Structured Learning for Multiple Object Tracking

Wang Yan
wy109@cs.rutgers.edu
Xiaoye Han
xiaoye@cs.rutgers.edu
Vladimir Pavlovic
vladimir@cs.rutgers.edu

Department of Computer Science
Rutgers, The State University of New Jersey
Piscataway, NJ, USA

Abstract

Many adaptive tracking-by-detection methods have been proposed to track object with slowly changing appearance. However, most of those methods are designed for single object tracking. This paper proposes a method for adaptively tracking multiple objects based on a modified structured Support Vector Machine (SVM). The method utilizes the inter-object constraints and the layout information, which are frequently present in multiple object tracking. Moreover, our approach detects the existences of objects by adding binary constraints in the structured SVM formulation, and therefore can handle frequent occlusions in multiple object tracking. In contrast, the original structured SVM assumes continual existence of the tracked object, making it susceptible to drift. Experimental results show the proposed method works better than existing adaptive tracking-by-detection method, as well as non-adaptive association-based multiple object tracking approach.

1 Introduction

Object tracking is of broad interest and has been broadly investigated. Many successful tracking methods use a static template to model the object appearance [7, 12]. This approach is reasonable when the training set is available and covers wide variations of the object appearances in the tracking task. However, there are many cases where the training set is not available as a prior, e.g. when tracking arbitrary objects and/or when the object appearance undergoes changes beyond the training set [13, 19, 20]. In such instances tracking methods with static templates perform poorly [3].

The key to overcome the difficulty is to adaptively update the appearance model during tracking, c.f., [3, 9, 10, 13, 16, 18, 19, 20, 21, 23], which will be referred as adaptive tracking-by-detection methods in this paper. These methods use previous tracking results to generate a new training set for object appearance, and update the current model to predict the object location in subsequent frames. However, most adaptive tracking-by-detection methods focus on single object or multiple unrelated objects. Although one can trivially engage several single object trackers to track multiple objects, such solution is frequently suboptimal because it does not utilize the inter-object constraints or the obejct layout information [14].

http://dx.doi.org/10.5244/C.26.48

This paper proposes an adaptive tracking-by-detection method for multiple object, inspired by recent works in [10] and [14]. The constraints for Structured Support Vector Machine (SVM) in [10] are modified to localize multiple objects simultaneously with both appearance and layout information. Moreover, additional binary constraints are introduced to detect the existences of respective objects and to prevent possible model drift. Thus the method can handle frequent occlusion in multiple object tracking, as well as objects entering or leaving the scene. The inter-object constraints are applied to diminish false detections, and are embedded in a linear programming technique for optimal position assignment. Experiments show this method works better than other adaptive single object tracker if tracking multiple objects. Moreover, it also outperforms non-adaptive association-based multiple object tracking method when tracking objects without enough training samples.

The rest of the paper is organized as follows. Sec. 2 discusses the works related to the proposed method, whose baseline version for single object is described in Sec. 3, and the multiple object version is described in Sec. 4. Sec. 5 presents the experimental results and Sec. 6 concludes the paper.

2 Related Work

This section reviews two categories of methods which are related to the proposed method, i.e. adaptive tracking-by-detection methods for single object tracking and non-adaptive association-based methods for multiple object tracking.

Adaptive tracking-by-detection method first learns initial appearance model from the first one or several labeled frames, and then tracks the object in the following frames. Once the new tracking result is available, it's used to updates the current appearance model. These methods can track objects with slowly changing appearance without any complete training set. However, if the tracker location is not precise, the updating mechanism will degrade the model by adding false positive samples. As a consequence, the tracker is very likely to drift over time. There are two ways to overcome this problem. One is to improve the model robustness to outlier samples. [9] formulates adaptive tracking-by-detection as a semi-supervised learning problem. It only uses the reliable samples from the first frame as labeled, and considers the one from other frames as unlabeled. [16, 18] incorporate more robust loos functions in boosting. Multiple Instance Learning (MIL) is used by [3, 23] to automatically elicit the best positive sample during training. The other way is to improve the tracker accuracy to suppress drift as much as possible. Inspired by successful application of structured SVM to object localization [5], [10] proposes to use it for adaptive tracking. It is important to note that the structured SVM is not a classifier, and more closely resembles a regressor on the continuous location output.

The most significant difference between the proposed method and the existing adaptive tracking-by-detection methods is the former explicitly models the multiple object tracking with additional constraints and information, i.e., two or more objects cannot be too close to each other unless occluded, and the layout of objects should be consistent between consecutive frames. Although [3] claims that their method could be possibly applied to multiple objects it leaves this extension as future work. In addition, most adaptive tracking-by-detection methods other than [10] use boosting as the classifier, while the proposed method employs a modified structured SVM which works like a combination of regressor and classifier. While [10] is closely related to our proposed method, because both use the structured SVM, two important differences exist. First, the proposed method includes binary constraints for object

existence, while [10] assumes continual existence of one and only one object in each frame. Second, [10] uses location with the maximum response as the tracking result, while the proposed method uses linear programming for multiple object location assignment because of the inter-object and layout constraints.

In the context of multiple object tracking, association-based methods play a prominent role. They typically construct multi-object graphs by first detecting candidate object locations in each frame, using graph nodes to represent candidate 2D [11, 14, 15, 24] or 3D [2, 4] object locations in different frames or candidate tracklets [2, 11] over sets of frames, and assigning weights to nodes [2] and/or edges [14] based on object appearance and/or layout. Disjoint paths corresponding to different objects are then found by searching for maximal subgraphs. Linear programming [2, 4, 14], Hungarian algorithm [11, 15] and min-cost flow algorithm [24] are used to solve the optimization problem. The proposed method differs from association-based method by adaptive learning the appearance model during tracking. Although [15] also learns to update the similarity model, it focuses on adaptation of edge weights while keeps the appearance model (detector) unchanged. In contrast, our approach adapts the appearance models of all tracked objects.

3 Online Structured Tracking-Detection

3.1 Online structured tracking

This subsection will briefly review the application of structured SVM in tracking [10].

Given a set of frames $\{x_1, x_2, \ldots, x_n\}$ indexed by time, and $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$ is the corresponding set of labeling, i.e. bounding boxes. The existence of the object within the frame is always assumed. Structured SVM tries to find a model $f(x, \mathbf{y})$, such that the task of predicting object location in a testing frame x could be conquered by $\arg\max_{\mathbf{y}} f(x, \mathbf{y})$. It's general to assume that f is an inner product in some implicit feature space, i.e.

$$f(x, \mathbf{y}) = \langle \mathbf{w}, \Psi(x|_{\mathbf{y}}) \rangle, \tag{1}$$

where $x|_{\mathbf{y}}$ is the patch of frame x within bounding box $\mathbf{y}$ or the features extract from it, and $\Psi(\cdot)$ is the mapping from input space to the implicit features space. Therefore, a reasonable model could be obtained through the following optimization problem.

$$\min_{\mathbf{w}, \xi} \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i=1}^{n} \xi_i \qquad \text{s.t.} \tag{2}$$

$$\langle \mathbf{w}, \Psi(x_i|_{\mathbf{y}_i}) - \Psi(x_i|_{\mathbf{y}}) \rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i \quad , \quad i = 1, 2, \ldots, n, \quad \mathbf{y} \neq \mathbf{y}_i \tag{3}$$

$$\xi_i \geq 0 \quad , \quad i = 1, 2, \ldots, n, \tag{4}$$

where $\mathbf{y} \neq \mathbf{y}_i$ implies bounding box $\mathbf{y}$ in (3) could be anywhere else other than groundtruth $\mathbf{y}$, and

$$\Delta(\mathbf{y}_i, \mathbf{y}) = 1 - \frac{\mathbf{y}_i \cap \mathbf{y}}{\mathbf{y}_i \cup \mathbf{y}} \tag{5}$$

is the loss of predicting $\mathbf{y}$ when groundtruth is $\mathbf{y}_i$. (3) means that model response on shifted locations $\mathbf{y}$ should be less than the one on groundtruth location $\mathbf{y}_i$ by at least $\Delta(\mathbf{y}_i, \mathbf{y})$, with the possibility of few violations encoded by errors ξ_i.

Since the feasible space of $\mathbf{y}$ could be very large, the optimization (2) with the large number of constraints (3) can be hard to solve directly. As suggested in [22], one could start

with a few constraints, and then iteratively add the most violated and unselected constraints from each frame for re-training. According to (3), the most violated constraint for frame x_i is the constraint corresponding to the solution of

$$\arg\max_{\mathbf{y}\neq\mathbf{y}_i} \Delta(\mathbf{y}_i,\mathbf{y}) + \langle \mathbf{w}, \Psi(x_i|_{\mathbf{y}})\rangle. \tag{6}$$

[10] adaptively updates the model with previous tracking result, and then predict the current object location. It can track slowly changing object due to its adaptive nature, but it is also likely to drift when the object is occluded or out of the scene. As illustrated in Fig. 1(b), the tracker still tries to track the male's face even when it is occluded by the female in the second frame. Then the ongoing adaptation quickly adapts the tracker to the female's face, leading to model drift. In order to address this problem, one has to know whether the object exists in order to start or stop the adaptation accordingly.

(a) With binary constraint.

(b) Without binary constraint.

Figure 1: Single object tracking with structured SVM. The videos are included in the supplementary material.

3.2 Online structured tracking-detection

This subsection proposes to solve the existence problem by adding binary constraints.

Suppose Z is the training set of the object detector, and each $\mathbf{z} \in Z$ has the label $l_{\mathbf{z}} \in \{+1,-1\}$. Training samples could be collected from previous frames and/or from other independent sources. In this paper we only use the former, i.e., the patch from groundtruth bounding box $x_i|_{\mathbf{y}_i}$ is used as positive sample, and the patches $x_i|_{\mathbf{y}}$ from other bounding boxes $\mathbf{y}$, whose overlaps $\frac{\mathbf{y}_i \cap \mathbf{y}}{\mathbf{y}_i \cup \mathbf{y}}$ with the groundtruth $\mathbf{y}_i$ are less than a threshold, are used as negative samples. Note that, particularly in the case of multiple tracked objects, the negative examples can include bounding boxes of other objects. This can help additionally discriminate different objects during tracking.

For each $\mathbf{z} \in Z$, the binary constraint and corresponding slack variable constraint are

$$l_{\mathbf{z}}(\langle \mathbf{w}, \Psi(\mathbf{z})\rangle + b) \geq 1 - \eta_{\mathbf{z}} \quad , \quad \forall \mathbf{z} \in Z, \tag{7}$$

$$\eta_{\mathbf{z}} \geq 0 \quad , \quad \forall \mathbf{z} \in Z, \tag{8}$$

where b is the bias. Due to the new slack variables, the overall objective function (2) becomes

$$\min_{\mathbf{w},b,\xi,\eta} \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i=1}^{n} \xi_i + C_2 \sum_{\mathbf{z}\in Z} \eta_{\mathbf{z}}. \tag{9}$$

Objective function (9) and constraints (3)(4)(7)(8) lead to a new optimization problem, which could be recognized as a combination of structured SVM and binary SVM. If there is an object within the testing frame, it is reasonable to say that the model response at the object location is probably a local extrema according to the structured constraints (3), and the response itself will probably be positive according to the binary constraints (7). If there is no object, all responses will probably be negative. Therefore the following prediction procedure is justified.

$$\text{Prediction} = \begin{cases} \mathbf{y}^* & \text{if } f(x,\mathbf{y}^*) > 0 \\ \text{no object} & \text{otherwise} \end{cases}, \qquad \text{where } \mathbf{y}^* = \arg\max_{\mathbf{y}} f(x,\mathbf{y}). \tag{10}$$

Similar to the last subsection, the new optimization problem is also solved iteratively by recursively adding the most violated constraints. The structured constraint is again selected according to (6), while the following equation derived from (7) indicates the most violated binary constraint:

$$\arg\max_{\mathbf{z}\in Z} 1 - l_{\mathbf{z}}(\langle \mathbf{w}, \Psi(\mathbf{z})\rangle + b). \tag{11}$$

Just as [10], the new model could also be used for adaptive tracking. When an object is detected at the tracker location, the tracking result is added into the training set, and the model is updated by adding new structured and binary constraints. When there is no object detected, the tracker stops adaptation. Since this may be a miss-detection, and the object patch may be used as a negative sample if performing update, it is safer to exclude this frame from the training set. Fig. 1(a) shows the proposed tracker knows its target, i.e., the male face is not presented in the second frame, and does not output the bounding box nor update the model. When his face reappears, the tracker restarts tracking and adaptation. Comparing the results of the two models in Fig. 1(a) and 1(b), one can concluded that adding binary constraints is important for selective adaptation and suppression of drifting.

Using standard Lagrangian duality techniques and the same reparametrising as [6, 10], the new optimization problem could be converted into its dual form as follows.

$$\max_{\gamma,\beta} -\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}\sum_{\mathbf{y}} \gamma_{i\mathbf{y}} \Delta(\mathbf{y}_i,\mathbf{y}) + \sum_{\mathbf{z}\in Z} \beta_{\mathbf{z}} \qquad \text{s.t.} \tag{12}$$

$$\gamma_{i\mathbf{y}_i} \le C_1 \quad , \quad i = 1,2,\ldots,n, \tag{13}$$

$$\gamma_{i\mathbf{y}} \le 0 \quad , \quad i = 1,2,\ldots,n, \quad \forall \mathbf{y} \ne \mathbf{y}_i, \tag{14}$$

$$\sum_{\mathbf{y}} \gamma_{i\mathbf{y}} = 0 \quad , \quad i = 1,2,\ldots,n, \tag{15}$$

$$0 \le \beta_{\mathbf{z}} \le C_2 \quad , \quad \forall \mathbf{z} \in Z, \tag{16}$$

$$\sum_{\mathbf{z}\in Z} \beta_{\mathbf{z}} l_{\mathbf{z}} = 0, \tag{17}$$

$$\mathbf{w} = \sum_{i=1}^{n}\sum_{\mathbf{y}} \gamma_{i\mathbf{y}} \Psi(x_i|_{\mathbf{y}}) + \sum_{\mathbf{z}\in Z} \beta_{\mathbf{z}} l_{\mathbf{z}} \Psi(\mathbf{z}). \tag{18}$$

(18) indicates that $\|\mathbf{w}\|^2$ is represented by the linear combinations of inner products $\langle \Psi(\cdot), \Psi(\cdot)\rangle$, which could be easily computed by the kernel function without knowing $\Psi(\cdot)$. Moreover, the

combination coefficients are quadratic functions of γ and β. Thus the dual form is a standard quadratic programming problem with respect to γ and β.

3.3 Discussion

An alternative approach to the one proposed would be to consider the absence of the object as a virtual location $\mathbf{y} = null$, and use the original structured SVM to detect its existence and to predict its location (if exists) in one step by maximizing the model. However, this requires the definition of the mapping $\Psi(x|_{\mathbf{y}=null})$ or of the corresponding kernel $k(\cdot, \mathbf{y} = null)$, which are both hard to define. [5] indeed suggests a simple definition $k(\cdot, \mathbf{y} = null) = 0$, which is equivalently $\Psi(x|_{\mathbf{y}=null}) = \mathbf{0}$, and also defines loss function $\Delta(\mathbf{y} \neq null, \mathbf{y} = null) = 1$. However, they do not consider the absence case in their subsequent derivations. Using the same definition, it is easy to show that the structured constraint (3) becomes $\langle \mathbf{w}, \Psi(x_i|_{\mathbf{y}_i}) \rangle - 0 \geq 1 - \xi_i$ when groundtruth $\mathbf{y}_i \neq null$, and $0 - \langle \mathbf{w}, \Psi(x_j|_{\mathbf{y}}) \rangle \geq 1 - \xi_j$ otherwise. Obviously, those constraints are respectively special cases of (7) for positive and negative samples, when forcing bias $b = 0$. Therefore, adding binary constraints is a more general formulation. Moreover, the second type of constraint only appears when the object is not in the frame, which is a relatively rare case. In contrast, the proposed formulation is more flexible in that it generates more negative constraints, which are important to suppress the false alarm rate.

Note that using binary constraints alone is not sufficient. The localization accuracy of a binary detector is inadequate, compromising the adaptive tracker's performance [10].

4 Multiple Object Adaptive SVM Tracker

This section extends the model proposed in Sec. 3.2 to the multiple object case. Compared with the single object version, it utilizes the constraint that two or more objects can not appear in the same location in one frame, as well as the objects layout information.

The training set of the multiple object tracking-detection model includes a frame set $\{x_1, x_2, \ldots, x_n\}$ indexed by time and the corresponding set of structured labels $\{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n\}$, where $\mathbf{Y}_i = (\mathbf{y}_i^{(1)}, \mathbf{y}_i^{(2)}, \ldots, \mathbf{y}_i^{(K)})$ indicates the bounding boxes corresponding to K objects in frame i. If the k-th object does not appear in the i-th frame, $\mathbf{y}_i^{(k)} = null$. For the sake of clarity, the formulation will be introduced assuming all objects are present and the general case will be discussed afterwards.

The task is to design a function $f(x, \mathbf{Y})$ such that the object locations $\mathbf{Y}^*$ in frame x are given by $\arg\max_{\mathbf{Y}} f(x, \mathbf{Y})$. A reasonable assumption is

$$f(x, \mathbf{Y}) = \sum_{k=1}^{K} \langle \mathbf{w}^{(k)}, \Psi(x|_{\mathbf{y}^{(k)}}) \rangle + \langle \mathbf{v}, \Phi(\mathbf{Y}; \mathbf{Y}_{i-1}) \rangle, \tag{19}$$

where $\mathbf{Y}_{i-1}$ is the layout in previous $i-1$-th frame and $\Phi(\mathbf{Y}; \mathbf{Y}_{i-1})$ is the layout feature of size $\binom{K}{2} \times 2$, whose k-l-j-th element is

$$\Phi_{klj}(\mathbf{Y}; \mathbf{Y}_{i-1}) = \begin{cases} \left| \left(\mathbf{y}_{i-1}^{(k)}(j) - \mathbf{y}_{i-1}^{(l)}(j) \right) - \left(\mathbf{y}^{(k)}(j) - \mathbf{y}^{(l)}(j) \right) \right| & \text{if } \mathbf{y}_{i-1}^{(k)}, \mathbf{y}_{i-1}^{(l)}, \mathbf{y}^{(k)}, \mathbf{y}^{(l)} \neq null \\ 0 & \text{otherwise} \end{cases}, \tag{20}$$

while $\mathbf{y}(1)$ and $\mathbf{y}(2)$ are the horizontal and vertical coordinates of the bounding box $\mathbf{y}$'s center, respectively. Similar to Sec. 3.2, the model leads the following optimization.

$$\min_{\mathbf{w},\mathbf{v},\xi,\eta} \frac{1}{2}\left(\sum_{k=1}^{K}\|\mathbf{w}^{(k)}\|^2+\|\mathbf{v}\|^2\right)+C_1\sum_{i=2}^{n}\xi_i+C_2\sum_{k=1}^{K}\sum_{\mathbf{z}\in Z}\eta_{\mathbf{z}} \quad \text{s.t.} \tag{21}$$

$$\sum_{k=1}^{K}\langle\mathbf{w}^{(k)},\Psi(x_i|_{\mathbf{y}_i^{(k)}})-\Psi(x_i|_{\mathbf{y}^{(k)}})\rangle+\langle\mathbf{v},\Phi(\mathbf{Y}_i;\mathbf{Y}_{i-1})-\Phi(\mathbf{Y};\mathbf{Y}_{i-1})\rangle\geq\Delta^M(\mathbf{Y}_i,\mathbf{Y})-\xi_i,\forall i,\mathbf{Y}\neq\mathbf{Y}_i, \tag{22}$$

$$l_{\mathbf{z}^{(k)}}(\langle\mathbf{w}^{(k)},\Psi(\mathbf{z}^{(k)})\rangle+b^{(k)})\geq 1-\eta_{\mathbf{z}^{(k)}} \quad , \quad \forall k, \quad \forall\mathbf{z}^{(k)}\in Z^{(k)}, \tag{23}$$

$$\xi_i\leq 0 \quad , \quad \forall i, \tag{24}$$

$$\eta_{\mathbf{z}^{(k)}}\leq 0 \quad , \quad \forall k, \quad \forall\mathbf{z}^{(k)}\in Z^{(k)}. \tag{25}$$

(22) is the structured constraint, where $\mathbf{Y}_i$ is the groundtruth object location set for frame i, $\mathbf{Y}$ is the set of locations other than groundtruth, and

$$\Delta^M(\mathbf{Y}_i,\mathbf{Y})=\sum_{k=1}^{K}\Delta(\mathbf{y}_i^{(k)},\mathbf{y}^{(k)}) \tag{26}$$

is a combination of losses defined by (5) on each objects. (23) is the binary constraint, where $Z^{(k)}$ is the binary detector training set for k-th object. The optimization problem (21)-(25) is similar to the combination of K models proposed in Sec. 3.2, but the key differences are the additional model part $\mathbf{v}$ for inter-object layout, and one structured constraint for all objects rather than K. The dual form of the optimization problem could be easily deviated and shown to be a quadratic programming problem on the inner product feature space.

4.1 Training and predicting

Like other models with many constraints, the new model is also trained iteratively with newly added constraints. Searching for the most violated binary constraint is the same as in the previous models, but searching for the structured counterpart is more difficult due to the spatial occlusion constraint, i.e. two or more objects cannot occupy the same location. Therefore the search should be done for all objects jointly. According (22), the constraint selection problem is

$$\arg\max_{\mathbf{Y}\neq\mathbf{Y}_i}\Delta^M(\mathbf{y}_i,\mathbf{y})+\sum_{k=1}^{K}\langle\mathbf{w}^{(k)},\Psi(x_i|_{\mathbf{y}^{(k)}})\rangle+\langle\mathbf{v},\Phi(\mathbf{Y};\mathbf{Y}_{i-1})\rangle. \tag{27}$$

A series of binary indicator variable $e_{\mathbf{y}^{(k)}}\in\{0,1\}$ can be defined, i.e. $e_{\mathbf{y}^{(k)}}=1$ iff k-th object is at location $\mathbf{y}^{(k)}$. With those variables, (20) and using v_{klj} to represent the k-l-j-th element in $\mathbf{v}$, (27) is equivalent to

$$\arg\max_{\mathbf{E}}\sum_{k=1}^{K}\sum_{\mathbf{y}^{(k)}}e_{\mathbf{y}^{(k)}}(\Delta(\mathbf{y}_i^{(k)},\mathbf{y}^{(k)})+\langle\mathbf{w}^{(k)},\Psi(x_i|_{\mathbf{y}^{(k)}})\rangle)+$$
$$\sum_{k=1}^{K}\sum_{l=k+1}^{K}\sum_{j=1}^{2}v_{klj}\left|\left(\mathbf{y}_{i-1}^{(k)}(j)-\mathbf{y}_{i-1}^{(l)}(j)\right)-\left(\sum_{\mathbf{y}^{(k)}}e_{\mathbf{y}^{(k)}}\mathbf{y}^{(k)}(j)-\sum_{\mathbf{y}^{(l)}}e_{\mathbf{y}^{(l)}}\mathbf{y}^{(l)}(j)\right)\right| \quad \text{s.t.} \tag{28}$$

$$e_{\mathbf{y}^{(k)}} \in \{0,1\} \quad , \quad k = 1,2,\ldots,K, \quad \forall \mathbf{y}^{(k)}, \tag{29}$$

$$\sum_{\mathbf{y}^{(k)}} e_{\mathbf{y}^{(k)}} = 1 \quad , \quad k = 1,2,\ldots,K, \tag{30}$$

$$e_{\mathbf{y}^{(k)}} + \sum_{D(\mathbf{y}^{(l)},\mathbf{y}^{(k)})\leq\delta} e_{\mathbf{y}^{(l)}} \leq 1 \quad , \quad 1 \leq k < l \leq K, \tag{31}$$

$$\sum_{k=1}^{K} e_{\mathbf{y}_i^{(k)}} \leq K-1. \tag{32}$$

(30) implies that one object should only be assigned to one location, and (32) excludes the solution $\mathbf{Y} = \mathbf{Y}_i$. The most important occlusion constraint (31) indicates that if two locations are too close to each other according to some distance $D(\cdot,\cdot)$ and threshold δ, at most one of them is occupied by object. If the binary constraint (29) is relaxed to continuous variable between $[0,1]$, and the absolute value term is eliminated by auxiliary variables [14], the problem becomes a linear programming problem, which is a relaxation of the corresponding integer problem. The solutions are integer almost all the time, with no adverse empirical effects of the infrequent non-integer solutions. Note that the maximization problem in prediction is slightly different from the one in training, i.e. there is no $\Delta^M(\mathbf{y}_i,\mathbf{y})$ in the objective function and no $\mathbf{y} \neq \mathbf{y}_i$ constraint. However, the same technique could also applied with trivial changes.

4.2 Working with missing objects

Suppose there is a frame in the training set with only $L < K$ objects present. In this case, only constraints on part of the model components, i.e some components in $\mathbf{v}$, L $\mathbf{w}^{(k)}$s and L $b^{(k)}$s corresponding to the existing objects, will be used in the optimization. Accordingly, only the same part of model components will be used to find the most violated constraints. This is effectively the model degraded to L objects.

The general prediction consists of three steps. First, find all K locations by assuming all objects are present. Second, exclude the objects with negative responses to the corresponding binary classifiers as they are not present. Third, find the best locations for the remaining objects with corresponding model components. Although the rationale behind the prediction procedure is not directly justified by the problem formulation, empirical evidence suggests it as a reasonable choice. Object detectors could be used to remove locations with negative responses to further reduce the search space.

5 Experimental Result

The proposed multiple object tracking method is compared against state-of-the-art methods, including adaptive single object tracking-by-detection methods Struck [10] and Frag [1], as well as a non-adaptive multiple object tracking method (Huang's method) which is popular in a few papers [11, 15, 24].

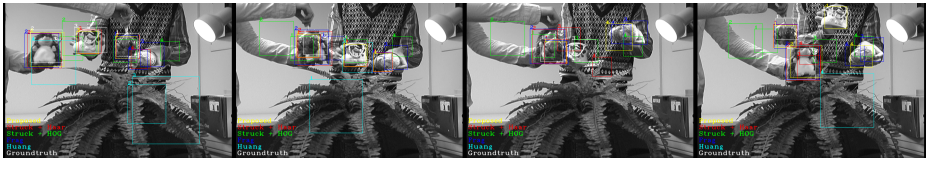
The proposed method uses standard Histogram of Oriented Gradient (HOG) as features from [8] and linear kernels. Struck and Frag use their default Haar features, respectively. In addition, Struck with the same HOG feature is also evaluated. Huang's method uses Haar feature, Local Binary Pattern (LBP) and HOG features for face detection in the first video, object detection in the second video and data association in both videos, respectively.

Method	Man in White	Man in Blue	Man in Black	Average
Proposed	0.7080	0.6699	0.7535	0.7105
Struck + Haar	0.3836	0.2222	0.0434	0.2164
Struck + HOG	0.3466	0.2096	0.0217	0.1926
Frag	0.0269	0.0014	0.0004	0.0096
Huang	0.6516	0.5805	0.7148	0.6490

Table 1: Normalized overlap on "motinas-multi-face-fast".

All methods are evaluated on two videos. In the first video "motinas-multi-face-fast" [17], three males move quickly and occlude each other frequently. We take the second video of four objects with DV. Two people move the objects around with clutter under a lamp, which aims to strengthen the lighting change. Fig. 2(a) and 2(b) show the tracking results of various methods on two videos, respectively.

(a) Tracking result on "motinas-multi-face-fast".



(b) Tracking result on our video.

Figure 2: Tracking result. The videos are included in the supplementary material.

Pascal VOC criterion is used to evaluate the performance, i.e. the normalized overlap between predicted position and the groundtruth. Table 1 & 2 show the results on two videos, respectively. In most of the cases, the proposed method significantly outperforms other adaptive single object methods, which quickly get adapted to other wrong image patches. The only exception is the Struck result of the candy bag, since the bag has never been occluded. However it is really a seldom case in multiple object tracking, where occlusion happens frequently. On the face video, a well-trained detector is used to generate the face candidates, and then Huang's method is applied to generate the face tracks. The figures in Table 1 indicate our method is comparable with Huang's method when a good detector is available. However for our video, neither enough training samples nor trained detectors are available. In order to apply Hunag's method, we use OpenCV to train four boosting detectors respectively with the first two frames and several background images. Due to the very limited positive samples, i.e. 2 for each object, it's very hard if not impossible to obtain any workable detector. As a consequence, Huang's method performs poorly on the second video. In contrast, our method starts only with the first two frames and labelings, but leads to a much better result because of its adaptive nature. Please note that our method doesn't even use the background images.

Method	Candy Bag	Hedgehog	China Boot	Easter egg	Average
Proposed	0.6809	0.7354	0.6333	0.5493	0.6497
Struck + Haar	0.7516	0.6555	0.4488	0.3387	0.5486
Struck + HOG	0.1562	0.0377	0.1520	0.1899	0.1339
Frag	0.5764	0.5552	0.0816	0.5300	0.4358
Huang	0.0242	0.1223	0.1480	0.0164	0.0777

Table 2: Normalized overlap on our video.

6 Conclusion

In this paper, we proposed a method for adaptively tracking multiple objects based on a modified structured SVM, which utilizes rich inter-object constraints and the layout information in multiple object tracking. Moreover, by adding binary constraints, the method detects the existence of the object and handles frequent occlusions well. The experiments verified its effectiveness compared to other state-of-the-art methods.

Acknowledgement

This work is supported in part by the National Science Foundation under Grant No. IIS 0916812.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[2] A. Andriyenko and K. Schindler. Globally optimal multi-target tracking on a hexagonal lattice. In *European Conference on Computer Vision*, 2010.

[3] B. Babenko, M. Yang, and B. Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, 2011.

[4] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.

[5] M. Blaschko and C. Lampert. Learning to localize objects with structured output regression. In *European Conference on Computer Vision*, 2008.

[6] A. Bordes, L. Bottou, P. Gallinari, and J. Weston. Solving multiclass support vector machines with larank. In *International Conference on Machine Learning*, 2007.

[7] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[8] P. Dollár. Piotr's Image and Video Matlab Toolbox (PMT). http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html.

[9] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *European Conference on Computer Vision*, 2008.

[10] S. Hare, A. Saffari, and P.H.S. Torr. Struck: Structured output tracking with kernels. In *IEEE International Conference on Computer Vision*, 2011.

[11] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *European Conference on Computer Vision*, 2008.

[12] Michael Isard and Andrew Blake. CONDENSATION—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[13] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 (10):1296–1311, 2003.

[14] H. Jiang, F. Fels, and J. Little. A linear programming approach for multiple object tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[15] C. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[16] C. Leistner, A. Saffari, P.M. Roth, and H. Bischof. On robustness of on-line boosting - a competitive study. In *IEEE International Conference on Computer Vision Workshops*, 2009.

[17] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro. Particle PHD filter for multi-target visual tracking. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.

[18] H. Masnadi-Shirazi, V. Mahadevan, and N. Vasconcelos. On the design of robust classifiers for computer vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[19] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):810–815, 2003.

[20] D.A. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

[21] A. Saffari, C. Leistner, M. Godec, and H. Bischof. Robust multi-view boosting with priors. In *European Conference on Computer Vision*, 2010.

[22] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning*, 2004.

[23] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof. On-line semi-supervised multiple-instance boosting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[24] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.