Parameter Estimation and Contextual Adaptation for a Multi-Object Tracking CRF Model

Alexandre Heili Jean-Marc Odobez
Idiap Research Institute – CH-1920, Martigny, Switzerland
École Polytechnique Fédérale de Lausanne – CH-1015, Lausanne, Switzerland
{aheili,odobez}@idiap.ch

Abstract

We present a detection-based approach to multi-object tracking formulated as a statistical labeling task and solved using a Conditional Random Field (CRF) model. The CRF model relies on factors involving detection pairs and their corresponding hidden labels. These factors model pairwise position or color similarities as well as dissimilarities, and one critical issue is to be able to learn their parameters in an accurate and unsupervised way. We argue in this paper that tracklets and local context can help to obtain relevant parameters. In this context, the contributions are as follows: i) a factor term global parameter estimation based on intermediate tracking results; ii) a detection dependent parameter adaptation scheme that allows to take into account the local detection contextual information during online tracking. Experiments on PETS 2009 and CAVIAR datasets show the validity of our approach, and similar or better performance than recent state-of-the-art algorithms.

1 Introduction

Tracking multiple people has several potential applications, notably in open-space, surveillance scenarios. Being able to follow people in a scene can help performing trajectory-based human activity analysis, or extracting further behavioral cues like body or head pose [9] to conduct higher level behavior and interaction interpretation. However, this remains a difficult task with important challenges such as appearance changes or occlusions.

Detection-based tracking methods have become popular in the literature [17, 7, 6]. In this context, the task of multi-person tracking can be formulated as a labeling problem, in which we want to assign consistent identity labels to the different people in the scene over time. The main idea is to perform data association between detections produced by a human detector in order to cluster them into distinct coherent tracks, having the same label. Such methods should overcome detector flaws like mis-detections, false alarms and imprecise bounding box localizations.

Our work is inspired by recent works on Conditional Random Field (CRF) models for tracking [6], and in particular by [16]. This approach exploits a CRF model in which factor terms account for both the similarity *and* dissimilarity of the features of detection pairs within a short-time interval T_w. The framework allows to consider any type of features rather than just positional information like in flow programming methods [7]. In addition, having relations between all detections less than T_w apart contrasts with approaches modeling tracks as sequences of detection pairs and only considering likelihood terms between consecutive detections of this ordered sequence. The denser graph involving both similarity/dissimilarity measures reinforces clusters having consistent features over different time intervals, and helps solving temporary ambiguities, like in the case of missing detections. Importantly, authors of [16] proposed an automatic method to learn the model parameters in an unsupervised manner.

However, as T_w increases, the unsupervised parameter learning of [16] becomes less accurate resulting in non-discriminant models, although the use of larger T_w values should increase the model robustness, e.g. by avoiding track fragmentation when no detections are observed over long periods. Another limitation was that the same factor terms (or equivalently parameters) were applied to every pair of detections whatever their surroundings. In practice though, if one would observe successive detections isolated in part of the scene, our prior that they should be associated should increase, even if their similarity is not as high as expected; and vice-versa, when multiple close-by detections are observed over time, associating them together should rely on some stricter similarity observation. Such fixed parameter settings are often encountered in tracking by detection clustering-based approaches, e.g. in [26]. To address these issues, the paper makes the following contributions:

- we propose to learn the model parameters from intermediate tracking results, leading to more accurate factor terms;
- we show that using larger T_w values improves results, provided that the parameters are learned as above;

- we propose a method to use the local context of each detection pair to adapt the location similarity parameters at test time.

Applied on the PETS and CAVIAR data, the algorithm produces equal or better performance than recent state-of-the-art methods, demonstrating the validity of our approach.

Paper plan: Section 2 analyzes related works. The CRF model is introduced in Section 3. Section 4 presents our unsupervised parameter learning and adaptation approach, while Section 5 addresses optimization. Experiments are presented in Sec. 6, and Sec. 7 concludes the paper.

2 Related Work

Tracking-by-detection has become a popular topic in computer vision. On the contrary to generative approaches like Bayesian methods [25], using a discriminative classifier to assess the presence of an object in a scene is generally more robust, as state-of-the-art detectors give very good performance on detecting humans [11][12].

Detection-based multi-person tracking has been modeled in numerous ways. For example, the authors of [7] formulate tracking as a flow optimization of people over a discrete grid representing the ground plane. One advantage is that their approach enforces tracks to remain within physical constraints and to start and end at specified possible sources and sinks, preventing tracks to end in the middle of the scene. In [22], the authors extend their method by adding global appearance constraints.

Labeling detections with identity labels can also be done jointly with finding smooth trajectories that best explain the data. The method proposed in [3] tackles the problem by alternating between discrete data association and continuous trajectory estimation using global costs. This method relies solely on trajectories and does not involve appearance of objects.

The complexity of the association increases rapidly depending on the size of the considered time window and the number of detections. Usually, data association is performed over batches of frames [15]. One solution to lower this complexity is to hierarchically perform the data association. In [17], low-level association is done over a small number of frames and the obtained tracklets are progressively associated at a higher level. As there are fewer tracklets than detections, the computational burden is decreased. The method presented in [5] uses a CRF model. This method is hierarchical like [17] and assumes low-level tracklets are already obtained. Their CRF models dependencies among these tracklets and uses energy costs based on motion models and online learned appearance models to perform tracklet association. In [14], motion planning is used to guide and refine the tracks of individuals. In the case of crowded scenes where occlusions between people are more likely to happen, multi-view methods like [7] have been proposed to fuse the information from several overlapping camera views.

One important issue for detection-based multi-person algorithms is the model adaptation. A pre-learned global affinity model usually does not work well under all scenarios. To achieve good performance, the learned model has to be adapted with regard to the local context (e.g. local crowd patterns, local scene structure...). To this end, some papers [20][24] learn context models. However, this cannot be done when there are not enough data to train the context model. In order to improve human detection and tracking, local crowd density estimations can be used [19]. This procedure can also be seen as an adaptation based on context. In the field of classifier-based tracking, the authors of [21] propose an adaptation scheme in which the weak classifiers are learned incrementally through online boosting, so that they conform with the changes over time. The response of the boosting classifier is combined with a mean-shift algorithm to track the object of interest. In [4], the authors propose to match tracklets in a bipartite graph matching framework. In this context, speed and appearance models within each tracklet are learned. Then, a prediction-based affinity between tracklet pairs is obtained through MCMC sampling. One contribution of their paper is tracklet adaptation: based on the variance of the observed features along a path, the affinity scores on edges where peaks occur are adapted using the Metropolis-Hastings algorithm. In our paper we also perform adaptation on the affinity scores. However, we refine the affinities between pairs of detections and instead of stochastically sampling the new scores after applying the pre-trained model as in [4], we do adaptation directly on the model.

In our proposed approach, we both leverage on tracklet information and learn local spatiotemporal context on the fly in order to perform model adaptation.

3 CRF Tracking Framework

Problem formulation. Let us define the set of detections of a video sequence as $R = \{r_i\}_{i=1:N_r}$, where N_r is the total number of detections. Each detector output $r_i = (t_i, X_i, h_i)$ is characterized by its frame of occurrence t_i, its ground-plane position X_i and its multi-resolution color histogram h_i. We formulate data association as a labeling problem, in which we seek for the optimal label field $L = \{l_i\}_{i=1:N_r}$, where l_i denotes the label of detection r_i, so that detections within a same track should be assigned the same label. The labeling uses similarity/dissimilarity measures between pairs of detections to perform the association. Labels can take their values in $\mathbb{N}$ as we do not know in advance the number of objects in the scene.

Similarly to [16], we can set this labeling task into a CRF formulation. Considering only the information given by pairs of observations, we can model directly the posterior probability of the label field given all the observations:

$$p(L|R,\lambda) = \frac{1}{Z(R)} \left(\prod_{(i,j)} \prod_{k=1}^{N_{f_2}} \Phi_k(l_i, l_j, r_i, r_j, \lambda^k) \right) \quad (1)$$

where for each detection pair we introduce factor terms Φ_k for N_{f_2} types of pairwise features and pairwise similarity/dissimilarity hypotheses. $Z(R)$ denotes a normalization term and $\lambda = \{\lambda^k\}$ the set of parameters of the factors.

Factor modeling. The factors Φ_k are modeled using a short-term, two-hypothesis and time-dependent pairwise approach. More precisely, we limit the number of detection pairs by only considering pairs (r_i, r_j) verifying $1 \leq |t_i - t_j| \leq T_w$. The factors are further defined as follows. For each valid pair, we extract as factor features f their Euclidean distance and their Bhattacharyya color distance D_h:

$$\begin{cases} f_1(r_i, r_j) = X_i - X_j \\ f_2(r_i, r_j) = D_h(h_i, h_j) \end{cases} \quad (2)$$

Then, we make a two-hypothesis model assumption: the distribution of the feature f_k only depends on whether the labels are the same (i.e. $l_i = l_j$ that is we have the hypothesis $H(l_i, l_j) = H_1$) or not (i.e. $l_i \neq l_j$ that is we have $H(l_i, l_j) = H_0$). Note that this two-hypothesis approach enables us to model not only the similarity between observation pairs, but also the dissimilarity, leading to a more discriminative model (see optimization section as well).

The probability distributions for each feature type are defined as follows. For the position feature, we assume that it follows a Gaussian distribution with 0 mean and whose covariance depends on the two label hypotheses H_0 or H_1 and also *on the time gap* $|t_i - t_j|$ between the detection pairs:

$$p(f_1(r_i, r_j) = f | H(l_i, l_j) = H, \lambda^1) = \mathcal{N}(f \,;\, 0, \Sigma^H_{|t_i - t_j|}) \quad (3)$$

For the color feature, we use a non-parametric model, in which we discretize the Bhattacharyya distance into several bins, and use for each time gap Δ and hypothesis H (H_0 or H_1) a multinomial distribution of parameters $\alpha_{\Delta,H}$ as distributions of the quantized D_h. In other terms, we have:

$$p(f_2(r_i, r_j) = f | H(l_i, l_j) = H, \lambda^2) = \alpha_{|t_i - t_j|, H}(b(f)) \quad (4)$$

where $b(f)$ denotes the bin index associated with the Bhattacharyya distance f after quantization.

Model summary and parameterization. Ultimately, the posterior probablity we wish to maximize can thus be rewritten as follows:

$$p(L|R,\lambda) \propto \prod_{\Delta=1}^{T_w} \prod_{\substack{(i,j) \\ |t_i - t_j| = \Delta}} \prod_{k=1}^{2} p\left(f_k(r_i, r_j) | H(l_i, l_j), \lambda^k_\Delta\right) \quad (5)$$

The parameters of the models are thus defined for each feature as $\lambda^k = \{\lambda^k_\Delta, \Delta = 1..T_w\}$, with $\lambda^1_\Delta = \{\Sigma^{H_0}_\Delta, \Sigma^{H_1}_\Delta\}$ for the position feature, and with $\lambda^2_\Delta = \{\alpha_{\Delta,H_0}, \alpha_{\Delta,H_1}\}$ for the color feature. It is worth emphasizing that each factor is time-sensitive, as the parameters depend on the time between the detection pairs.

4 Unsupervised Parameter Learning

The appropriate setting of the model parameters is of crucial importance for achieving good tracking results. Since feature distributions exhibit time dependencies, one wants to use parameters for each time gap rather than using a single parameter set regardless of the time gap. However, this increases the parameter space size, so that manual setting of these parameters is not a good option. Similarly, one would like to avoid supervised learning, as this would require tedious track labeling for each scene or camera.

In the following, we present the different methods used to automatically set the parameters. They are all fully unsupervised, i.e. they only require the detections without any labels as input. First, we summarize the approach proposed in [16] that only relies on raw detections, and show its limitation. In Sec. 4.2, we present our method to obtain more accurate learning for longer association window durations T_w, and finally show how to adapt the parameters in function of the local context of each detection pair.

4.1 Unsupervised Learning from Detections

Learning the model parameters λ can be done in a fully unsupervised way using a sequence of detection outputs, either a training sequence for the scene, or the whole test sequence (batch mode) or detection outputs until the given instant (online mode).

Given that no label is provided, the intuition used in [16] to collect training data was the following: for a given detection at time t, the closest detection amongst the detections at time $t + \Delta$ should statistically correspond to a detection of the same track. Thus, for each time gap Δ, for each detection its closest and second closest detection Δ frames away are identified and the features f_1 or f_2 (cf Eq. 2) of the obtained pairs are computed and collected into appropriate sets. For the position, both the features with the closest and second closest detections were put in the same set $\mathcal{P}_\Delta$. Then, the feature distribution in this set was assumed to follow the mixture model:

$$p(f_1) = \sum_{m=1}^{2} \pi_m \mathcal{N}(f_1; 0, \Sigma_m) \qquad (6)$$

whose parameters were estimated using an Expectation-Maximization (EM) algorithm. From the resulting covariances, the smallest one (as measured by the determinant magnitude) was taken as the covariance $\Sigma_{\Delta}^{H_1}$ and the largest one as $\Sigma_{\Delta}^{H_0}$. Hence, the 2D Gaussian for hypothesis H_1 is much peakier than the one representing H_0, meaning that a pair of detections (r_i, r_j) within a close distance will be more likely under H_1 ($l_i = l_j$) than under H_0 ($l_i \neq l_j$).

Since we used a non-parametric model for the color, using a mixture model like above is impossible. Thus, in a more direct way, the features f_2 between all detections and their closest detection Δ frames apart were put in the same set $\mathcal{C}_{\Delta,H_1}$, while the feature f_2 of other detection pairs were put in $\mathcal{C}_{\Delta,H_0}$. The color parameters $\{\alpha_{\Delta,H_0}, \alpha_{\Delta,H_1}\}$ were then simply estimated by taking normalized and smoothed histograms of $\mathcal{C}_{\Delta,H_0}$ and $\mathcal{C}_{\Delta,H_1}$ respectively.

4.2 Learning from Tracklets

We just showed that model parameters could be learned from pairs of closest and second closest detections, with the intuition that, closest pairs likely come from the same individuals and pairs of second closest likely come from different individuals. This assumption holds reasonably well for small values of Δ or low crowding, but might not be verified for larger temporal gaps. Indeed, given the current location of a detection, its closest detection a long time later might just be another pedestrian passing near this location. The collected feature sets might thus not correspond to their hypothesis of coming from the same or different tracks, and become more blended resulting in non-discriminant parameter estimates. This is illustrated for the color model in the left column of Fig. 1, which displays the probability of Bhattacharyya distances between two detections under each hypothesis. When Δ is small, the separation between the 2 distributions remains clear, but after 2 seconds ($\Delta = 15$), the distributions get almost super-imposed.

To address this issue, the main idea is to use the tracklets from intermediate results to build the feature sets $\mathcal{P}_{\Delta}$, $\mathcal{C}_{\Delta,H_0}$ and $\mathcal{C}_{\Delta,H_1}$ required to estimate the model parameters. Indeed, in Sec. 6, we show that very reliable tracklets can be obtained when using a reasonably small T_w value. Thus, taking pairs of detections within and between these tracklets gives a better idea about whether they come from the same person or not.

Thus in practice, parameters were learned as follows. We apply our tracking algorithm using a small value $T_w^{\star}$ of the association window and parameters obtained using the method of Sec. 4.1. Then, we build $\mathcal{P}_{\Delta}$, $\mathcal{C}_{\Delta,H_0}$ and $\mathcal{C}_{\Delta,H_1}$ appropriately from the resulting tracklets, and relearn the parameters up to the desired T_w value. Note that the method is still unsupervised and the relearned models are still global (i.e. not specific to any track or detection).

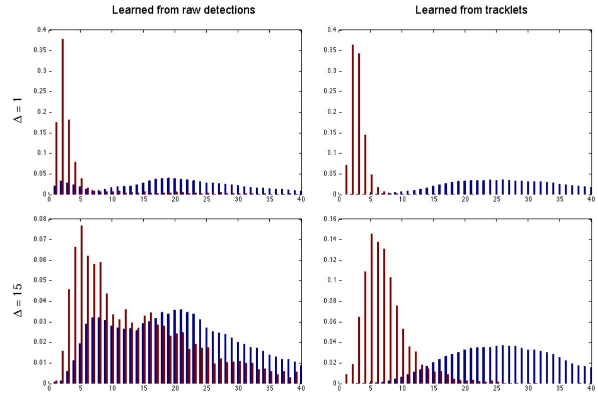

Figure 1: Estimated color models (red histograms of Bhattacharyya distances when labels are supposed to be the same (H_1), and blue when they are not (H_0)), for different values of Δ, and using different collected feature sets: raw detections (left column) and from tracklets (right).

The effect of this step is illustrated on the right column of Fig. 1. As can be seen, parameters learned from tracklets (with $T_w^{\star} = 8$) are more sensible (and still discriminative), especially for large values of T_w, as compared to those obtained using the method from Sec. 4.1.

4.3 Local Contextual Adaptation

The factors terms of Eq. (3) that give the distance feature probabilities for similarity and dissimilarity hypotheses are global: the same model parameters ($\Sigma_{\Delta}^{H_0}$ and $\Sigma_{\Delta}^{H_1}$) are used for all detection pairs. However, in practice, the local crowd pattern can vary a lot and one may want to adapt this global model according to the the spatiotemporal context of each detection pair. For example, if the local space-time region of a video is crowded, then the distances between detections in this region will be smaller, and we would like our model to shrink the spread of feature distributions for both hypotheses. On the other hand, if the neighborhood of a detection is not crowded, we might want to relax our models, so as to allow the association to be done for larger distances. This could be useful to compensate for potentially large detection localization jitter due for example to projected shadows on the floor, as will be shown in experiments.

So to conduct contextual adaptation, we propose to adapt the $\Sigma_{\Delta}^{H_0}$ and $\Sigma_{\Delta}^{H_1}$ covariances from new observations $d_i^{H_0}$ and $d_i^{H_1}$ respectively representing the local context of the detection r_i, using a MAP procedure:

$$\Sigma_{\Delta,i}^{\star} = \arg\max_{\Sigma_{\Delta}} p_0\left(\Sigma_{\Delta}\right) p\left(d_i | \Sigma_{\Delta}\right) \qquad (7)$$

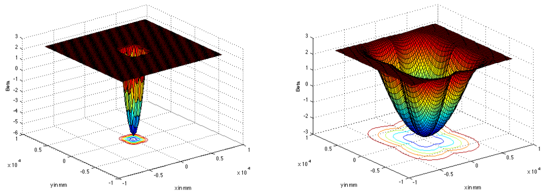

Figure 2: The β surface and iso-contours (below) for the position model for $\Delta = 3$ (left) and $\Delta = 15$ (right)

where $p_0\left(\Sigma_\Delta\right)$ denotes a prior distribution over Σ_Δ, and $p\left(d_i|\Sigma_\Delta\right)$ is the multivariate Gaussian of the factor term of Eq. (3). Note that in Eq. (7) and hereafter we drop the superscript H_1 and H_0 on Σ for simplicity, as the parameters for H_1 and H_0 are updated in the same manner. For $\Sigma_\Delta^{H_1}$, the new observation $d_i^{H_1}$ for adaptation is given by the distance to the closest detection with a Δ time gap (implicitely assumed to belong to the same track), while for $\Sigma_\Delta^{H_0}$, $d_i^{H_0}$ is the distance to the second closest detection.

For simplicity, we assume that our prior is the conjugate distribution to our Gaussian likelihood $p\left(d|\Sigma_\Delta\right)$, that is an Inverse-Wishart parameterized by κ_Δ and mode Σ_Δ^g, where Σ_Δ^g are the global parameters estimated using the method of Sec. 4.2. Under this assumption, the MAP estimation for the detection r_i is given by:

$$\Sigma_{\Delta,i}^\star = \frac{\left(\kappa_\Delta + 3\right)\Sigma_\Delta^g + d_i d_i^T}{\kappa_\Delta + 4} \tag{8}$$

After adaptation, the feature likelihood of a pair (r_i, r_j) with a time difference $\Delta = |t_j - t_i|$ replacing Eq. (3) is calculated based on both their updated covariances:

$$p(f_1|H, \lambda^1) = \sqrt{\mathcal{N}(f_1; 0, \Sigma_{\Delta,i}^{\star,H})\mathcal{N}(f_1; 0, \Sigma_{\Delta,j}^{\star,H})} \tag{9}$$

Note that this adaptation step is unsupervised, as we do not use label information, and that the parameter κ_Δ controls the relative importance of prior information against the new observation during adaptation.

5 Optimization

It can be shown that maximizing equation 5 is equivalent to minimizing the energy function [13]:

$$U(L) = \sum_{\substack{(i,j) \\ |t_i - t_j| \leq T_w}} \beta_{ij}^{\text{potts}} . \delta(l_i - l_j) \tag{10}$$

where $\delta(.)$ is the Kronecker function ($\delta(a) = 1$ if $a = 0$, $\delta(a) = 0$ otherwise), and the Potts coefficient for each pairwise link is defined as:

$$\beta_{ij}^{\text{potts}} = \log\left[\frac{\prod_{k=1}^2 p(f_k(r_i, r_j)|H_0)}{\prod_{k=1}^2 p(f_k(r_i, r_j)|H_1)}\right] \tag{11}$$

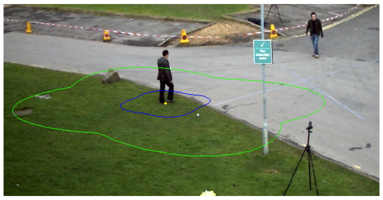

Figure 3: The iso-contour of value 0 of the β surface for the position model for $\Delta = 3$ (blue) and $\Delta = 15$ (green), centered around one detection.

The Potts coefficient simply gives the loglikelihood ratio between the two hypotheses for a pair and acts like a "cost" of associating the pair. The more negative this coefficient will be, the more likely the pair of detections should be associated, so as to minimize the energy in Equation 10.

As an illustration, the Potts coefficient in function of the distance in x and y is shown in Fig. 2 for the 2D position model, for two different time intervals. We can observe that β is highly negative for distance features close to 0 and increases with the distance. The iso-contours of the β surface are also shown. Amongst them, the zero-contour is a good indicator of the learned model, as it shows the frontier between the domain where hypothesis H_1 prevails and the one where H_0 prevails. By comparing the evolution of the surface with the time gap Δ, we can see that it becomes wider as Δ increases. It makes sense as when time goes by, we can expect a larger movement, so association should be allowed for larger distance features. This is further shown in Fig. 3, the zero contours of β for two different values of Δ are shown on one image of the PETS S2.L1 sequence and are centered around one detection r_0. After $\Delta = 3$ frames, any detection that falls within the blue contour will vote strongly for the association with r_0 (negative cost). After $\Delta = 15$ frames (corresponding to around 2 seconds in this case), the model is more relaxed and favors association within the green contour.

Implementation. Optimization was conducted using first a Sliding Window technique, where the labeling is done locally at each frame given the links with the past detections. At the end of the sequence, an Iterated Conditional Mode (ICM) algorithm is run over the nodes until convergence in order to correct possible mistakes. This local ICM optimization randomly visits nodes and deterministically updates their labels so as to minimize the energy.

6 Experiments

The two datasets used for experiments are described in Sec. 6.1. In both cases, the part-based detector [12] was applied to obtain the raw detections. We then introduce the performance metrics and results.

6.1 Datasets

We evaluate our approach on two datasets. The first one is CAVIAR [1]. It consists of 26 videos recorded at 25 frames per second in a monocular scene of a corridor. The average video length is 1500 frames. Challenges in this dataset arise from reflections on the floor, projected shadows, occlusions, and numerous possible entry and exit points. Dense bounding box annotation is available for the CAVIAR corpus. We test our method on a set of 20 testing videos containing a total of 140 ground truth objects, so as to be able to compare our results with [26].

The second dataset is PETS'09 [2]. We use the moderately crowded sequence S2.L1 of 795 frames designed especially for multi-target tracking evaluation. The sequence is recorded at 7 fps. Although multiple videos from cameras are available, we only performed monocular tracking using view 001. We use the available sparse ground truth annotation for the evaluation, with bounding boxes annotated every 5 frames.

6.2 Evaluation Metric

We use several types of measures to perform tracking evaluation. The first set of measures is introduced in [23]. We use tracker purity TP and object purity OP to evaluate tracking performance. Both measures take values between 0 and 1. TP gives an idea on how reliable the obtained tracks are, in terms of how much the estimates stick to the ground truth they identify. In absence of identity switches, TP is close to 1. On the other hand, OP gives an insight on how long (in percentage of length) ground truth tracks are covered by their respective single best assigned output track.

Secondly, the CLEAR MOT metrics MOTA and MOTP are used [8]. "Multi-Object Tracking Accuracy" (MOTA) combines missed detections, false positives and identity switches into a single evaluation measure. On the other hand, "Multi-Object Tracking Precision" (MOTP) gives a measure on bounding boxes localization accuracy.

Finally, a subset of the measures introduced in [18] are used. Frag is the number of times that a ground truth trajectory is interrupted in the tracking result, while IDS is the total number of identity switches, i.e. it indicates the number of times an output track successfully tracks several ground truth targets.

Note that some of these measures are correlated, insofar as they more or less attempt to quantify the same phenomenon. For example, a high fragmentation will go together with a low object purity. Nevertheless, in order to present exhaustive evaluations, quantify the benefit of our different contributions, and compare with the state-of-the-art algorithms, we use all these measures in our experiments.

6.3 Results

Results on PETS. We show the evaluation measures and the processing framerate (fps) of our tracker (without counting the time required to perform detection) for 7 different experimental setups (or conditions) to evaluate different elements of the approach: size of the association window T_w, parameter estimation and adaptation, graph simplification, optimization procedure. We use $T_w = 8$ for setups I and II, and $T_w = 15$ for the other setups. Furthermore, conditions I and III are relying on the models learned from the detections using the "closest assumption" only (cf Sec. 4.1) whereas II and IV to VII use the refined models obtained from intermediate tracklets. Note that the simplest condition I is equivalent to [16].

The results for the different conditions are summarized in Table 1. The first observation is that in all cases, the tracker purity is close to 1, which indicates that each tracklet mainly follows a single target in the scene. Comparing I and II (where T_w=8), we can notice that the refinement of the parameter estimation using tracklets has almost no effect on the performance, indicating that the "closest assumption" used to learn parameters in the default approach is valid for small T_w, as motivated in Sec. 4.1. On the other hand, for longer association windows T_w=15 it has a huge impact (see III and IV). Using the default model parameters in this case leads to precise tracklets (TP=0.98, IDS=0), but very fragmented ones (OP=0.32, Frag=28). As can be seen, this fragmentation is dramatically reduced when using the refined parameter estimates obtained from tracklets, showing the benefit and validity of our approach. Furthermore, comparing II and IV, we can observe that using longer association window sizes T_w helps having more consistent tracks, but comes at the cost of a slower processing rate, because of the presence of much more links in the graph. Indeed, there is a street light in the middle of the scene, and people tend to get occluded by it (and are thus not detected) for quite some time. Also, inter-person occlusions are also frequent. Thus, having a larger T_w can help recovering from these long occlusions.

We observed so far that the best tracking results are obtained for T_w=15, with refined parameter estimates. This is the baseline used in setups V, VI and VII. A first modification that we used to reduce computation is to remove edges in the graph that might not be so informative. This is achieved by dropping links between pairs of detections further apart than a large threshold on the distance, which mainly discourages improbable associations. This is what is done in setup V. As can be seen, while tracking performance is unchanged, a significant speed-up is obtained (4.2 fps vs 2.5 fps). While preserving the accuracy brought by a higher T_w, we can even be faster than I and II in which $T_w = 8$.

Table 1: Performance evaluation for different experimental setups. PET (Parameters Estimated from Tracklets) refers to the case where the global model parameters were estimated from intermediate tracklets results, as described in Sec. 4.2, while Context. adapt. refers to the local contextual adaptation (cf Sec. 4.3).

	$\mathbf{T_w}$	**PET**	**Dropped links**	**Context. adapt.**	**Block opt.**	TP	OP	MOTA	MOTP	Frag	IDS	fps
I [16]	**8**	**NO**	**NO**	**NO**	**NO**	0.95	0.67	0.77	0.67	11	1	3.5
II	**8**	**YES**	**NO**	**NO**	**NO**	0.95	0.67	0.76	0.67	12	1	3.5
III	**15**	**NO**	**NO**	**NO**	**NO**	0.98	0.32	0.90	0.66	28	0	1.9
IV	**15**	**YES**	**NO**	**NO**	**NO**	0.95	0.86	0.74	0.66	4	3	2.5
V	**15**	**YES**	**YES**	**NO**	**NO**	0.95	0.86	0.74	0.66	4	3	4.2
VI	**15**	**YES**	**YES**	**YES**	**NO**	0.95	0.85	0.73	0.66	4	3	1.5
VII	**15**	**YES**	**YES**	**NO**	**YES**	0.98	0.89	0.89	0.66	3	1	1.45

Table 2: Comparison with state-of-the-art techniques

	MOTA	MOTP	IDS	TP	OP
[22]	-	-	9	0.62	0.65
[3]	0.89	0.56	10	-	-
Ours	0.89	0.66	1	0.98	0.89

Table 3: Tracking performance on CAVIAR

	No.GT	IDS	Frag
Ours w/o context adaptation	140	26	100
Ours w. context adaptation	140	11	74
[26] algo 1	140	7	58
[26] algo 2	140	15	20

In the setup VI, we add our contextual adaptation scheme in the framework. To the contrary of what is observed on other datasets (see later for CAVIAR for instance) parameter adaptation does not improve the tracking performance on this dataset.

For all setups I to VI, optimization was conducted locally by only updating labels one node at a time. In VII, we further add to the optimization a block ICM step inspired by [10]. In brief, in this method the objective function remains the same, but the granularity is different as we add the constraint that all labels inside blocks (defined as the nodes with the same label in the current results within a window of duration $2 \times T_w$) should be changed (or not) simultaneously. One sweep of such block ICM through the whole sequence is enough to correct 2 identity switches. The remaining IDS happens in frame 207 when a person exits the field of view and its label is 'caught' by another person entering immediately after in a close-by location.

Finally, we compare in Table 2 our tracking results with two state-of-the-art multi-person trackers [22, 3]. As can be seen, we outperform these methods and achieve a much lower number of identity switches than both approaches which is also reflected by the high TP we get. At the same time, we are able to maintain longer (less fragmented) tracks, which can be seen in our high OP.

Figure 4 shows the output of our tracking algorithm on PETS data. We can observe that despite the occlusions and people passing close to each other, we are able to maintain correct labels, even when working with in a single view.

Results on CAVIAR data. They are shown in Table 3, where T_w was chosen to be 1 second for our experiments. There, on the contrary to the PETS sequence, we observe a beneficial effect of the contextual adaptation step, which is able to correct some identity switches and reduce the track fragmentation. Figure 5 illustrates the effect of this contextual adaptation. For this person, its current and previous locations are shown by the black dots. The red contour is the zero iso-line of the global β for $\Delta = 1$, and it can be noticed that the current detection does not lie inside the area this red contour defines. However, given the local context (the current detection is the closest to the point, and the second closest is quite far apart), the model is adapted and the new contour for $\beta = 0$ is show in green. We can observe that while the previous model was too strict, the adapted one now allows for association.

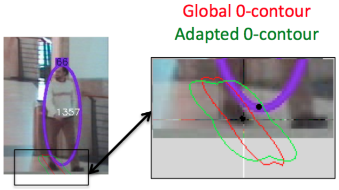

Figure 5: Example of contextual adaptation effect on CAVIAR. Black dots denote the previous and current detection locations.

We observed a wide disparity of the performance among different videos. In fact, the results depend on the complexity of the scene, namely on the number of people which cause more occlusions. We also compared our results with the two algorithms proposed by [26]. In algorithm 2, they use the same network flow formulation as in algorithm 1 but add an explicit occlusion model. Even though our association horizon is of only 1 second, it is enough to get comparable results to [26] in terms of fragmentation and IDS.

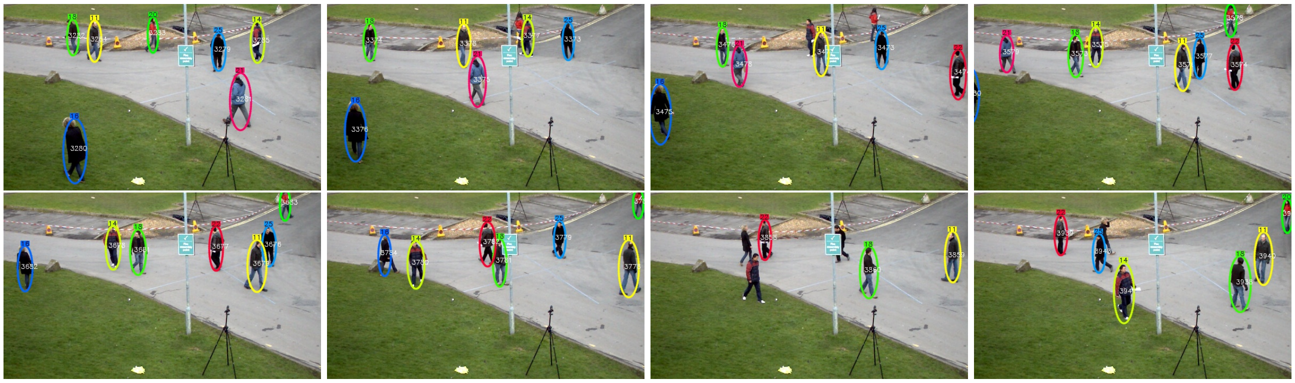

Figure 4: Tracker output on some frames of PETS S2.L1 view 001 (shown every 15 frames).

7 Conclusion

We used a CRF model for detection-based multi-person tracking, in which the task of tracking is formulated as a labeling process. Within this framework, we presented an automatic model parameter estimation approach relying on intermediate tracking results. Moreover, we presented a way of adapting model parameters based on local spatiotemporal context. We showed that we are able to obtain accurate tracks both on the PETS 2009 sequence and on CAVIAR data. Provided we relearn our model parameters as explained above, we showed that increasing T_w allows to recover from longer occlusions. However, there is a temporal limit above which our pairwise association may not bring any information anymore. To solve longer occlusions or more complex ambiguities in the case of larger crowdings, we will need to exploit other cues or cost terms, for example by using global costs on trajectories, using motion similarity, or modeling entry/exit maps of the scenes, for instance.

Acknowledgments

This work has been funded by the Integrated Project VANAHEIM (248907) supported by the European Union under the 7^{th} framework program. The authors would like to thank Cheng Chen for useful discussions.

References

[1] http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/.

[2] http://www.cvg.rdg.ac.uk/PETS2009/.

[3] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012.

[4] B. Song, T.-Y. Jeng, E. Staudt, and A. K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *ECCV*, 2010.

[5] B. Yang, C. Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a CRF model. In *CVPR*, 2011.

[6] B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In *CVPR*, 2012.

[7] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *Winter-PETS, 2009*.

[8] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *J. Image Video Process.*, 2008.

[9] C. Chen, A. Heili, and J.-M. Odobez. A joint estimation of head and body orientation cues in surveillance video. In *ICCV, SISM Workshop*, 2011.

[10] R. T. Collins. Multitarget data association with higher-order motion models. In *CVPR*, 2012.

[11] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, 2010.

[12] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010.

[13] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, November 1984.

[14] H. Gong, J. Sim, M. Likhachev, and J. Shi. Multi-hypothesis motion planning for visual object tracking. In *ICCV*, 2011.

[15] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011.

[16] A. Heili, C. Chen, and J.-M. Odobez. Detection-based multi-human tracking using a CRF model. In *ICCV VS Workshop*, 2011.

[17] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*, 2008.

[18] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybrid-boosted multi-target tracker for crowded scene. In *CVPR*, 2009.

[19] M. Rodriguez, I. Laptev, J. Sivic, and J. Audibert. Density-aware person detection and tracking in crowds. In *ICCV*, 2011.

[20] M. Yang, Y. Wu, and G. Hua. Context-aware visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009.

[21] T. Parag, F. Porikli, and A. M. Elgammal. Boosting adaptive linear weak classifiers for online learning and tracking. In *CVPR*, 2008.

[22] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *ICCV*, 2011.

[23] K. Smith, D. Gatica-Perez, S. Ba, and J. Odobez. Evaluating multi-object tracking. In *CVPR EEMCV Workshop*, San Diego, June 2005.

[24] X. Song, X. Shao, H. Zhao, J. Cui, R. Shibasaki, and H. Zha. An online approach: learning-semantic-scene-by-tracking and tracking-by-learning-semantic-scene. In *CVPR*, 2010.

[25] J. Yao and J.-M. Odobez. Multi-camera multi-person 3D space tracking with MCMC in surveillance scenarios. In *ECCV, M2SFA2 Workshop*, 2008.

[26] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR, 2008*.