Group Tracking: Exploring Mutual Relations for Multiple Object Tracking

Genquan Duan[1], Haizhou Ai[1], Song Cao[1], and Shihong Lao[2]

[1] Computer Science & Technology Department, Tsinghua University, Beijing, China
dgq08@mails.tsinghua.edu.cn, ahz@mail.tsinghua.edu.cn
[2] Development Center, OMRON Social Solutions Co., Ltd., Kyoto, Japan
lao@ari.ncl.omron.co.jp

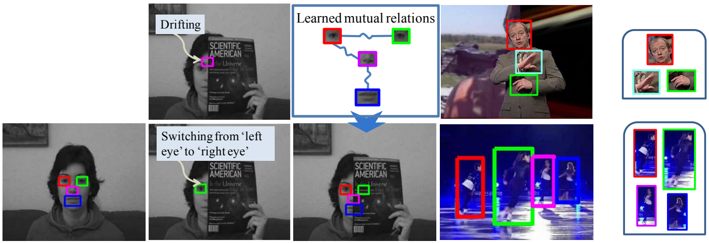
Abstract. In this paper, we propose to track multiple previously unseen objects in unconstrained scenes. Instead of considering objects individually, we model objects in mutual context with each other to benefit robust and accurate tracking. We introduce a unified framework to combine both Individual Object Models (IOMs) and Mutual Relation Models (MRMs). The MRMs consist of three components, the relational graph to indicate related objects, the mutual relation vectors calculated within related objects to show the interactions, and the relational weights to balance all interactions and IOMs. As MRMs are varying along temporal sequences, we propose online algorithms to make MRMs adapt to current situations. We update relational graphs through analyzing object trajectories and cast the relational weight learning task as an online latent SVM problem. Extensive experiments on challenging real world video sequences demonstrate the efficiency and effectiveness of our framework.

1 Introduction

Visual object tracking is a major component in computer vision and widely applied in many domains, such as video surveillance, driving assistant systems, human computer interactions, etc. As an object to be tracked is unknown in many applications, many researchers adopt online algorithms [1–3] to extract the object from background, which mainly encounters the great challenges from occlusions, changing appearances, varying illumination and abrupt motions. Recently, some authors utilized the context (e.g. feature points [4] and regions in similar appearances to the target [5]) for robust single object tracking.

Problem Statement: In this paper, we propose to track multiple previously unseen objects. A direct solution for this problem is to consider objects individually and utilize some approaches designed for single object tracking. Such a solution relies heavily on *individual object models* (IOMs). Once IOMs become inaccurate, the targets tend to be lost easily. In fact, the mutual relations among objects are another important kind of information and worth taking into account. If we have obtained reliable *mutual relation models* (MRMs) at a given time, we can use them to predict more accurate locations of objects as shown in

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part III, LNCS 7574, pp. 129–143, 2012.



(a) A simple scene (b) Typical errors of IOMs (c) Accurate results of IOMs+MRMs (d) More challenging scenes (e) How to learn robust mutual relations?

Fig. 1. Examples of mutual relations. (a,b,c) demonstrate a toy example, where mutual relation models (MRMs) can benefit robust tracking when some individual object models (IOMs) are inaccurate (For the scene in (a), the reason is occlusions). The switching error in (b) is due to similar appearances of 'right eye'and 'left eye'. As all objects have rigid-like relations and they come from the same object, robust MRMs can be easily learned for this toy. However, they are difficult to be learned in many other scenes, such as in (d)(top) illustrating objects with abrupt motions and (d)(bottom) showing objects which are originally different. We propose to online learn MRMs for multi-object tracking.

Fig. 1. This visual tracking scenario tracks multiple objects simultaneously and understands their mutual relations, termed *group tracking* for simplicity.

Proposal: Our main idea is to *model objects in mutual context* with each other and combine both IOMs and MRMs for robust and accurate tracking. The usage of MRMs needs to answer two crucial questions in the following.

When do MRMs work? The most ideal situation is that all IOMs are accurate enough, and MRMs are not necessary. However, when IOMs are inaccurate, MRMs will play an important role in robust tracking. In fact, it is not easy to know whether IOMs are accurate enough to ignore MRMs. Moreover, if all IOMs lost accuracies simultaneously, MRMs will become useless because any estimates of objects are unreliable at these moments. Therefore, we assume that MRMs exist all the time and there are some accurate IOMs during tracking.

How do MRMs work? MRMs provide mainly three kinds of helpful information for tracking. The first is to indicate mutually related objects (i.e. objects impact on each other), termed the *relational graph*. The second is to know how much impact one object has on another one, named as the *mutual relation vectors*. The last is to balance all interactions of objects and the responses of IOMs, called as the *relational weights*. When these three kinds of information are properly determined, MRMs will play an important role in robust tracking.

Contribution: Our main contribution can be summarized into two folds. (1) We model objects in mutual context, and propose a unified framework to integrate individual object models and mutual relation models for multiple previously unseen objects tracking (Sec.3). (2) We extend latent SVM into an online version for learning the relational weights (Sec.4.1). To the best of our knowledge, our online algorithm is the first one to take advantage of LSVM for the tracking problems.

Organization: Related work is presented in the next section. Then, problem formulation, the mutual relation modeling and the inference are described consecutively in Sec.3, Sec.4 and Sec.5. Experimental results and discussions are provided in Sec.6. Finally, the conclusion and future work are given in Sec.7.

2 Related Work

Object Tracking. Adam et al. [6] represented the target based on histograms of local patches and combined votes of matched local patches using templates to handle partial occlusions. Since the templates are not updated, it may fail in handling appearance changes. In order to cope with appearance variations, Babenko et al. [1] proposed an online multiple instance learning algorithm to learn a discriminative detector for the target. To avoid drifting over time, Kalal et al. [7] combined an adaptive Kanade-Lucas-Tomasi feature tracker and several restrictive learning constraints to establish an incremental classifier. To further deal with non-rigid motion or large pose variations, Kwon et al. [3] extended the conventional particle filter framework with multiple motion and observation models to account for appearance variations caused by changes of pose, illumination and scale as well as partial occlusions. Considering that bounding box based representations provide a less accurate foreground/background separation, Godec et al. [2] extended Hough Forests to online domain and coupled the voting based detection and back-projection with a rough segmentation based on GrabCut for accurately tracking non-rigid and articulated objects.

Besides these single object tracking approaches, there are also lots of approaches for multi-object tracking, where most of them assume that the object category is known and depend on offline trained specific object detectors, such as [8–10]. Breitenstein et al. [8] used detection as their observation model and integrated pedestrian detectors into a particle filtering framework to track multiple persons. Huang et al. [9] proposed to associate detected results globally, assuming that the whole sequence is achieved in advance. Different from [8, 9] considering object objects individually, Pellegrini et al. [10] modeled human interactions to some degree and introduced a dynamic social behaviors to facilitate multi-people tracking. Similar to [10], group tracking considers mutual relations among objects.

Human Pose Estimation. Felzenszwalb et al. [11] utilized Pictorial Structure (PS) to estimate human poses in static images. Sapp et al. [12] proposed Stretchable Models to estimate articulated poses in videos with rich features like color, shape, contour and motion cues. Since the object is human, relations of human parts can be predefined and their distributions can be learned on annotated human pose datasets in [11, 12]. But these priors do not exist in group tracking, because objects are previously unseen.

Structural Learning and Latent SVM. We propose an online latent structural learning algorithm to update relational weights. Tsochantaridis et al. [13] optimized a structural SVM objective function, which is a convex optimization

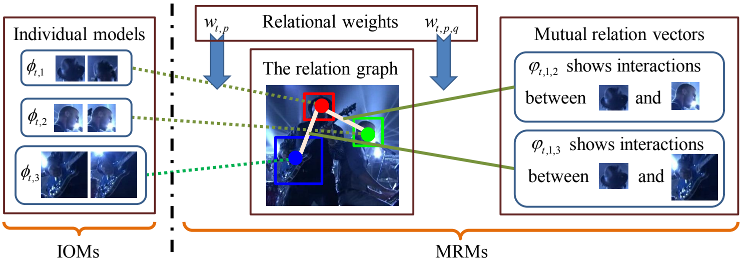


Fig. 2. An illustration of our problem formulation. We combine IOMs and MRMs for group tracking, where MRMs consist of relational weights, the relational graph and mutual relation vectors.

problem widely applied to a variety of problems in computer vision [14, 15]. Branson et al. [14] utilized online structural learning algorithm to train deformable part based model interactively. By introducing latent variables, Felzenszwalb et al. [16] proposed Latent SVM to handle object detection challenges. Ramanan et al. [15, 17] made some improvements on Latent SVM and demonstrated state-of-the-art results in object classification and human pose estimation.

3 Problem Formulation

We denote the sequences of observations and object states from frame 1 to frame T as $O_{1:T} = \{O_1, \ldots, O_T\}$ and $S_{1:T} = \{S_1, \ldots, S_T\}$. Observations consist of image features which can include image patches, corner points or the outputs of specific generative/discriminative models. We encode the states of M objects at time t as $S_t = \{s_{t,m}\}_{m=1}^{M}$. Each object state is represented by its center $\mathrm{x} = [x \quad y]^T$, width w and height h. Thus, we formalize the state of mth object at time t as $s_{t,m} = (\mathrm{x}_{t,m}, w_{t,m}, h_{t,m})$. We can now write the full score associated with a configuration of object states as

$$f(O_t, S_t) = \sum_{p \in V_t} w_{t,p} \cdot \phi_{t,p}(O_t, s_{t,p}) + \sum_{(p,q) \in E_t} w_{t,p,q} \cdot \varphi_{t,p,q}(s_{t,p}, s_{t,q}) \qquad (1)$$

where $\phi_{t,p}$ is the response of the individual model for object p, and $\varphi_{t,p,q}$ is the mutual relation vector between two objects p and q. $w_{t,p}$ and $w_{t,p,q}$ are weight parameters. $G_t = (V_t, E_t)$ represents the relational graph whose nodes are objects and edges indicate mutually related objects. For concise descriptions, we write $\beta_t = \big[\{w_{t,p}\}; \{w_{t,p,q}\}\big]$ and $\psi(O_t, S_t) = \big[\{\phi_{t,p}\}; \{\varphi_{t,p,q}\}\big]$, and thus the scoring function in Eq.(1) can be rewritten as $f(O_t, S_t) = \beta_t \cdot \psi(O_t, S_t)$. This formulation is illustrated in Fig. 2.

Individual object models (IOMs) compute the local score of placing one object at a specific frame location, corresponding to $\phi_{t,p}$. The larger score indicates the higher probability of the object at this location. As reviewed in Sec.2, there are many algorithms for single object tracking and most of them can be easily integrated into our framework. Taking MIL [1] as an example, MIL represents each object by a bag of samples and online learns a discriminative classifier. The learned MIL classifier can return the confidence of an image patch to an object.

Mutual relation models (MRMs) consist of the relational weights, the mutual relation vectors and the relational graph, corresponding to β_t, $\varphi_{t,p,q}$ and G_t respectively. Firstly, relational weights balance the two terms in Eq.(1). If relational weights $\{w_{t,p,q}\}$ are set to zero, Eq.(1) will equal to the case that ignores mutual relations. Secondly, mutual relation vectors are calculated between related objects to show the interactions. Thirdly, the inferences of Eq.(1) for cyclic and acyclic graphs are very different. In the next section, we will present details of modeling mutual relations.

4 Modeling Mutual Relations

In this section, we describe particularly how to learn relational weights, calculate mutual relation vectors and update relational graphs.

4.1 Learning Relational Weights through Online LSVM

Given a set of frames and object configurations $\{(O_1, S_1), \ldots, (O_T, S_T)\}$, the task is to learn proper relational weights (β_t) to balance the two terms in Eq.(1). We cast this task as a maximum margin structured learning problem (Structured SVM [13]), where we consider mutual relation vectors as latent variables. In addition, there is no need to depict the objects in a long time ago and they might differ a lot with the current one in appearances, and therefore we only consider samples in a short time period τ. Similar to [14], we formulate this problem to search the optimal weight β^* that minimize the error function $F_T(\beta)$

$$F_T(\beta) = \frac{\lambda}{2}\|\beta\|^2 + \frac{1}{\tau} \cdot \sum_{t=T-\tau+1}^{T} l_t(\beta) \tag{2}$$

$$l_t(\beta) = \max_{\hat{S}_t} \beta \cdot \psi(O_t, \hat{S}_t) - \beta \cdot \psi(O_t, S_t) + \Delta(S_t, \hat{S}_t) \tag{3}$$

where $\hat{S}_t$ is a particular object configuration and $\Delta(S_t, \hat{S}_t)$ is the loss function, equaling to the number of missing objects when the ground truth is S_t. This criterion attempts to learn a set of weight parameters β, so that the score of any other choice of object configurations $\beta \cdot \psi(O_t, \hat{S}_t)$ is less than that of the ground truth configuration $\beta \cdot \psi(O_t, S_t)$ by at least $\Delta(S_t, \hat{S}_t)$.

Comparison with [14]. Before designing efficient algorithms to learn relational weights, we should consider three main issues. (1) **Labeled samples**. In our tracking problem, we have only one labeled result S_1 at $t = 1$, while $\{S_t\}_{t=2}^{T}$ are tracked results, which are assumed as ground truths and will not change after T. In contrast, [14] have many well-labeled ground truths for training but there are no consistencies between images. To some extent, video consistencies are the most important information to make our problem tractable. (2) **Working samples**. We exploit recent τ frames to depict object relations, while [14] utilizes all possible training samples to learn a powerful model. (3) **Learning speeds**.

The frames come one by one during tracking, which is an incremental way similar to [14]. Thus, it requires a fast online learning algorithm.

Optimization. The form of our learning problem is a Structured SVM. There exist many well-tuned solvers such as the cutting plane solver of SVM^{struct} [18] and the Online Stochastic Gradient Descent (OSGD) solver [19]. While SVM^{struct} solver is most commonly used, OSGD solver has been observed faster in practice [17]. Since the speed is important for tracking, we choose the OSGD solver for our learning problem. When an example (O_t, S_t) comes, [19] proposed to take an update as

$$\beta_t = \beta_{t-1} - \frac{1}{\lambda_t}(\beta_{t-1} + \nabla l_t) \tag{4}$$

where $\nabla l_t(\beta)$ is the sub gradient of the hinge loss $l_t(\beta)$, which can be computed by solving a problem similar to an inference problem

$$\nabla l_t(\beta) = \psi(O_t, \bar{S}_t) - \psi(O_t, S_t) \tag{5}$$

$$\bar{S}_t = \arg\max_{\hat{S}_t} \quad \beta \cdot \psi(O_t, \hat{S}_t) + \Delta(S_t, \hat{S}_t) \tag{6}$$

The update in Eq.(4) needs at least one whole iteration for each new sample and is designed to keep the learning ability on all samples. Because of characteristics of our task (objects in a short time period and video consistencies), we modify Eq.(4) as

$$\beta_t = \beta_{t-1} - \min\left(\frac{1}{\lambda_{t-1}}, \frac{l_{t-1}(\beta_{t-1})}{\|\nabla l_{t-1}\|^2}\right)\nabla l_{t-1}. \tag{7}$$

Our update directly combines the weights and the sub-gradient of the hinge loss in the previous time. It is non-iterative because β_{t-1}, ∇l_{t-1} and $l_{t-1}(\beta_{t-1})$ are all calculated before time t[1]. Note that, the coefficient of ∇l_{t-1} in Eq.(7) might be tuned, which is different from $\frac{-1}{\lambda_t}$ in Eq.(4). Relatively, our update is more reasonable for tracking. In addition, there is no need to determine τ explicitly in this online update.

Computation. The computationally taxing portion of Eq.(7) is to calculate the sub-gradient of $l_t(\beta)$ using Eq.(6). It is too computationally expensive to sample locations around the ground truths of objects like [14, 17]. Thus, we select $\hat{S}_t$ from Θ_t, where Θ_t contains all configurations inferred by Eq.(1) except the tracked result, whose details will be described in Sec.5. Then, one needs to loop over Θ_t to get the most violated configuration $\bar{S}_t$. For each testing on $\hat{S}_t$, one needs to compute $\psi(O_t, \hat{S}_t)$ containing scores of IOMs and mutual relation vectors, and $\Delta(S_t, \hat{S}_t)$ comparing a predicted configuration with the ground truth, both of which make the computation as $O(M)$. Therefore, the sub gradient can be calculated in $O(M|\Theta_t|)$. Moreover, if some objects are missing (i.e. very small scores of IOMs), there is no need to update weights related to these objects.

4.2 Calculating Mutual Relation Vectors

Mutual relation vectors show the interactions between related objects. They can be interpreted as a spring model that represents the relative placement of

[1] This update can be extended easily to an iterative way. However, we have not found significantly improvements in our experiments.

two objects. Ideally, more accurate locations of related objects indicate higher scores. Unfortunately, only the first frame is labeled as stated before. Motivated by deformable part based model in [16, 17], we calculate mutual relation vectors with the changes of relative locations between two objects.

Following denotations in Sec.3, the relative location of p with respect to q at time t is $z_{t,p,q} = \mathrm{x}_{t,p} - \mathrm{x}_{t,q}$ and the estimated relative location is $\tilde{z}_{t,p,q} = \tilde{\mathrm{x}}_{t,p} - \tilde{\mathrm{x}}_{t,q}$, where $\tilde{\mathrm{x}}_{t,p}$ and $\tilde{\mathrm{x}}_{t,q}$ are estimated locations for object p and q respectively. The Kalman Filter [20] is adopted as the algorithm to estimate object locations, where both linear Gaussian observation and motion models are assumed for each object. Then, we define the mutual relation vectors as $\varphi_{t,p,q}(s_{t,p}, s_{t,q}) = [dx \quad dy \quad dx^2 \quad dy^2]^T$ where $[dx \quad dy]^T = z_{t,p,q} - \tilde{z}_{t,p,q}$. Although $\tilde{z}_{t,p,q}$ is always biased from the ground truths, it does not have a direct impact on the whole tracking system because mutual relation vectors are taken as latent variables and relational weights are updated online.

4.3 Updating Relational Graphs

The relational graph determines the complexity of inferring Eq.(1). A general representation is a full connected graph. However, there is no guarantee to get global optimizations on it. As our goal is to make use of frames in a short time period, we suggest that a simplified graph, i.e. tree, is reasonable enough, where dynamic programming can be used for efficient inferences which will be explained in Sec.5. However, after a long time period or when some objects move violently, one graph may lose the effectiveness and even interfere with tracking

Table 1. Our tracking algorithm

```
Input: video sequences $O_{1:T}$.
Initialize: multiple objects $S_1 = \{s_{1,1}, \ldots, s_{1,M}\}$, the relational graph $G_1 = (V_1, E_1)$ by
$E_1 = \emptyset$, the relational weight $\beta_1 = [\mathbf{1}; \mathbf{0}]$ by default ($w_{1,p}$=1,$w_{1,p,q}$=0), the frame number
of updating relational graphs $N_U$, the object speed threshold $\theta_v$, and the frame number of
updating relational weights $k$.
Output: tracked objects in all frames $S_{1:T}$.
Learn individual object models at the first frame.
For $t = 1 : T$
    – Apply individual object models at sampled locations.
    – Find the best configuration by inferring Eq.(1).
    – Store the other inferred results into $\Theta_t$.
    – Update individual object models in predicted locations $S_t$.
    – Update relational weight $\beta_{t+1}$ by Eq.(7) on $\Theta_t$.
    – If $t == k$ (i.e. constructing the relational graph), or $t > k \&\& (t \% N_U == 0 || \Omega_t \neq \emptyset)$
      where $\Omega_t = \{v_{t,p} | v_{t,p} > \theta_v\}$, (i.e. updating the relational graph).
        + Evaluate connected weights between objects by Eq.(8).
        + Generate the Minimum Spanning Tree on a full connected graph.
        + If the graph structure is changed, set $\beta_{t-k+1} = [\mathbf{1}; \mathbf{0}]$, update $\beta_{i+1}$ on $\Theta_i$
          for $i = t - k + 1$ to $t$, and obtain $\beta_{t+1}$. End If.
    – End If
End For
```

performances. Although this interference may be reduced to some extent by relational weights, it is supposed to be more reasonable to update the relational graph over video sequences.

Adaptive Relational Graphs. Intuitively speaking, a reasonable adaption encourages linking objects moving similarly and disconnecting those moving differently. Therefore, we learn reasonable relational graphs through analyzing object trajectories and define the connected weights between two objects based on their motion similarities in the latest k frames

$$\pi_{t,p,q} = 1/(1 + \sum_{i=t-k+2}^{t} \|\mathrm{v}_{i,p} - \mathrm{v}_{i,q}\|_2) \tag{8}$$

where $\mathrm{v}_{i,j} = \mathrm{x}_{i,j} - \mathrm{x}_{i-1,j} (j = p, q)$ and $\| \cdot \|$ is L_2 norm. After calculating all connected weights, one can perform Minimum Spanning Tree algorithms to generate a new tree structure. Once the graph structure is changed, the relational weights should be re-learned correspondingly. It is not reasonable to set the relational weights by default, and thus we utilize stored results of latest k frames $\{\Theta_i\}_{i=t-k+1}^{t}$ to learn β_{t+1}, in which we set β_{t-k+1} by default.

5 Inference

The inference of our formulation is to maximize Eq.(1) over all involved objects,

$$S_t^* = \arg\max_{S_t} f(O_t, S_t). \tag{9}$$

Since we restrict the relational graph to be a tree, the inference can be efficiently solved by dynamic programming. Letting $child(p)$ be the set of children of object p in G_t, we compute the message of p passing to its parent q as follows.

$$score(p) = w_{t,p} \cdot \phi_{t,p}(O_t, s_{t,p}) + \sum_{b \in child(p)} m_b(p) \tag{10}$$

$$m_b(p) = \max\left(score(b) + w_{t,b,p} \cdot \varphi_{t,b,p}(s_{t,b}, s_{t,p})\right) \tag{11}$$

Eq.(10) computes the local score of p at all possible locations, by collecting all messages from its children. Eq.(11) computes the messages at all possible locations of b and finds the best scoring location. Once all messages are passed to the root object, one can generate object configurations by starting from a root location and backtrack to find the location of each object in each maximal point with the aid of keeping track of the argmax indices. The configuration with the maximum root score corresponds to the tracked result in a frame.

Computation. After applying IOMs in one frame, we have scores of objects at a number of locations. For considering the efficiency of online tracking, we only keep L highest scoring locations for each object. Then, we have to traverse over L possible parent locations and compute over L possible child locations in Eq.(11), resulting the computation $O(L^2)$ for each object. Therefore, the total complexity of M objects is $O(ML^2)$. As each root location corresponds to an inferred configuration, there are totally L inferred configurations. Except the tracked result, other inferred configurations are stored in Θ_t for learning relational weights as in Sec.4.1. Thus, we have $|\Theta_t| = L - 1$ and the complexity of learning relational weights is $O(M|\Theta_t|) = O(ML)$.

Till now, we have elaborated our framework for group tracking. For easy references, we summarize the whole tracking algorithm in Tab.1.

6 Experiment

In this section, we carry out experiments to demonstrate the advantages of MRMs for multi-object tracking, where tracking objects are not constrained to be within a specified class. Besides MRMs, IOMs are also very important for the whole tracking algorithm. As stated in Sec.3, many approaches for single object tracking can be used as IOMs. But in this paper, we focus on showing the performance of MRMs combined with one kind of IOMs. In the experiment, we select MIL [1] as IOMs, which has achieved promising results in [1]. We use the default parameters of MIL as in [1] and will demonstrate in the following that the combination of MIL and MRMs improves tracking performances further. Combining other kinds of IOMs with MRMs is one direction of the future work. All experiments below are conducted on an Intel Core(TM)2 2.33GHz PC with 2G memory.

6.1 Experimental Setup

Parameters. We select at most $L = 100$ locations for each object in each frame and always set the weight update parameter $\lambda_t = 1000$ in Eq.(7). We update the relational graph every $N_U = 50$ frames or if the speed of one object is larger than $\theta_v = 8$. If the change of the relational graph is detected in Tab. 1, we use the tracked results of latest $k = 3$ frames to learn new relational weights.

Datasets. To build up and utilize mutual relations efficiently, a proper test sequence requires that objects are fully or partially visible in most frames. Currently, there are no standard datasets for this objective. Therefore, we select ten sequences from [1, 3, 21, 22] for evaluations, where we can easily label multiple objects for tracking. We select *david* indoor, occluded *face*, occluded *face2*, *coke* and *cliffbar* from [1] and *shaking*, *animal* and *skating1*[2] from [3], which are widely used for single object tracking and cover the difficulties of occlusions, changing appearances, varying illumination and abrupt motions. [21] is a widely used dataset for pedestrian tracking. We select a relatively challenging sequence, *Seq.*#3, because of frequently and significantly light changes. [22] is a *sign* language dataset which focuses on locating arms and hands. Abrupt motions of hands, similar appearances of hands and relatively low resolutions make this dataset very difficult for online visual tracking. The whole sequence is very large, and we select a representative clip (#191∼#490) for evaluation.

[2] There are no several objects existing simultaneously in most frames of *skating1* due to abrupt motions. To track several objects, we select #210∼#345 for evaluations.

Metrics. Center *location errors* along with frame numbers are widely used for evaluating single object tracking algorithms and the mean error over all frames is a summarized performance. However, if a tracker tracks a nearby object but not the concerned one for a long time, location errors fail to correctly capture tracker performances. Another metric is the *overlap criteria*, where a result is positive only if the intersection is larger than 50% of the union comparing to the object with the same label. Using this metric, we can calculate the detection rate by $TruePos/(FalseNeg + TruePos)$.

Compared Algorithms. To our knowledge, there are no implementations available publically for group tracking, and thus we compare our approach with the algorithms applying IOMs individually. Some applied state-of-the-art algorithms for single object tracking are Online-AdaBoost (OAB) [23], FragTrack [6], MIL [1], TLD [7] and VTD [3]. We utilize the codes provided by the authors and carefully adjust the parameters of the trackers for fair comparisons. We use the best five results from multiple trials and average the location errors and detection rates, or directly take results from the prior works.

6.2 Performance Evaluation

6.2.1 Evaluation on Two Objects

First, we evaluate our approach on the simplest situation, tracking two objects together as illustrated in red in Fig.3, where the relational graph always connects the two objects. We show the location error plots of some objects in Fig.4, and summarize the average location errors and detection rates in Fig.5. Generally, our approach has improved MIL significantly in most sequences with lower location errors and higher detection rates. Due to the challenges of the testing sequences, MIL does not perform as well as VTD and TLD in many sequences, while our approach performs as well as or even better than TLD and VTD in some sequences. Note that Fig.5(a) demonstrates the reduced average location errors of MRMs+IOMs compared to only IOMs where MIL is selected as IOMs. Although our current implementation is not the best in average location errors, we believe that our framework can also improve their performances to some extent if we select other algorithms as IOMs.

We mention some typical results on *face*, *coke*, *shaking* and *skating1*. Because of frequent occlusions in *face*, MIL performs much less precisely than Frag, TLD and VTD. Our approach utilizes effective MRMs to achieve as precise results as Frag, TLD and VTD. The evaluated object in *coke* is specular, which causes some difficulties to learn generative models and discriminative classifiers for it. Thus VTD and MIL drift easily, while TLD obtains more stable results with structured constraints on samples. With the help of effective MRMs, our approach achieves the highest detection rate in *coke*. Because of abrupt motions and severe illumination changes in *shaking* and *skating1*, TLD and MIL become inaccurate and lost objects in early frames. Our approach obtains a lower detection rate than VTD in *shaking*. The main reason might be that our approach can learn relatively robust mutual relations, but the applied individual model,

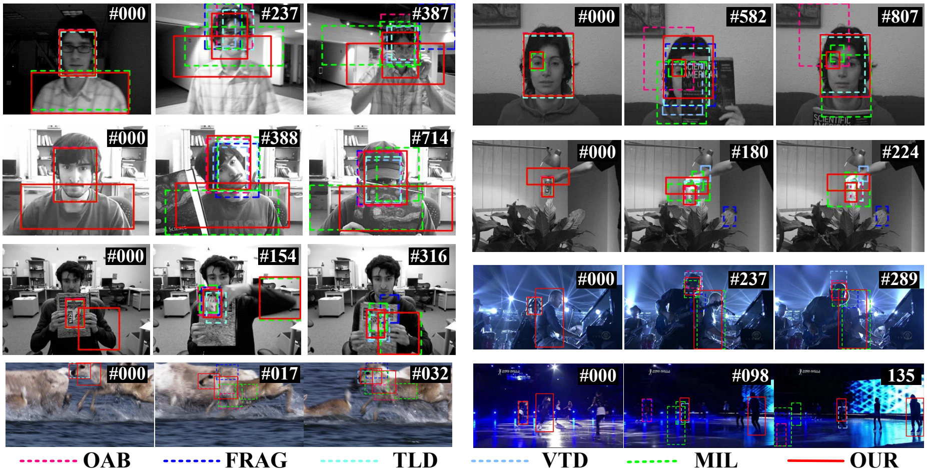


Fig. 3. Qualitative results on *david*, *face*, *face2*, *coke*, *cliffbar*, *shaking*, *animal* and *skating1* sequences from left to right and top to down

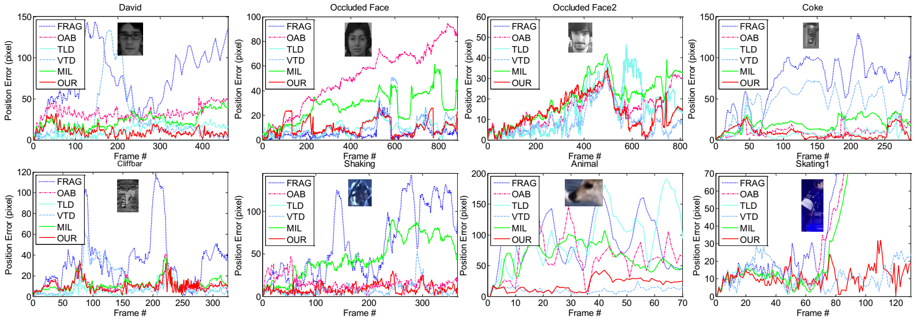
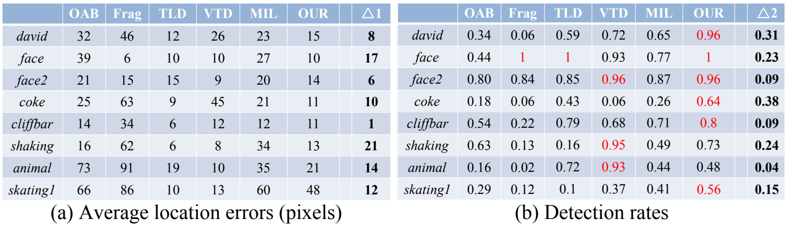


Fig. 4. Location error plots of some objects on test sequences

MIL is not designed to handle with both abrupt motions and severe illumination changes. Moreover, our detection rate in *skating1* is higher than that of VTD. The reason is that the tracking objects in *skating1* move violently from the very beginning and undergo large variations of poses and scales, which causes many difficulties for building holistic appearance models.

6.2.2 Evaluation on More Objects

Then, we carry out experiments in more complex situations, tracking more than two objects. Different from fixed relational graphs previously, we show the performances of adaptive relational graphs in this subsection. We select two challenging sequences, *Seq.#3* [21] and *Sign* [22], and numerate fixed relational graphs to compare with adaptive relational graphs as shown in Fig.6(a,c).



	OAB	Frag	TLD	VTD	MIL	OUR	△1
david	32	46	12	26	23	15	**8**
face	39	6	10	10	27	10	**17**
face2	21	15	15	9	20	14	**6**
coke	25	63	9	45	21	11	**10**
cliffbar	14	34	6	12	12	11	**1**
shaking	16	62	6	8	34	13	**21**
animal	73	91	19	10	35	21	**14**
skating1	66	86	10	13	60	48	**12**

(a) Average location errors (pixels)

	OAB	Frag	TLD	VTD	MIL	OUR	△2
david	0.34	0.06	0.59	0.72	0.65	0.96	**0.31**
face	0.44	1	1	0.93	0.77	1	**0.23**
face2	0.80	0.84	0.85	0.96	0.87	0.96	**0.09**
coke	0.18	0.06	0.43	0.06	0.26	0.64	**0.38**
cliffbar	0.54	0.22	0.79	0.68	0.71	0.8	**0.09**
shaking	0.63	0.13	0.16	0.95	0.49	0.73	**0.24**
animal	0.16	0.02	0.72	0.93	0.44	0.48	**0.04**
skating1	0.29	0.12	0.1	0.37	0.41	0.56	**0.15**

(b) Detection rates

Fig. 5. Quantitative results. After comparing our approach with MIL, Δ_1 in (a) shows reduced mean location errors, and Δ_2 in (b) shows improved detection rates.

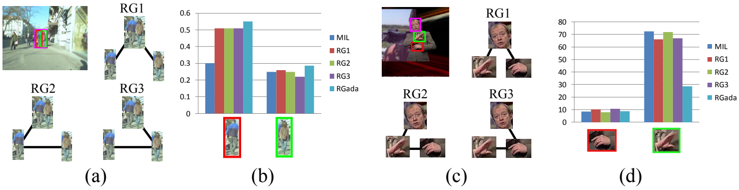


Fig. 6. Tracking results on *Seq.*#3 (a,b) and *Sign*(c,d). (a,c) show tracking objects in rectangles and numerated trees of relational graphs. (b) compares the detection rates on *Seq.*#3 and (d) compares average location errors on *Sign*, where RGada indicates the approach of using adaptive relational graphs.

Tracking objects in *Seq.*#3 are man, woman and their entirety which are walking together with frequently light changes. As *Seq.*#3 is fully annotated with rectangles, we use detection rates to compare our approaches with MIL in Fig.6(b). Due to frequently light changes, MIL drifts easily on the man and woman. Our approaches, with fixed relational graphs and the adaptive relational graph, all show significantly improvements than MIL on the man, and particularly the adaptive relational graph improves the detection rate by about 25%. However, the improvement of the adaptive structure on the woman is slightly (about 3.7%) and RG3 drops the tracking performance slightly, mainly because light changes on the woman are more severe than those on the man.

Tracking objects in *Sign* are head, left hand and right hand, whose motions are abrupt. Because *Sign* is sparsely annotated with masks while tracked results are in fixed ratios, we use average location errors to compare our approaches with MIL in Fig.6(d). MIL tracks the left hand well, and our approaches with mutual relations also keep the accuracies on it. However, due to violently motions and changed appearances of the right hand, MIL and fixed relational graphs drift easily. In contrast, the adaptive relational graph performs more accurate and always tracks the real objects as shown in Fig.7.

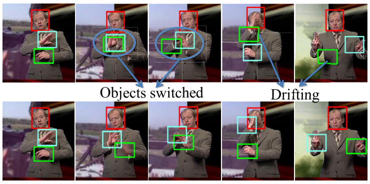


Fig. 7. Tracking results of MIL(top) and OUR(bottom) on #1, #11, #24, #32 and #163 from *Sign*.

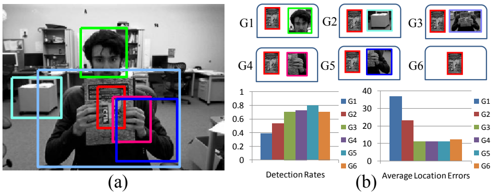


Fig. 8. Examples for the impacts of selected objects on tracking. Please see Sec.6.2.3 for more details.

6.2.3 Impacts of Selected Objects

Our framework consists of IOMs and MRMs. Accurate IOMs are the basis of learning robust MRMs, and in return the improved accuracies of object locations by robust MRMs will make IOMs more accurate. If selected objects encounter great challenges, IOMs may be inaccurate in some frames and the entire performances are also affected. For example, the detection rates of *coke*, *shaking*, *animal* and *skating1* are less than those of other sequences in Sec.6.2.1, mainly because of severe illumination changes and abrupt motions. If the connections of selected objects are relatively strong, robust MRMs can be efficiently learned, such as the sequences in Sec.6.2.1. Due to frequently light changes, the connection of pedestrians in *Seq.*#3 is weak, leading to the reduced performance of RG3 in Fig.6(b). Similarly, the connection of hands in *Sign* is weak due to abrupt motions and fixed relational graphs drift easily.

As our main contribution is in MRMs, we give more analysis on MRMs, especially in two-object tracking because it is the most basic situation. We choose a typical sequence *cliffbar*, where the evaluated object is bar as shown in red in Fig.8(a). We select five candidates, head, box, torso, hand and elbow, and then track both bar and one candidate each time, resulting in the five groups G1∼G5 in Fig.8(b)(top). As MIL performs similarly on these objects, the performances are mainly influenced by MRMs. The connections between objects in G3/G4/G5 are relatively stronger than those in G1/G2. It is because bar directly links with hand, elbow and torso, and their motions are related with each other, but these connections are much weaker between bar and head or box. This explains the results in Fig.8(b)(bottom), where G3/G4/G5 achieve slightly better results than MIL, but G1/G2 reduce the performance a lot.

Overall, our approach can exploit MRMs well for robust tracking when selected objects have relatively strong connections and some of the IOMs are relatively accurate. To enhance connections between objects, we need more proper models to measure interactions of objects and construct relational graphs. To cope with inaccurate IOMs, we may utilize proper IOMs for different objects. They are both beyond the scope of this paper and will be studied in the future.

6.3 Discussion

Speed. The inference is highly efficient in our experiment (less than 10ms), while the bottleneck of our implementation is online learning IOMs. We observe that the overall speed except the inference is slightly slower than tracking objects individually. A partial reason might be that inferred locations are not discriminative and it causes some difficulties for online learning object models.

Comparison with Other Works. Although [8, 9] and our approach are all proposed for multi-object tracking, their concentrations are different. Since tracked objects are known in [8, 9], [8, 9] are provided with relatively good offline learned object models (detectors) and are able to cope with many objects simultaneously. Thus, [8, 9] focus on handling with ID switches because the offline models could not distinguish objects in the same category, and coping with fragments because the offline models could not cover all kinds of objects in this category. However, as objects are previously unseen in our problem, we should online build models for objects, where great challenges come from changes of objects and the surroundings. Therefore, our main concern is to build up proper IOMs and MRMs to avoid drifting and switching errors particularly when tracked objects are in similar appearances.

7 Conclusion and Future Work

In this paper, we present a novel framework of combining IOMs and MRMs to track multiple previously unseen objects. In MRMs, the relational graph indicates related objects, the mutual relation vectors show the interactions, and the relational weights balance all interactions and IOMs. We use online LSVM to learn relational weights and analyze object trajectories to update relational graphs. Various experiments show the advantages of our framework.

There are several interesting ways to extend this work in the future. Firstly, there is no single algorithm that can cover all difficulties in visual tracking. Fortunately, our framework provides an easy way to integrate different IOMs and we will try to find if some models can complement each other. Secondly, we will collect proper real world datasets for direct evaluations on tracking multiple (≥ 3) previously unseen objects. Finally, our framework can be extended to track some specific object if we online select some regions or points like [4].

Acknowledgements. This work was supported in part by the National Science Foundation of China under Grant No.61075026, the National Basic Research Program of China under Grant 2011CB302203.

References

1. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR (2009)
2. Godec, M., Roth, P.M., Bischof, H.: Hough-based tracking of non-rigid objects. In: ICCV (2011)

3. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR (2010)
4. Grabner, H., Matas, J., Gool, L.V., Cattin, P.: Tracking the invisible: Learning where the object might be. In: CVPR (2010)
5. Dinh, T.B., Vo, N., Medioni, G.: Context tracker: Exploring supporters and distracters in unconstrained environments. In: CVPR (2011)
6. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR (2006)
7. Kalal, Z., Matas, J., Mikolajczyk, K.: P-n learning: Bootstrapping binary classifiers by structural constraints. In: CVPR (2010)
8. Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Online multi-person tracking-by-detection from a single, uncalibrated camera. PAMI 33, 1820–1833 (2011)
9. Huang, C., Wu, B., Nevatia, R.: Robust Object Tracking by Hierarchical Association of Detection Responses. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 788–801. Springer, Heidelberg (2008)
10. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: ICCV (2009)
11. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. IJCV 61, 55–79 (2005)
12. Sapp, B., Weiss, D., Taskar, B.: Parsing human motion with stretchable models. In: CVPR (2011)
13. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. JMLR 6, 1453–1484 (2005)
14. Branson, S., Perona, P., Belongie, S.: Strong supervision from weak annotation: Interactive training of deformable part models. In: ICCV (2011)
15. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV (2009)
16. Felzenszwalb, P., McAllester, D., Ramaman, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
17. Yang, Y., Ramanan, D.: Articulated pose estimation using flexible mixtures of parts. In: CVPR (2011)
18. Finley, T., Joachims, T.: Training structural svms when exact inference is intractable. In: ICML (2008)
19. Hazan, E., Agarwal, A., Kale, S.: Logarithmic regret algorithms for online convex optimization. Machine Learning 69, 169–192 (2007)
20. Kalman, R.E.: A new approach to linear filtering and prediction problems. Transactions of the ASME-Journal of Basic Engineering 82, 35–45 (1960)
21. Ess, A., Leibe, B., Gool, L.V.: Depth and appearance for mobile scene analysis. In: ICCV (2007)
22. Buehler, P., Everingham, M., Huttenlocher, D., Zisserman, A.: Long term arm and hand tracking for continuous sign language tv broadcasts. In: BMVC (2008)
23. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via online boosting. In: BMVC (2006)