Communication,
SKSVMACET, Laxmeshwar ,India

Abstract— -Model free trackers can track the hard-handed objects and analysis of these objects with single bounded box. Model free trackers are playing vital role with simultaneously tracking multiple objects with similar appearance. To solve this we used a new multi-object model free tracker by considering the spatial constraints between the objects. In this paper, we explained an online structured SVM algorithm, which helps to learn the spatial constraints. We demonstrated with experimental results, how the structure-preserving object tracker (SPOT) performance betterments in multi-object tracking and also single object trackers can tracking different parts of the objects occurring at the same time.

Index Terms-SVM Algorithm, image processing, MATLAB

I. INTRODUCTION

In a wide range of domains tracking is a fundamental problem in computer vision with applications. While significant progress has been made in tracking of specific objects, the development of trackers that work well on arbitrary objects remains hard. In model free tracking, in video sequence the object of interest is manually annotated in the first frame. The annotated object needs to be tracked throughout the remainder of the video.

Commonly, a tracking system comprises three main components:

1. Appearance model: Based on the local image appearance it predicts the same that the object is present at a particular location.
2. Location model: It predicts the prior probability that the object is present at a particular location.
3. Search strategy: It finds the maximum a posterior location of the object, in our model free tracker, using histogram-of-gradient (HOG) features the appearance model is implemented.

In many applications, it is necessary to track more than one object. To track multiple objects, run multiple instance of a single object tracker is a simple approach. Such an approach fails to exploit spatial constraints between the objects, so in this paper we argue that this is sub optimal. For instance, cars drives in the same direction on the freeway, nearby people tend to walk in same direction on the side walk, camera shakes all objects will move roughly in the same direction. We show that these types of a spatial relations between objects in model free tracking. In particular we develop a structure preserving object tracker (SPOT) that incorporates spatial constraints between objects using a pictorial structures framework.

Experimental evaluations show that the incorporation of a structural constraints leads to substantial performance improvements in multiple object tracking: SPOT performs very well on Youtube videos with rapidly moving objects,. camera motion, object appearance changes and occlusions. SPOT may be used to improve single tracker objects.

In summary, our main contributions are:

1. We present new approach that performs online learning of pictorial structures models that incorporates spatial constraints between the objects.
2. By simultaneously tracking informative part of the object and a target object we show that our approach may improve the performance of single object model free trackers.
3. We show that our approach can be used to tailor state-of-the-art generic object detectors to particular object.

II. RELATED WORK

For structured prediction models our structure preserving object tracker incorporates ideas from the prior work on model free tracking.

1 Model free tracking

Model free trackers can be categorizes into two types, (1)generative trackers that model only the appearance of the object itself [19],[20],[21] and (2)discriminative trackers that model the appearance of both object and the background [8],[9],[22],[23],[24],[25].Because without considering the appearance of the background, the generative trackers model the appearance of the target, when the background is cluttered or when multiple moving objects are present then it is failed. The target object from the background id distinguishes by classifiers of discriminative trackers. Recent results recommend that discriminative trackers outperform generative trackers [9](similar results have been obtained for model-based trackers, e.g,[26],[27]).This result is supported by theoretical results showing that discriminative models always outperforms their generative counterparts on a discriminative task such as object tracking [28].

This paper will be mainly focus on learning discriminative object appearance models. We will work more on model-free tracking focus on exploring different feature representations for the target object, including feature representations based on points contour, integral histograms, subspace learning [21],sparse representations, and local binary patterns [9]. In this work, we use HOG features instead we capitalize on the success of the Dalal-Triggs [11] and Felzenszwalb [6] detectors.

Recent work on model-free tracking also focuses on developing new learning approaches to better discriminative the target object from the background. In particular prior studies have investigated approaches based on boosting [23], random forest [9],multiple instance learning [8],and structured output learning to predict object transformations our tracker is similar to these approaches in that it updates the appearance model of the target object online. Our tracker differs from previous approaches in that it uses a learner to identify the object parts; we use structured SVM model that is updated online.

This paper is related to many previous studies that illustrate the potential of using contextual information to improve the performance of model-free trackers. In particular ,identify one or more auxiliary points that are tracked in order to improve the location prior of the tracker. Auxiliary regions are generally defined as regions that are salient, that co-occur with the target objects, and whose moment is correlated with

that of the target object. They can be tracked using color based mean shift tracking ,optical flow, or KLT tracking. Our work differs from in that we adapt our model of the spatial relation between objects using an online-learning approaches and our work also differs from previous work in that we do not make distinction between "target" and "auxiliary "objects. Specifically, we use the same features and tracking-by-detection approach for both types of objects and the interaction between target and auxiliary objects is "undirected". This undirected interaction between target and auxiliary objects may lead to a more robust tracker.

In model-based tracking, several studies have also proposed the use of contextual information to improve the performance of the tracker. For instance,[23] makes assumptions on the environment in order to learn and exploit relations between different object group such as humans and chairs. Stalder et al. [27] propose an approach to increase the robustness of model-based-trackers by incorporating constraints on the size of the target objects, on preponderance of the background, and on the smoothness of track trajectories. Roth et al.[28] exploit prior information on the location of target objects by learning separate detection classifier for each image location. Our study is different from [26],[27],[28] in that we take a model-free instead of a model-based approach: our tracker relies on just a single annotation of the target objects.

2. Deformable Template models

Deformable template models exhibits an object by asset of part models which are spatially related. An individual part models the observe portion of the object and spatial relationship between the parts and modeled by certain number of joint distribution over part locations. Most commonly used decision for this joint distribution over the part location involves:1. PCA based models: PCA based models are used in active appearance, constrained local, active shape models which are used in medical image analyses and that use a low rank Gaussian model. 2. Pictorial structure models: Pictorial structures model represents spatial relationship between object parts by springs which can be compressed or exerted.

In model free tracking to train PCA based models commonly there is a insufficient training data. For instance, representation of the hierarchical structure of the human upper body using light parts and further divide these upper body parts into smaller sub parts in order to make the model more flexible. Detectors for the arbitrary objects typically used generic structure model.1)A star shape models which represents the object by root node. ii)A Chow-Liu tree which models only spatial relation between nearby parts.

The key difference with prior studies; however is that we not learn the parameters of the springs in the pictorial structures model based on a large collection of annotated data, but that we learn the spring parameters in an online fashion during tracking.

3. Online learning for structured prediction

The techniques of structured prediction are two types they are

[1] Conditional random field

[2] Structured support vector machine

These predictions make over huge output spaces that have same inherent shape or structure such as sequence, object configurations and it is also used in different type of computer vision problems. The all possible object configuration are searching in the detection of multiple objects can generally be framed same also structured prediction problem.

In our task a structured predictor is not a normal it trained on online based on detections in earlier frames. We are learned algorithm in online based. So online learning of structured predictors are normally based different learning like sub radiant descent algorithms. These two types of learning algorithms assert it means perform cheap parameter updates.

The parameter updates are based on same examples like a single training example or small batch of examples. In task is better in practice after we update our multi object tracker is using a algorithm and another one algorithm is passive aggressive algorithm this is used by the step size for every parameter update.

III. METHODOLOGY

SPOT TRACKER

Object tracker is formed by the popular Dalal-trigg detector by the basis of our structure preserving.

Uses of HOG features:[1]Describe the image patches and SVM to predict the object presence.[2]Magnitude and direction of the image gradient in small cells are the measure of HOG features.

We used small cells 8×8 pixels cells constant normalization is applied on rectangular. A cells of blocks are connected spatially the contrast normalizing L2 norm of all histograms in block according to the L2 norm the resulting values are clipped at 0.2 and then normalized. The advantages of HOG features are 1)They consider more edge orientation than just horizontal ones, they are small regions. They are robust to changes in the illumination of tracked object. 2)HOG features are more sensitive to the spatial location this objection is used to update the classifiers small localization errors. May thus propagate overtime causing the tracks to drift at high frame rates are more over in efficient implementations can extract HOG features.

1. Model

We represent the bounding box that indicates object $i \in V$ (with V representing the set of objects we are tracking) by $B_i = \{X_i, w_i, \mathrm{h_i}\}$ with center location $X_i = (x_i, y_i)$,width w_i, and height hi. The HOG features extracted from image I that correspond to locations inside the object bounding box B_i are concatenated to obtain a feature vector $\emptyset(I; B:)$. Subsequently, we define a graph $G = (V, E)$ over all objects $i \in V$ that we want to track with edges $(i, j) \in E$ between the objects. The edges in the graph model can be viewed as springs that represent spatial constraints between the tracked objects. Next, we define the score of a configuration $C = \{B_1, \ldots, .. B_{|V|}\}$ of multiple tracked objects as the sum of two terms:

(1) an appearance score that sums the similarities between the observed image features and the classifier weights for all objects and (2) a deformation score that measures how much a configuration compresses or stretches the springs between the tracked objects (the relative angles of the springs are not taken into account in our model).

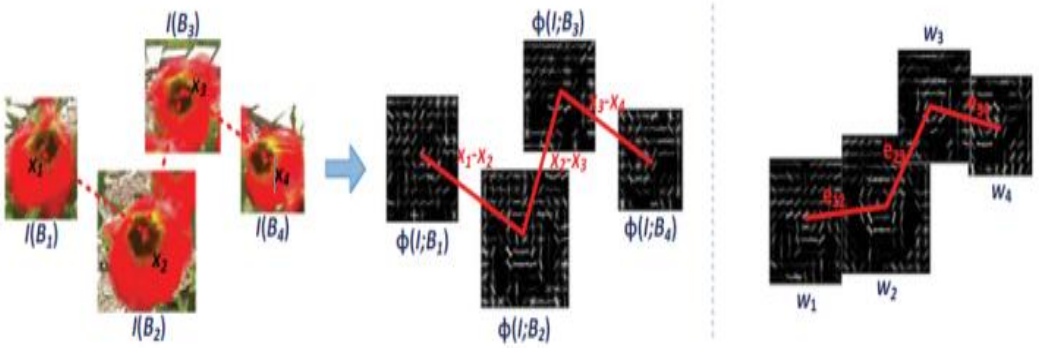

Fig III.1a: Structure model

The left image shows a configuration formed by some patches, where vertices and edges indicate objects and spatial relationships between objects. The middle image shows the features extracted from the left configuration, where vertices and edges indicate HOG features of objects and relative location vectors between objects. The right images shows the structure model we trained, where vertices and edges indicate trained HOG templates of objects and relative location vectors between objects. Our goal is to find the configuration match the trained structure model best in each frame, and update the structure model as well.

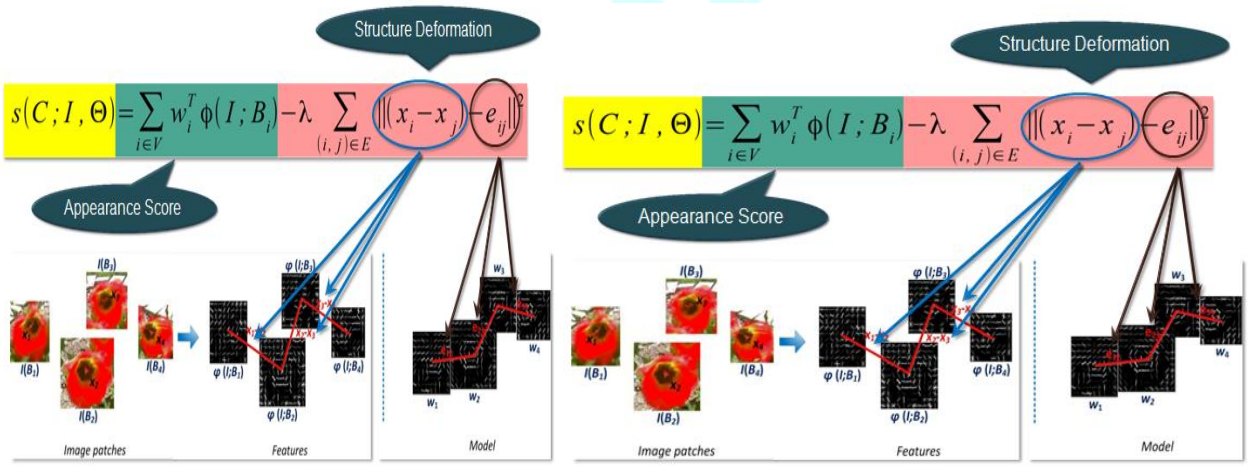

Fig.III.1 b: Appearance and Deformation score

Mathematically, the score of a configuration C is defined as:

$$S(C;I,\Theta) = \sum_{i\in V} W_i^T \emptyset(I;B_i) - \lambda \sum_{(i,j)\in E} ||(x_i - x_j) - e_{ij}||^2 \qquad (1)$$

Herein, the parameters w_i represent linear weights on the HOG features for object i, e_{ij} is the vector that represents the length and direction of the spring between objects i and j, and the set of all parameters is denoted by $\Theta = \{W_1, \ldots, W_{|V|}, \ldots e_1, \ldots, e_{|E|}\}$. We treat the parameter λ as a hyperparameter that determines the trade-off between the appearance and deformation scores. We use Platt scaling to convert the configuration score to a configuration likelihood p(C|I;Θ), Our model is illustrated in Fig 3.1a. In preliminary experiments, we also tried to incorporate (temporal) Gaussian priors on the object locations in the score s(C;I,Θ), but this did not appear to lead to performance improvements: the location prior specified by the spatial constraints in s(C;I,Θ) appears to be sufficient.

2 Inferences

Given the parameters of the model, finding the most likely object configuration amounts to maximizing Eq. (1) over all possible configurations C. This maximization is intractable in general because it requires searching over exponentially many configurations, but for tree-structured graphs G, a combination of dynamic programming and min- convolutions can be used to perform the maximization in linear time.

The dynamic programming equations for maximizing Eq. (1) take the form of a message passing procedure on the tree G. Herein, a message from object i to j is computed by (1) gathering all incoming messages into object i, (2) combining the resulting score with the object detection scores for object i, and (3) transforming the result to incorporate for the stress in the edge between object i and j. The message-passing equations take the form:

$$R_{ij}(x) = W_i^T \emptyset(I;B_i) + \sum_{\forall K\neq j:(K,i)\in E} \mu_{K-Vi}(X), \qquad (2)$$

$$\mu_{i\to j}(X) = \max[R_{ij}(X') - \lambda||(x - x') - e_{ij}||^2], \qquad (3)$$

Where $B_i = \{X_i, w_i, \mathrm{h_i}\}$. The message-passing procedure emanates from an arbitrary root object; after a full forward- backward pass on the pictorial-structures tree, the sum of all incoming messages into the root corresponds to the configuration C^* that maximizes s(C; I, Θ). The configuration C^* can be recovered by backtracking the computations already performed. See [6] for more details.

The inference algorithm can be implemented very efficiently because the negate of Eq. (3) is a two-dimensional min-convolution problem. The one-dimensional counterpart of Eq. (3) can be rewritten in the form:

$$D(p) = min_q[f(q) + (p-q)^2] \qquad (4)$$

where p, q ∈ G with G a one-dimensional grid. The value of D(p) can be computed $\forall_p \in G$ in linear time by looping over the parabolas at all locations p, maintaining a list of coordinates for which the parabolas considered until now have the lowest value. At each step in this loop, the intersection of the current parabola with the last parabola in the list is computed analytically. Based on the location of this intersection, the range of coordinates for which the current parabola has the lowest value can be determined, from which we can decide whether or not the last parabola in the list should be removed or not. The min-convolution algorithm is extended to two dimensions by first running the one dimensional algorithm over all rows of the image, and then running the one-dimensional algorithm on the output of this first stage across all columns.

3. Learning

Like other model-free trackers [8], [9], [23], we use the tracked object configurations in previous frames as positive examples to update our model. After observing an image I, the most likely object configuration C is found by maximizing Eq. (1). We assume that this object configuration C is a true positive example, and therefore, we would like the score at this ground truth location s(C; I,Θ) to be larger than any other configuration $\hat{C}$ by at least a margin$\Delta(C,\hat{C})$. Herein, Δ is a task-loss function that equals zero if $\hat{C} = C$.The task loss is assumed to be non-negative$\forall \hat{C} \neq C: \Delta(C,\hat{C}) > 0$, and upper bounded by a constant M, $\exists M \forall C: max_{\hat{c}} \Delta\left(C,\hat{C}\right) < M$. To adapt our model in such a way that it assigns a higher score to the positive locations and lower scores to other locations, we update the mode(l parameters Θ by taking a gradient step in the direction of the structured SVM loss function . Here, the structured SVM loss l is defined as the maximum violation of the task loss by configuration $\hat{C}$:

$$l(\theta; I, C) = max_{\hat{C}}\left[s\left(\hat{C}; I, \theta\right) - s(C; I, \theta) + \Delta\left(C,\hat{C}\right)\right] \quad (5)$$

Fig III.3: Structured SVM

The structured SVM loss function does not contain quadratic terms, but it is the maximum of a set of affine functions. As a result, the structured SVM loss in Eq. (5) is a convex function in the parameters Θ.

In our SPOT tracker, we define the task-loss $\Delta(C,\hat{C})$ based on the amount of overlap between the correct configuration C and the incorrect configuration $\hat{C}$:

$$\Delta\left(C,\hat{C}\right) = \sum_{i \in V}\left(1 - \frac{B_i \cap \hat{B}_i}{Bi \cup \hat{B}i}\right). \quad (6)$$

Here in, the union and intersection of two bounding boxes Bi and ^Bi are both measured in pixels. If $\hat{C}$ has no overlap with C, the task loss equals |V| , and the task loss equals zero iff $\hat{C}$=C. In preliminary experiments, we also explored task losses that only considered the distance between object centers (and not the size of objects), but we did not find these to improve the performance of our tracker.

The gradient of the structured SVM loss in Eq. (5) with respect to the parameters Θ is given by:

$$\nabla_\Theta\,(\Theta\,; I, C) = \nabla_\Theta\, s(C^*; I, \Theta) - \nabla_\Theta s(C; I.\Theta), \quad (7)$$

in which the "negative" configuration C* is given by:

$$C^* = argmax_{\hat{C}}\left(s\left(\hat{C}; I, \Theta\right) + \Delta\left(C,\hat{C}\right)\right) \quad (8)$$

In practice, this gradient is not very appropriate for learning the parameters of our multi-object tracker because due to the influence of the deformation score, the negative configuration C^* tends to comprise bounding boxes B_i that have very low appearance scores $W_i^T \emptyset(I; B_i)$. As a result, these bounding boxes are inappropriate examples to use as a basis for updating wi. For instance, if we would incorporate the deformation score when identifying a negative example in the Air Show video , we would often select empty sky regions as negative examples. These regions are very uninformative, which hampers the learning of good appearance models for the airplanes under consideration. To address this problem, we ignore the structural constraints when selecting the negative configuration. Specifically, we use a search direction p for learning that is defined as:

$$p = \nabla_\theta\, \tilde{s}\left(\tilde{C}^* \mathrm{I}, \theta\right) - \nabla_\theta\, \mathrm{s}(\mathrm{C}; \mathrm{I}, \theta), \quad (9)$$

where the negative configuration is given by:

$$\tilde{C}^* = argmax_{\hat{C}}\left(\tilde{s}\left(\tilde{C}; I, \theta\right) + \Delta\left(C,\hat{C}\right)\right), \quad (10)$$

with the negative appearance score $\tilde{s}$ being defined as:

$$\tilde{s}\left(\tilde{C}^*; I, \theta\right) = \sum_{i \in V} W_i^T \emptyset(I; Bi), \quad (11)$$

It is straightforward to show that the inner product between the search direction and the true gradient direction remains positive, $p.\nabla_\theta\, l(\theta\,; \mathrm{I}, \mathrm{C}) > 0$, as a result of which the learner will still converge (under suitable conditions on the step size). The configuration $\tilde{C}^*$ can be computed efficiently by (1) adding a term to each object filter response that contains the ratio of overlapping pixels for a bounding box at that location with the detected bounding box to account for the task loss Δ and (2) re-running exactly the same efficient inference procedure as the one that was used to maximize Eq. (1) over configurations.

We use a passive-aggressive algorithm to perform the parameter update. The passive-aggressive algorithm sets the step size in such a way as to substantially decrease the loss, while ensuring that the parameter update is not too large. In particular, the passive-aggressive algorithm uses the following parameter update:

$$\Theta \leftarrow \Theta - \frac{l(\Theta, I, C)}{||p||^2 + \frac{1}{2K}} p, \qquad (12)$$

where $K \in (0, +\infty)$, is a hyper parameter that controls the "aggressiveness" of the parameter update. In preliminary experiments, we also experimented with a confidence-weighted learning algorithm that finds an individual step size for each parameter. However, the performance of this confidence weighted learning algorithm was very similar to that of the passive-aggressive algorithm, which is why we use the passive-aggressive algorithm in the remainder of the paper. When an object is occluded, we do not want to update the appearance model for that object (even if its location was correctly predicted thanks to the spatial constraints in our model). To avoid erroneous updates of our appearance model, we only update a weight vector w_i corresponding to detection $\boldsymbol{B}_i$ when the exponentiated score for that object exceeds some threshold. In particular, we only update the w_i and e_{ij} whenever

$$\frac{1}{Z}\exp\left(W_i^T \emptyset(I; Bi)\right) > 0.4.$$

The weights w_i are initialized by training an SVM that discriminates between the manually annotated object from 50 randomly selected negative examples (extracted from the first frame) that have little to no overlap with the ground-truth annotation. The parameters e_{ij} are initialized based on the ground-truth annotations of the objects: $e_{ij} \leftarrow X_i - X_j$, with X_i and X_j the locations of objects i and j in the first frame.

4 Graph Structure

A remaining issue is how we determine the structure of the graph G, i.e., how we decide on which objects are connected by an edge. Ideally, we would employ a fully connected graph G, but this would make inference intractable.

Hence, we explore two approaches to construct a tree on the objects i ∈V: (1) a star model [6] and (2) a minimum spanning tree model. In the star model, each object is connected by an edge with a dummy object r ∈ V that is always located at the center of all the objects, i.e., there are no direct relations between the objects. This requires a minor adaptation of the score function:

$$s(C; I, \Theta) = \sum_{\substack{i \in V \\ r}} W_i^T \emptyset(I; B_i) - \lambda \sum_{(i,j) \in E} ||(x_i - x_r) - e_i||^2 \qquad (13)$$

The minimum spanning tree model is constructed based on the object annotations in the first frame; it is obtained by searching the set of all possible completely-connected tree models for the tree that minimizes $\sum(i,j) \in E||x_i - x_j||^2$, where x_i and x_j are the locations of objects i and j in the first frame.

5 Computational Complexities

The main computational costs of running our tracker are in the extraction of the HOG features (which are shared between object detectors) and in the computation of the appearance score per object; after these quantities are computed, the maximization of Eq. (1) takes only a few milliseconds. The computational complexity grows linearly in the number of objects being tracked (i.e., in |V|).

IV. EXPERIMENTS

Our proposed algorithm is implemented on "MATLAB 2014a" version, core i5 processor with 1.7 GHZ speed. It is possible to track approximately four objects simultaneously in real-time (in small videos of size 320 × 240 pixels).

We performed two sets of experiments to evaluate the performance of our tracker. In the first set of experiments, we evaluate the performance of the SPOT tracker on a range of multi-object tracking problems, comparing it to the performance of various state-of-the-art trackers that do not employ structural constraints between the objects.

1 Experiment: Multiple-Object Tracking

We first evaluate the performance of the SPOT tracker on videos in which multiple objects need to be tracked.

1.1Setup

We used six videos are downloaded from YouTube with multiple objects in this set of experiments. The videos were selected based on characteristics that are challenging for current model-free trackers, such as the presence of multiple, nearby objects with a very similar appearance. The average length of the videos is 842 frames.

We experiment with three variants of the SPOT tracker: (1) a baseline tracker that does not use structural constraints (i.e., a SPOT tracker with λ=0; no-SPOT), (2) a SPOT tracker that uses a star tree to model spatial relations between objects (star-SPOT), and

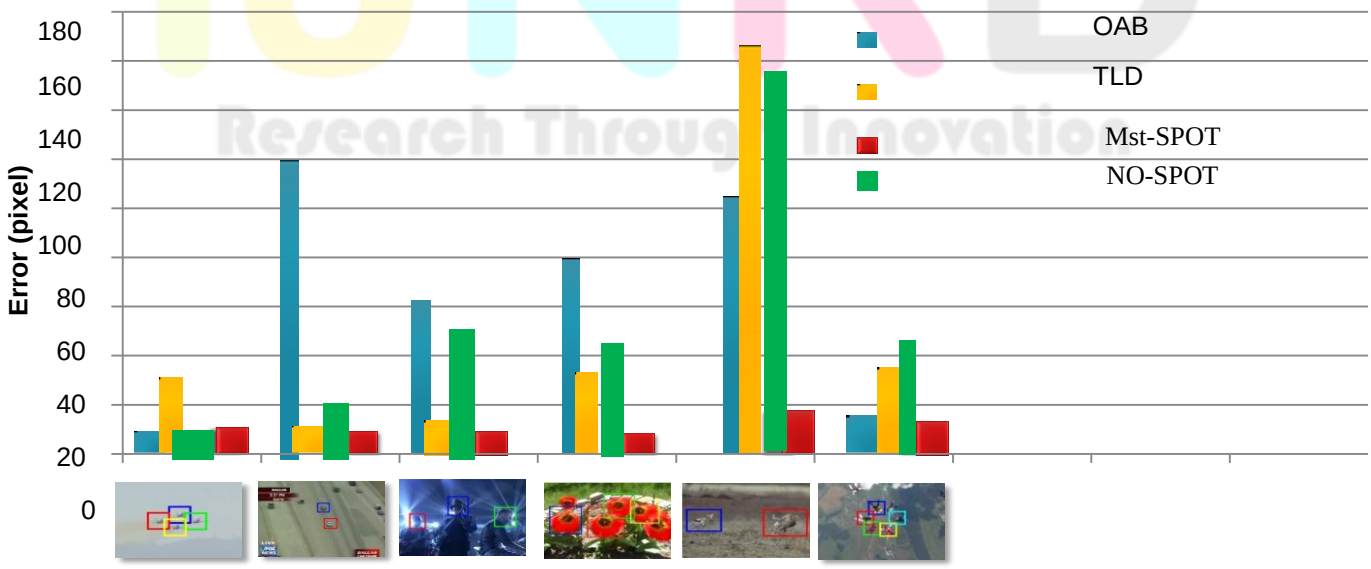

Figure.IV.1.1. Average location error of five multiple-object trackers on all six videos (lower is better). Figure best viewed in color.

(3) A SPOT tracker that uses a minimum spanning tree to model structural constraints (mst-SPOT).
We compare the performance of our SPOT trackers with that of two state-of-the-art (single- object) trackers, viz. the OAB tracker [23] and the TLD tracker [9], of which we run multiple instances to separately track each object. The OAB and TLD trackers were run using the implementations provided by their developers.

Following [8], we evaluate the performance of the trackers by measuring:(1) average location error (ALE): the average distance of the center of the identified bounding box to the center of the ground-truth bounding box. (2) correct detection rate (CDR): the percentage of frames for which the overlap between the identified bounding box and the ground truth bounding box is at least 50 percent.

For each video, these two measurements are averaged over all target objects, and over five separate runs. In all experiments with star-SPOT and mst- SPOT, we fixed λ=0.001 and K=1. In preliminary experiments, we found that the results are very robust under changes of λ and K.

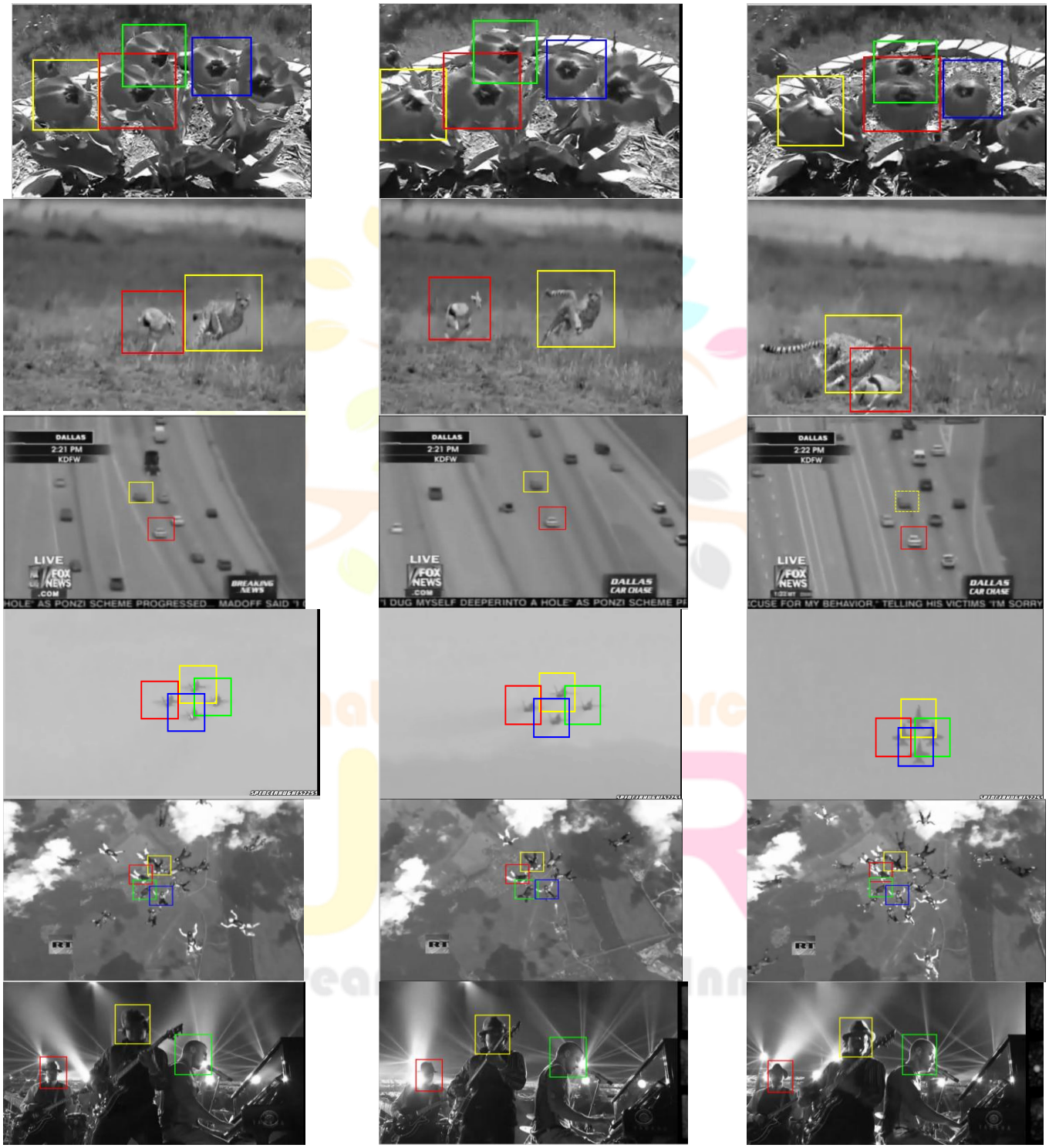

Figure 6.1.2. Tracking results obtained by mst-SPOT on all six videos used in Experiment 1 (from top to bottom: Red Flowers, hunting ,Car Chase, Air Show, Sky Diving and Shaking). The colors of the rectangles indicate the different objects that are tracked.

Red flowers. The video shows several similar flowers that are moving and changing appearance due to the wind, and that sometimes (partially) occlude each other; we track four of these flowers. The baseline trackers lose track very quickly due these partial occlusions. By contrast, the two SPOT trackers flawlessly track all flowers during the entire length of the video (2,249 frames), because they can use the structural constraints to distinguish the different flowers.

Hunting. The cheetah and gazelle in this video clip are very hard to track, because their appearance changes significantly over time and because their relative location is changing (the cheetah passes the gazelle). Nevertheless, the SPOT trackers can exploit the fact that both animals move in a similar direction, which prevents them from losing track.

Car chase. This video is challenging because (1) the two cars are very small and (2) both cars are occluded for around 40 frames, while various other cars are still visible. Whereas this occlusion confuses the baseline trackers, the two SPOT trackers do not lose track because they can use the location of one car to estimate the location of the other.

Taken together, the results presented in Fig. 5 show (1) that our baseline no-SPOT tracker performs on par with state-of-the-art trackers such as TLD; and (2) that the use of spatial constraints between objects leads to substantial performance improvements when tracking multiple objects, in particular, when minimum spanning trees are used (mst-SPOT).

Air Show. The video contains a formation of four visually similar planes that fly very close to each other; the video contains a lot of camera shakes. Whereas our baseline trackers (OAB, TLD, and no-SPOT) confuse theplanes several times during the course of the video,star-SPOT and mst-SPOT track the correct plane throughout the entire video.

Shaking. The video contains a formation of three dancers. They moving and changing appearance due to the shaking. we track three of these dancers using our SPOT tracker.

Sky Diving. The video contains number of people that fly very close to each other; the video contains a lot of camera shakes. We track four girls of these people by using mst-SPOT. It tracks the correct persons throughout the entire video.

The performance improvements are particularly impressive for videos in which objects with a similar appearance are tracked, such as the Car Chase and Red Flowers videos, because the structural constraints prevent the tracker from switching between objects. Structural constraints are also very helpful in videos with a lot of camera shake (such as the Air Show video), because camera shake causes all objects to move in the same direction in the image. The SPOT tracker even outperforms single- object trackers when perceptually different objects are tracked that have a relatively weak relation in terms of their location, such as in the Hunting video, because it can share information between objects to deal with, e.g., motion blur and camera shake. The mst-SPOT tracker outperforms the star SPOT tracker in almost all videos, presumably, because a minimum spanning tree introduces direct (rather than indirect) constraints between the locations of the objects. We also performed experiments with multi-scale versions of our SPOT tracker. The multiple-scale trackers are run at three scales for each frame (viz. relative scales 0.9, 1.0, and 1.1), and select the highest posterior probability among the three scales to determine the location and scale of the object. In all experiments, we assume that the aspect ratio of the rectangular bounding box is fixed.

In particular, we use the (multi-scale) CXT tracker, a multi-scale version of the TLD tracker [9], and a multi-scale version of the no-SPOT tracker as a baseline.2 We compare the performance of these baseline trackers with that of our multi-scale mst-SPOT tracker. On videos such as the Parade and Sky Diving, the CXT and TLD trackers appear to outperform mst-SPOT despite their lack of structural constraints. We surmise this has to do with the size of the objects: the fixed-size of HOG cells (of 8×8 pixels) that SPOT uses in its appearance model may be too large for the small target objects in these two videos. Indeed, the Haar-representations that the CXT and TLD tracker employ may be more suited for tracking such small objects. Comparing the results in fig 6, it can be seen that the difference in performance between single-scale and multi-scale SPOT trackers is rather small, presumably, because most videos exhibit relatively little scale changes over time.

We compare our tracker with that of three state-ofthe-art pedestrian detectors that are used in a tracking-by- detection scenario: (1) a Dalal-Triggs detector (HOG) that was obtained by training a linear SVM on HOG features[11], (2) a Felzenszwalb detector (LatSVM) that was obtained by training a latent SVM on HOG features at two scales [6], and (3) the "fastest pedestrian detector in the wild" (FPDW) that was obtained by Doll_ar. All three detectors were trained on the pedestrian class of the Pascal VOC 2007 challenge. We ran the three detectors using their default parameter values.

CONCLUSION AND FUTURE WORKS

In this paper we have developed a new model-free tracker that tracks multiple objects by combining multiple single-object trackers via constraints on the spatial structure of the objects. Our experimental results show that the resulting SPOT tracker substantially outperforms traditional trackers in settings in which multiple objects need to be tracked. We have also showed that the SPOT tracker can improve the tracking of single objects by

Including additional detectors for object parts. Moreover, the SPOT tracker can be used to adapt a generic detector to a specific object while tracking. The computational costs of our tracker only grow linearly in the number of objects (or object parts) that is being tracked, which facilitates real-time tracking. Of course, the ideas presented in this project may readily be implemented in other model-free trackers that are based on tracking-by-detection, such as the TLD tracker [9]. It is likely that including structural constraints in those trackers will lead to improved tracking performance, too.

In future work, we aim to explore the use of different structural constraints between the tracked objects; for instance, for tracking certain deformable objects it may be better to use a structural model based on PCA (as is done in, e.g., constrained local models). We also plan to develop approaches that identify the relevant parts of a deformable object in a more principled way during tracking via online learning algorithms for latent SVMs.

REFERANCES

[1] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features,"Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 511-518,2001.

[2] M. Isard and J. Maccormick, "Bramble: A Bayesian Multi-BlobTracker,"Proc. IEEE Conf. Computer Vision and Pattern Recognition,pp. 34-41, 2001.

[3] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van, "Online Militiperson Tracking-by-Detection from a Single,Uncalibrated Camera,"IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 33, no. 9, pp. 1820-1833, Sept. 2011.

[4] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll Never Alone: Modeling Social Behavior for Multi Target Tracking,"Proc. IEEE Int'l Conf. Computer Vision, pp. 261-268,2009.

[5] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke, "Coupling Detection and Data Association for Multiple Object Tracking,"Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1948-1955, 2012.

[6] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan,"Object Detection with Discriminatively Trained Part Based Models,"IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 32,no. 9, pp. 1627-1645, Sept. 2010.

[7] V. Lepetit and P. Fua, "Key point Recognition Using Randomized Trees,"IEEE Trans. Pattern Analysis and Machine Intelligence,vol. 28, no. 9, pp. 1456-1479, Sept. 2006.

[8] B. Babenko, M.-H. Yang, and S. Belongie, "Robust Object Tracking with Online Multiple Instance Learning, "IEEE Trans. Pattern Analysis and Machine Intelligence,
vol. 33, no. 8, pp. 1619-1632, Aug.2011.

[9] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-n Learning: Bootstrapping Binary Classifiers by Structural Constraints,"Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 49-56, 2010.

[10] A. Yilmaz, O. Javed, and M. Shah, "Object Tracking: A Survey,"ACM Computing Surveys, vol. 38, no. 4, article 13, 2006.

[11] N. Dalal and B. Triggs, "Histograms of Oriented Gradients forHuman Detection,"
Proc. IEEE Conf. Computer Vision and PatternRecognition, pp. 886-893, 2005.

[12] Z. Kalal, K. Mikolajczyk, and J. Matas, "Face-TLD: Tracking-Learning-Detection Applied to Faces,"Proc. IEEE Conf. Int'l Conf.Image Processing, 2010.

[13] C. Caraffi, T. Vojir, J. Trefny, J. Sochman, and J. Matas, "A System for Real-time Detection and Tracking of Vehicles from a Single Car-Mounted Camera,"Proc. IEEE 15th Int'l Conf. Intelligent Trans-portation Systems (ITS), pp. 975-982, Sept. 2012.

[14] M. Fischler and R. Elschlager, "The Representation and Matching of Pictorial Structures,"IEEE Trans. Computers, vol. C-22, no. 1,pp. 67-92, Jan. 1973.

[15] S. Branson, P. Perona, and S. Belongie, "Strong Supervison from Weak Annotation: Interactive Training of DeformablePart Models,"Proc. IEEE Int'l Conf. Computer Vision,pp.1832-1839, 2011.

[16] Manisha Chate, S.Amudha,Vinaya Gohokar, "Object Detection and tracking in Video Sequences", ACEEE Int. J. on Signal & Image Processing, Vol. 03, No. 01, Jan 2012.

[17] M.J. Black and A.D. Jepson, "Eigentracking: Robust Matching and Tracking of Articulated Objects Using a View- Based Representation," Intl J. Computer Vision, vol. 26, no. 1, pp. 63-84, 1998.

[18] H. Grabner, M. Grabner, and H. Bischof, "Real-Time Tracking via On-Line Boosting," Proc. British Machine Vision Conf., pp. 47-56, 2006.

[19] Y. Wu, J. Lim, and M.-H. Yang, "Online Object Tracking: A Benchmark," Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2013.

[20] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yuang, "Incremental Learning for Robust Visual Tracking," Intl J. Computer Vision, vol. 77, no. 1, pp. 125-141, 2008.

[21] G.J. Edwards, C.J. Taylor, and T.F. Cootes, "Interpreting Face Images Using Active Appearance Models," Proc. IEEE Conf. Automatic Face and Gesture Recognition, pp. 300-305, 1998.

[22] C. Chow and C. Liu, "Approximating Discrete Probability Distributions with Dependence Trees," IEEE Trans. Information Theory, vol. IT-14, no. 3, pp. 462-467, May 1968.

[23] M. Blaschko and C. Lampert, "Learning to Localize Objects with Structured Output Regression," Proc. European Conf. Computer Vision, pp. 2-15, 2008.

[24] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," Proc. 18th Int'l Conf. Machine Learning, pp. 282- 289, 2001.

[25] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large Margin Methods for Structured and Interdependent Output Variables," J. Machine Learning Research, vol. 6, pp. 1453-1484, 2005.

[26] Y. Li and R. Nevatia, "Key Object Driven Multi-Category Object Recognition, Localization and Tracking Using Spatio- Temporal Context," Proc. European Conf. Computer Vision, pp. 409-422, 2008.

[27] S. Stalder, H. Grabner, and L. Van Gool, "Cascaded Confidence Filtering for Improved Tracking-by-Detection," Proc. European Conf. Computer Vision, pp. 369-382, 2010.

[28] P.M. Roth, S. Sternig, H. Grabner, and H. Bischof, "Classifier Grids for Robust Adaptive Object Detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2727-2734, 2009.