Multi-target tracking by learning local-to-global trajectory models

Shun Zhang, Jinjun Wang *, Zelun Wang, Yihong Gong, Yuehu Liu
Institute of Artificial Intelligence and Robotics, Xi’an Jiaotong University, Xi’an, Shaanxi 710049, China

ABSTRACT

The multi-target tracking problem is challenging when there exist occlusions, tracking failures of the detector and severe interferences between detections. In this paper, we propose a novel detection based tracking method that links detections into tracklets and further forms long trajectories. Unlike many previous hierarchical frameworks which split the data association into two separate optimization problems (linking detections locally and linking tracklets globally), we introduce a unified algorithm that can automatically relearn the trajectory models from the local and global information for finding the joint optimal assignment. In each temporal window, the trajectory models are initialized by the local information to link those easy-to-connect detections into a set of tracklets. Then the trajectory models are updated by the reliable tracklets and reused to link separated tracklets into long trajectories. We iteratively update the trajectory models by more information from more frames until the result converges. The iterative process gradually improves the accuracy of the trajectory models, which in turn improves the target ID inferences for all detections by the MRF model. Experiment results revealed that our proposed method achieved state-of-the-art multi-target tracking performance.

Crown Copyright © 2014 Published by Elsevier Ltd. All rights reserved.

1. Introduction

Vision based multi-target tracking aims at locating all targets of interest, inferring their trajectories and maintaining their identities from image observations in a video sequence. It is important for many computer vision applications such as video surveillance, robotics and activity analysis. Compared with single target tracking task, multi-target tracking is significantly more difficult because it has to face the following challenging problems. First, targets may be partially or completely occluded by other targets or foreground objects and become unobservable for a short time, which may confuse tracking algorithms and may result in target losses or ID switches between the targets. Second, since targets may enter or leave the scene at any moment, the number of targets is usually unknown and may vary over time. And last but not least, it is difficult to distinguish targets with similar appearances, especially when they are close to, or partially occlude each other.

In recent years, many of the successful tracking methods perform tracking by detection. These methods apply a pre-learned object detector to conduct object detections in every frame through the video stream. The tracking problem is solved by the data association [1–8] which strives to establish a unique identity for each target, and to simultaneously estimate the motion patterns of all targets and the assignment of detections to targets, by linking similar detections across frames. This method is essentially more flexible and robust in complex environments, since it can not only naturally handle re-initialization in tracking when a target is lost, but also avoid excessive model drift. However, it poses another difficult challenges. Since the detector’s output is only partly reliable, missed detections (false negatives) and incorrect detections (false positives) may happen frequently in the detection process, which provides misleading information to association algorithms. If the targets are overlapped, the task is further complicated and it is much more difficult to retrieve the real targets among those detections and assign the labels of detections for each of them in every frame.

To deal with this data association problem, many of the previous methods [9–14] associated the detections locally, i.e. using local information from a few neighboring frames or frame by frame. These methods generally integrated several cues of the image information such as appearance, motion, size and location [15–19] to measure the similarity between detections from two consecutive frames. Given only the image information in a small time window, the local association methods are difficult to tackle the long-term occlusion due to the ambiguous and noisy observations, which make them incline to result in tracking failures (e.g. trajectory fragmentation and identity switches).

In contrast to the local tracking methods, many of the latter approaches [20–22,23,24] have achieved great progress by using global inference over all trajectories simultaneously in a longer period. As they consider more global information, these association

* Correspondence author. Tel./fax: +86 29 83395146.
E-mail address: jinjun@mail.xjtu.edu.cn (J. Wang).

http://dx.doi.org/10.1016/j.patcog.2014.08.013
0031-3203/Crown Copyright © 2014 Published by Elsevier Ltd. All rights reserved.

Please cite this article as: S. Zhang, et al., Multi-target tracking by learning local-to-global trajectory models, Pattern Recognition (2014),
http://dx.doi.org/10.1016/j.patcog.2014.08.013
approaches can do better to overcome errors by detections and association. With the increasing of the frame numbers, however, the hypothesis search space of those alternatives grows exponentially, so that these global association algorithms usually require lots of computation which make them unsuitable for real-time processing. Moreover, they also typically assume that all detections are correct which is not always accurate [7].

Some more recent approaches [25–29] combined the local linking method and the global association method in a kind of hierarchical framework, which first linked easy-to-connect detections in neighboring frames into tracklets, and then progressively linked tracklets into longer ones based on some global association approaches, such as the Hungarian algorithm [30] and the network flow [31]. They presented a formulation with two separate optimization problems: linking detections are solved using the local information in lower stages and linking tracklets are solved using the global information in higher stages. Splitting the problem in two phases has, obviously, several disadvantages are solved using the local information in lower stages and linking tracklets are solved using the global information in higher stages. Furthermore, transforming from the local process to the global process is generally specified heuristically and experimentally.

In this paper, we propose a unified multi-target tracking method to progressively grow local tracklets into long trajectories by combining the local and global information. Unlike many previous hierarchical methods which heuristically formulate the data association method as two separate optimization schemes, we present a unified algorithm for finding the joint optimal assignment. We online learn the trajectory model (TM) for each target to maximize a joint object function. By repeating the proposed algorithm, the trajectory model can be learned more accurately from local to global information. The trajectory model is composed of a set of cues including appearance, velocity, size, and position. We segment the whole long video stream into multiple non-overlapping sliding windows. For each temporal sliding window, the trajectory models are firstly initialized by very limited, local image information (e.g. information from the first frame). The initial trajectory models are used to link those visually resembling, easy-to-connect detections in neighboring frames into separated tracklets. Then the appearance, motion and some other parameters of trajectory models are updated by those reliable tracklets, such that the trajectory models become more accurate. The updated trajectory models are reused to link the separated tracklets that belong to the same target. As the iterative process continues, the trajectory models become accurate, and the broken tracklet get connected to form longer trajectories.

The main contributions of this paper include the following:

- A unified framework which online learns the local-to-global trajectory models.
- An iterative algorithm to alternately update the trajectory models and link detections or tracklets into longer fragments.
- Formulating the data association problem as inferences of target IDs for all the detections using the MRF model.
- And employing the loopy belief propagation (LBP) algorithm to optimize the MRF model so as to generate separated tracklets, which has low complexity compared with most state-of-the-art methods.

Our experimental results show superior tracking performance on several public datasets and computational speed of the proposed method.

The rest of this paper is organized as follows. Section 2 briefly describes related works in the literature. Section 3 introduces an overview of our approach. Sections 4 and 5 present the proposed multi-target tracking framework and its inference algorithm, respectively. Section 6 provides the performance evaluation and comparison results, and Section 7 concludes the paper.

2. Related work

Fostered by the recent progress in object detection techniques such as [33–35], there is a rich body of multi-target tracking works based on tracking by data association approaches. They applied an object detector learned off-line or on-line to yield the per-frame detections. However, the detector posed many misleading errors including missed detections, false alarms and inaccurate detections. Detection based tracking method must overcome the failures of the detector, and the difficulties caused by occlusions, initialization and termination of targets and similar appearance among multiple targets. To overcome the data association problem, there are three main strategies: associating the detections locally, associating them globally, and combining the local method and global association method.

Much effort [9–14] has been made to deal with the data association problem locally. A frame-by-frame tracking method in [9] presented an affinity measure between detections based on cues from position, size, and color and used a greedy algorithm to match the detection responses and hypotheses. The trajectory initialization and termination were based on the evidence collected from the detection responses. This association method depends a lot on the detector and easily results in tracking failures when the targets are occluded for a long period. In [10], a mixture particle filter method was used to associate the Adaboost detections of an unknown number of objects. It assigned a mixture particle filter to each target, and constructed the proposal distribution for the particle filter from a mixture of the Adaboost detections in the current frame and the dynamic model predicted from the previous time step. The detection responses were used to generate new particles and evaluate existing particles. However, increasing the number of particles requires more computational cost. The local association methods are likely to lead to drift when multiple targets are close to each other, since the noisy target detections significantly increase the difficulty of data association.

To consider associations beyond the local basis, a rich body of data association approaches propose inference over multiple targets by seeking to resolve the drift problem in a longer period. Multi-Hypothesis Tracking (MHT) [36] and Joint Probabilistic Data Association Filters (JPDAF) [37] are among the earliest widely used techniques for global data association. They maintain multiple hypotheses until enough evidence can be collected to resolve the ambiguities. Recently, a variety of global data association approaches [20–22,27,23,24] which try to simultaneously optimize all trajectories by diverse optimization algorithms have been developed. For example, [24] used Viterbi algorithm to get optimal object sequences, [38,39] used the Hungarian algorithm to simultaneously optimize all trajectories, [22] used Quadratic Boolean Programming to couple the detection and estimation of trajectory hypotheses, [2] formulated multi-target tracking as the maximum weight-independent set problem, [21,40] proposed a linear programming approach to search multiple paths, [41,42] involved the minimization of complex energy functions and relied on brute-force algorithms to search for locally optimal solutions, and [7] used a min-cost network flow to model the MAP data association problem. Although these approaches have been demonstrated to improve tracking performance, they are computationally exponential both in memory and time.

The hierarchical association framework [27,43,44,29,25,28] combined the local linking methods and global association methods for better tracking performance in detection based tracking literature. These methods tackled the tracking problem by progressively connecting short detections or tracklets into longer ones. They typically split the data association into two separate optimization problems: linking detections locally in lower stages and linking tracklets globally in higher stages.
For example, [26] proposed a two-stage association method, which combined local tracklets filtering and global tracklets association to track multiple targets through occlusions. The particle filter was used to generate a set of reliable tracklets in the local stage and a modified Hungarian algorithm was used to optimize the data association in the global stage. Moreover, [25] proposed a three-level association approach: a two-threshold strategy was used to link the detections from consecutive frames into short tracklets at the bottom level, and the iterative Hungarian algorithm and EM-like algorithm were used at the higher levels. [44] extended the three-level work by online learning a more discriminative appearance model based on several image descriptors instead of measuring the distance of two holistic histograms. [28] employed the similar hierarchical framework for tracking sports players with the Hungarian algorithm and the cost flow algorithm replaced at the higher levels.

In this paper, we propose a novel association-based tracking method to combine the local and global information for better tracking performance. Unlike many hierarchical methods which split the data association into two separated optimization problems, we establish a unified framework based on the introduced trajectory models for automatically relearning from the local to global information. The local-to-global trajectory models can be used to link detections from consecutive frames into tracklets, and also link separated tracklets that belong to the same targets into long trajectories. The proposed method achieves state-of-the-art performance with relatively low computational complexity of $O(S \times N^2)$ within a given sliding window, where $S$ is the iteration times of trajectory model learning and $N$ is the number of detection nodes.

### 3. Outline of our approach

As illustrated in Fig. 1, our paper presents a multi-target tracking method to automatically link the detections or tracklets into trajectories based on a novel local-to-global trajectory model. At the beginning of each sliding window, the trajectory models are initialized by the local information from the first frame or from the previous sliding window. We introduce a pairwise Markov Random Field (MRF) model to infer target IDs for all detections in the sliding window and employ the loopy belief propagation algorithm to solve the joint object function – maximal conditional probability of MRF model. The detections with the same label in adjacent frames are linked to form reliable tracklets. Then the trajectory models are updated by the reliable tracklets. Meanwhile, the number of trajectory models can be reset to eliminate the false models caused by false alarms and add new models for newly emerging targets. We alternatively optimize the trajectory models for all targets and maximize the conditional probability of the MRF model until the result converges. As the MRF model yields more complete target trajectories, we can build more accurate trajectory models with more global information collected from more frames. The more accurate trajectory models will contribute to more accurate probabilistic computations of the MRF model, which in turn will yield better data association solutions.

### 4. Proposed model

Generally the multi-target tracking problem can be formulated as inferences of target IDs for all detections in a given time window (the length of window is $T_w$). Let $Y = \{y_1, y_2, \ldots, y_N\}$ be a set of detections and $L = \{l_1, l_2, \ldots, l_N\}$ be their labels (target IDs). Our overall goal is to find the optimal assignment for the identity of targets based on the detection set. It is equivalent to maximize the conditional probability $P(L|Y)$ of the MRF model as shown in Fig. 2, where for each node $i$, $y_i$ and $l_i$ correspond to its observation and its state to be estimated, respectively. Assume that there are $K$ targets in the scene, then $l_i \in \{1, \ldots, K\} \cup \varnothing$, where $\varnothing$ denotes false detections. Using this model, $P(L|Y)$ is defined as

$$P(L|Y, \Gamma) = \frac{1}{Z_p} \prod_{i=1}^N \Phi(l_i, y_i; \Gamma) \prod_{ij} \Psi(l_i, l_j, y_i, y_j; \Gamma),$$

where $Z_p$ is the normalization factor.

In Eq. (1), the probability depends on a hyper-parameter set $\Gamma$. It is composed of the trajectory models for all targets, $\Gamma = \{\tau_1, \ldots, \tau_K\}$. Each $\tau_k \in \Gamma$ is defined as $\tau_k = [\mu_k, \sigma_k, \Gamma_k, T_k]$, where $\tau_k = [\mu_k, \sigma_k] \Gamma_k$ denotes the position parameters of target $k$ that include the initial position $p_{\mu_k}$, the Kalman Filter parameters with the transition matrix $T_k$ and observation matrix $O_k$, and the variance $\sigma_k$. $T_k = [\mu_k, \sigma_k]$ denotes the mean and variance of its $(d', d^2)$ velocity, $\tau_k = [\mu_k, \sigma_k]$ denotes the scalar mean and variance of its size, and $\Gamma_k$ represents a target-specific classifier that is trained using the previous detections, and consequently used to classify the new detections. We also represent each detection with $y_i = [y_i', y_i^2', y_i^3']$, including its $(d', d^2)$ position $y_i'$, velocity $y_i^2'$, appearance $y_i^3'$, and size $y_i^4$.
The unary term $\Phi(l_j, y_i; \Gamma)$ in Eq. (1) describes how the hidden state value $l_j$ fits the observation $y_i$. $\Phi(\cdot)$ is evaluated by a generative probability $P(y_i | l_j; \Gamma)$. Assuming the independence among the attributes, $P(y_i | l_j; \Gamma)$ can be factorized as

$$P(y_i | l_j = k; \Gamma) = P(y_i | r_{t_k}) = P(y_i | r_{t_k}^i)P(y_i | r_{t_k}^e)P(y_i | r_{t_k}^v),$$

where $k = \{1, \ldots, K\}$. To identify false detections, we compute the probability $P(y_i | l_j = \emptyset; \Gamma)$ by

$$P(y_i | l_j = \emptyset; \Gamma) = 1 - \sum_{k=1}^{K} P(y_i | l_j = k; \Gamma)$$

Using the trajectory models $\Gamma$, the probabilities in Eq. (2) are defined as

$$P(y_i | r_{t_k}^e) = H(y_i | r_{t_k}^e), \quad P(y_i | r_{t_k}^v) \sim \mathcal{N}(\mu_k; \sigma_k^v),$$

where $H(y_i | r_{t_k}^e)$ is the decision function for target-specific classifier $r_{t_k}^e$, and $\mathcal{N}(\cdot)$ denotes a Gaussian distribution. $P(y_i | r_{t_k}^v)$ is computed using $\tilde{p}_k^i$ which denotes the estimated position of target $k$ at frame $t$. It is estimated by Kalman Filter $\tilde{p}_k^i = f(p_k^i, t; T_i, O_k)$ using the parameters in $t_k^i$ \[45\].

The pairwise term $\Psi(l_i, l_j, y_i, y_j; \Gamma)$ in Eq. (1) defines the probability that two adjacent nodes possess the same label, given as follows:

$$\Psi(l_i, l_j, y_i, y_j; \Gamma) = \begin{cases} 
\Psi_1(l_i, l_j, y_i, y_j; \Gamma) & \text{if } t_i = t_j + 1, \\
\Psi_2(l_i, l_j, y_i, y_j; \Gamma) & \text{if } t_i = t_j, \\
0 & \text{otherwise},
\end{cases}$$

where $t_i$ denotes the frame number from which the observation of node $i$ is detected. The pairwise term is defined on such a neighborhood system in which the neighborhood $\mathcal{N}(i)$ of node $i$ consists of all the nodes from both the previous frame $t_i - 1$, the next frame $t_i + 1$ and all other nodes within the same frame $t_i$. In Eq. (4), different types of neighbors are treated differently. $\Psi_1(\cdot)$ treats the neighbors from the previous and the next frames $t_i \pm 1$, given by

$$\Psi_1(l_i, l_j, y_i, y_j; \Gamma) = \delta(l_i - l_j) \cdot \text{sim}(y_i, y_j; \Gamma) + (1 - \delta(l_i - l_j)) \cdot (1 - \text{sim}(y_i, y_j; \Gamma))$$

where $\delta(\cdot)$ is the delta function that takes value 1 when its argument is zero and 0 otherwise. $\text{sim}(y_i, y_j; \Gamma)$ is an affinity model that defines the similarity of two detections, and is factorized in the same way as Eq. (2):

$$\text{sim}(y_i, y_j; \Gamma) = \text{sim}(\cdot | l_i = k; \Gamma) = P(y_i^f | r_{t_k}^f)P(y_i^e | r_{t_k}^e)P(y_i^v | r_{t_k}^v).$$

It is seen that Eq. (5) encourages assigning the same label to the detections with high similarity scores.

$\Psi_2(\cdot)$ treats the neighbors from the same frame $t_i$, and is defined as

$$\Psi_2(l_i, l_j, y_i, y_j; \Gamma) = \begin{cases} \alpha & \text{if } l_i = l_j \in \{1, \ldots, K\}, \\
\beta & \text{otherwise},
\end{cases}$$

where $\alpha$ is a small constant value to penalize the assignment of the same label to two detections in the same frame, and $\beta(\geq \alpha)$ is the probability that two detections are assigned different labels, or at least one detection is assigned false positive $\emptyset$. Obviously, this function promotes different target assignments to detections in the same frame, which enforces the intuition that the same target cannot appear in more than one place at the same time.

5. Model inference

In this section, we present an iterative algorithm to automatically learn trajectory models from the local to global information. It alternatively optimizes the trajectory models for all targets and maximizes the conditional probability of MRF model, as shown in Fig. 1 of Section 3. For a long video sequence, we break it into multiple non-overlapping short sliding windows. At the beginning of the tracking process, since there is no corresponding information among the detections from different frames, the number of targets in the scene as well as the initial trajectory models can only be estimated using the detection result from the first frame. Target ID inferences with an MRF model defined by such local trajectory models only yield fragmented tracklets. These tracklets tend to link those visually resembling, easy-to-connect detections in neighboring frames. Meanwhile, they are relatively reliable segments of the final trajectories, thus using these intermediate tracking results to re-update the trajectory models with more global information from more frames, we can build better trajectory models. The better trajectory models can be reused to define better MRF model, which in turn will yield better target trajectories. This alternative process is repeated until the results converge. The following subsections describe the local-to-global learning process in more details.

5.1. Initializing the local trajectory models

The initialization of the trajectory models has to handle the following two tasks: (1) initializing the trajectory models $\Gamma$ at the beginning of the tracking task, i.e. in the first iteration of the first sliding window; and (2) initializing $\Gamma$ every time when sliding the analysis windows, i.e., in the first iteration of all analysis windows except the first one.

For initializing $\Gamma$ at the beginning of the tracking task, we set the number of trajectories $K$ equal to the number of detections at the first frame. For each $r_k \in \Gamma$, its parameters $\mu_k^v$ and $\mu_k^e$ are initialized using the corresponding attributes $y_k^v$ and $y_k^e$ of detection $y_k$. $p_k^i$ is set to $\tilde{y}_k$ and $t_k^v$ is a initial classifier trained by the positive samples collected around the position $y_k^v$ and the negative samples collected around the other targets. The variance parameters $\sigma_k^v$ and $\sigma_k^e$ are specified heuristically based on experiments, while $T_k$ and $O_k$ are initially defined as a linear model.

For initializing $\Gamma$ when sliding the analysis windows, we use the number of finally obtained trajectory models from the precedent sliding window to initialize $K$, and use the final trajectory models from the precedent window to initialize the trajectory models in the current window.

5.2. Maximization of MRF conditional probability

The subsection introduces the maximization solution for conditional probability $P(\mathbf{L} | \mathbf{Y}, \Gamma)$ using the MRF model whose generative
and link probabilities are defined in Section 4 by the established trajectory models $\Gamma$. We employ the sum-product loopy belief propagation (LBP) algorithm [46] which computes the marginal distribution by iteratively passing messages between neighbors. Let $m_{ij}(l_i)$ denote the message from node $j$ to node $i$ as "how confident node $j$ believes node $i$ should take value $l_i"$. The messages are updated according to the following rules:

$$m_{ij}(l_i) = \sum_{l_j} \mathcal{Q}(l_j, y_j; \Gamma) \mathcal{Q}(l_i, y_i, y_j; \Gamma) \prod_{q \in N(j) \setminus i} m_{qj}(l_j),$$

(8)

where $N(j) \setminus i$ represents all the neighborhood nodes of $j$ except for node $i$, as shown in Fig. 3(a).

The belief $b(l_i)$ at a node $i$ can be calculated through a belief-readout process by

$$b(l_i) = \frac{1}{Z_b} \mathcal{Q}(l_i, y_i; \Gamma) \prod_{q \in N(i)} m_{qj}(l_j),$$

(9)

where $Z_b$ normalizes $b(l_i)$ into a probability score, as shown in Fig. 3(b). Belief is the estimated marginal probability, and a high value of $b(l_i)$ means the marginal value that node $i$ belong to the state $l_i$ is high.

During the LBP inference process, we update one node at a time and sweep all the nodes by a “left-right fashion”[47]. The BP message-update equations are iterated until they converge, then the beliefs can be read out from Eq. (8). In order to select confident nodes, we set a threshold for $b(l_i)$, i.e. node $i$ will be assigned label $k$ when $b(l_i, \tau_\tilde{\tau}) > T_b$. Thus the nodes with the same label $k$ in adjacent frames are linked to form tracklet $TL_k$, which is relatively reliable segment of the final target trajectory.

5.3. Local-to-global trajectory model learning

After maximizing the MRF conditional probability by the LBP algorithm and generating a set of confident and separated tracklets, the trajectory model learning has to handle the following two tasks: (1) updating the number of trajectory models $K$ to accommodate false positive detections and newly emerging targets; and (2) updating $\Gamma$ by the reliable tracklets.

Task (1) is accomplished as follows. We delete $\tau_k$ from $\Gamma$ whose tracklets are shorter than a threshold $T_{len}$. This aims to eliminate tracklets formed by false positive detections, which are unstable, random, and inconsistent, and will not form long tracklets under the proposed MRF model. The trajectory models corresponding to targets leaving the scene will not be deleted because their tracklets are generally longer than $T_{len}$.

When a new target emerges, there is no appropriate $\tau_k$ to model it, thus it is more likely to be assigned label $\emptyset$ by the LBP algorithm. After each detection $y_i$ has been assigned a label $l_i \in \{1, \ldots, K\} \cup \emptyset$, we apply a low level association process [25] to see if the detections with label $\emptyset$ can be linked to form a valid tracklet of longer than $T_{len}$ frames. If any such tracklets can be obtained, we use Eq. (10) to build $\tilde{\tau}$ for it, and check if $\tilde{\tau}$ is different from any existing $\tau_k \in \Gamma$. If $\tilde{\tau}$ is sufficiently different, it will be added into $\Gamma$. The only reason why $\tilde{\tau}$ could be similar to an existing $\tau_k$ is that a long trajectory is incorrectly broken into segments, and one of these segments is incorrectly labeled as $\emptyset$ by the LBP algorithm when the trajectory model is not too accurate. In this case, we simply discard $\tilde{\tau}$ and update $\tau_k$’s parameters using Eq. (10). As our experiments have shown, with such trajectory model updates, this type of detections labeled as $\emptyset$ will eventually disappear in the subsequent iterations.

For Task (2), to update $\tau_k \in \Gamma$, we select the tracklet which is composed of the nodes assigned label $k$. We also require that the starting frame of the tracklet should be in the first several frames of the analysis window. As shown in Fig. 4, two targets intersect with each other in the middle of their paths and the purple target is occluded by the green one. As the initial trajectory model is not too accurate, the purple target’s trajectory is broken into two separated tracklets $TL_2$ and $TL_3$ (Fig. 4(C)), where $TL_3$ is given an incorrect label. In the next iteration, as the tracklets $TL_1$ and $TL_3$ start in the first several frames of the analysis window, they are used to update the respective trajectory models (Fig. 4(D)). By the next LBP algorithm, the purple trajectory model still link detections
into the tracklet $\mathbf{T}_{L_2}$, and it fits well with those nodes forming $\mathbf{T}_{L_3}$ in terms of their attributes. Thus $\mathbf{T}_{L_2}$ are very likely to be assigned the same label as $\mathbf{T}_{L_2}$ (Fig. 4(E)). In this way, the broken tracklets get reconnected in the iterative process.

Denote the tracklet with label $k$ as $\mathbf{T}_{L_k} = \{\mathbf{y}_{k1}, \mathbf{y}_{k2}, \ldots, \mathbf{y}_{kN_k}\}$. The parameters $\tau_k^i$ and $\tau_k^v$ are updated as follows:

\[
\mu_k^i = \frac{1}{N_k} \sum_{j=1}^{N_k} y_{kj}^i, \quad \sigma_k^i = \sqrt{\frac{1}{N_k} \sum_{j=1}^{N_k} (y_{kj}^i - \mu_k^i)^2};
\]

\[
\mu_k^v = \frac{1}{N_k} \sum_{j=1}^{N_k} y_{kj}^v, \quad \sigma_k^v = \sqrt{\frac{1}{N_k} \sum_{j=1}^{N_k} (y_{kj}^v - \mu_k^v)^2}.
\]

(10)

For $\tau_k^i$, we use the dataset in $\mathbf{T}_{L_k}$ to update the Kalman Filter parameters $\mathbf{T}_k$ and $\mathbf{O}_k$ using the learning algorithm in [48]. We use a Kalman Filter to model the data sequences. Here, $\sigma_k^v$ is updated by

\[
\sigma_k^v = \sqrt{\frac{1}{N_k} \sum_{j=1}^{N_k} (y_{kj}^v - \mu_k^v)^2}.
\]

(11)

Subsection 5.4, Classifier construction and update

As mentioned in Section 4, $\tau_k^v$ is a target-specific classifier which is trained online by the detections and consequently used to score the new detections. Our online learning process consists of three blocks: (i) a classifier $\tau_k^v$ for the $k$th target to be learned; (ii) a collection of labeled training examples $S = \{(x, y)\}$ as training set, where $x$ is an example from a feature space $\mathcal{X}$ and $y$ is a label from a space of labels $\mathcal{Y} = \{0, 1\}$; and (iii) a supervised training method $c(\cdot): \mathcal{X} \rightarrow \mathcal{Y}$ that trains a classifier $c$ from labeled training set.

As mentioned in Section 5.1, we initialize a classifier for each detection of the first frame. We select 10 image patches that are closest to the detection response $\mathbf{y}_0^i$ and add them to the positive sets. The negative samples are collected from detection responses of different targets. In addition, for retraining the classifier $\tau_k^v$ from detection responses in reliable tracklets, we collect the new positive samples from the confident tracklet $\mathbf{T}_{L_k}$ and select 10 image patches near each detection of $\mathbf{T}_{L_k}$. Similarly, new negative samples are collected from detection responses of different targets.

Each sample $x \in \mathbb{R}^m$ is represented by a low-dimensional feature $\mathbf{v} = (v_1, \ldots, v_q) \in \mathbb{R}^q$ with $m \gg n$ as adopted in [49]. The features are modeled with naive Bayesian classifiers [50],

\[
\tau_k^v = H(\mathbf{v}) = \log \left( \frac{\prod_{i=1}^{q} p(v_i | y = 1)p(y = 1)}{\prod_{i=1}^{q} p(v_i | y = 0)p(y = 0)} \right)
\]

\[
= \sum_{i=1}^{q} \log \left( \frac{p(v_i | y = 1)}{p(v_i | y = 0)} \right)
\]

(12)

where we assume uniform prior $p(y = 1) = p(y = 0) = 0.5$. As explained in [49], the conditional distributions $p(v_i | y = 1)$ and $p(v_i | y = 0)$ in the classifier $H(\mathbf{v})$ are assumed to be Gaussian distributed with four parameters $(\mu_i^v, \sigma_i^v, \mu_i^o, \sigma_i^o)$ where

\[
p(v_i | y = 1) \sim N(\mu_i^v, \sigma_i^v), \quad p(v_i | y = 0) \sim N(\mu_i^o, \sigma_i^o)
\]

(13)

\[
\mu_i^v = \lambda \mu_i^o + (1 - \lambda) \mu_i^v, \quad \sigma_i^v = \sqrt{\lambda (\sigma_i^o)^2 + (1 - \lambda) (\sigma_i^v)^2 + \lambda (1 - \lambda) (\mu_i^o - \mu_i^v)^2}
\]

(14)

$\lambda$ is the learning rate (empirically set to 0.85). $\rho_i^1 = 1/\text{num}$ \(\sum_{k=1}^{N_k} y_{kj}^i\) and $\sigma_i^1 = \sqrt{1/\text{num} \sum_{k=1}^{N_k} (y_{kj}^i - \mu_{ij}^1)^2}$ are the average and standard deviation of the $i$th feature extracted from $\text{num}$ positive samples, respectively. The update rules for $\mu_i^o$ and $\sigma_i^o$ are similarly defined. For each target in the current sliding window, the reliable tracklets corresponding to the targets are used for feature extraction and the classifier update. The updated classifiers are used for scoring the similarity probabilities between the appearance of all detections and the trajectory models.

Algorithm 1. Local-to-global trajectory model learning.

Input: A set of detections $Y$ yielded by the object detector; the analysis window length $T_w$; the number of analysis windows $L$; the number of iterations $S$ in each window.

For $t = 1 \rightarrow L$

for $\text{iter} = 1 \rightarrow S$

if $\text{iter} = 1$

if $t = 1$

Initialize the trajectory models by the detections from the first frame at the beginning of the tracking task.

else

Initialize the trajectory models when sliding the temporal windows:

$\Gamma_{t+1} = \Gamma_t$: reset $p_k^i$ for each $\tau_k \in \Gamma_{t+1}$.

else

Add and delete trajectory models.

Relearn the parameters of trajectory models by the reliable tracklets (Eqs. (10)–(12)).

end if

end for

end for

end if

end for

Link the discrete IDs with the same label and smooth with Kalman Filter to form the final target trajectory.

Output: The final trajectories of all targets.

5.5. Discussion

Robustness to occlusion: Our proposed framework can deal with occlusions naturally. Occlusion occurs either when several targets are overlapping together, or when a target is behind the background. For the first case, the corresponding detection will not get high belief (Eq. (9)) like other un-occluded detections (the occluded detection has high similarity with multiple targets, so that each belief is not high after the normalization). For the second case, there exist missed detections. In both cases, the trajectories before and after the frame with occlusion will be broken down into short tracklets, and hence the calculation of trajectory model will not be affected by occlusion. As the trajectory model becomes more accurate over iterations, the tracklets belonging to the same trajectory will be connected in such a way described in Section 5.3 (shown in Fig. 4).

Computational complexity: Our iterative process stops either when the specified number of iterations are finished, or when there is no change of label assignment. The overall procedure is summarized in Algorithm 1. Note that the LBP algorithm has complexity $O(N^2)$, much lower than [24] which is $O(n^3)$. Hence, the overall algorithm has complexity $O(S \times N^2)$ within a given analysis window, where $S$ is the number of iterations.
6. Experiment results

In this section we report experiment results of the proposed multi-target tracking method on three widely used pedestrian datasets: PETS09 [51], CAVIAR [52] and TUD-Stadtmitte [53].

6.1. Implementation details

Default settings: The three datasets PETS09, TUD-Stadtmitte and CAVIAR have different frame rates, and therefore their motion speed is not the same. The issue can easily be handled by specifying different sliding window length $T_a$ for them experimentally, which are 13, 23 and 22. Threshold $T_b$ is set to 0.9; $T_{len}$ is set to 3; and the parameters $\alpha$ and $\beta$ in Eq. (7) are set to 0.1 and 0.9, respectively, for all three datasets. The final target trajectories are all generated when the optimization converges after 3 iterations.

Influence of parameters: All the above parameters were chosen experimentally and mostly remained identical for all experiments on different sequences. To study the effect of $T_a$, $T_b$ and $T_{len}$ on final track results, we ran our tracking algorithm on the PETS-S2L1 sequence and modified the corresponding parameter while keeping all the other ones fixed. In Fig. 5, for each term, the relative change in performance, as measured by MOTA, is plotted against the parameter value. It can be seen that the parameters $T_a$ and $T_b$ have little influence on the tracking performance for a wide range of settings. For the figure, we can see that the MOTA metric decreases as $T_{len}$ increases. This is because more new targets are deleted and it makes more false negative tracks.

Computational time: We measure the execution time of our system on 20 videos in the evaluation from CAVIAR dataset, in which 2–6 pedestrians are present per frame. The experiments are performed on a recent 3.4 GHz PC with the program being coded in Matlab. With no particular optimization performed, the tracking speed of our system is about 31 fps on average, depending on the number of detections and targets in a sequence.

6.2. Evaluation metrics

To determine whether a track is matched to the true target, our experiment results on all test datasets are evaluated on the widely adopted metric, i.e. the intersection over union of the predicted and the true bounding boxes in 2D image space. The threshold of overlap is set to 0.5 for correct detection, and 0 means no overlap and 1 means that the prediction is correct. Some other state-of-the-art works compute the correspondence directly in world coordinates since all targets are tracked in 3D space. It often sets the hit/miss threshold to 1 m. Specially, we use * to mark the results evaluated on the ground plane in 3D space.

To make fair comparison with respective works, we used the original annotations of CAVIAR, PETS-S2.L1 (see footnote 1), PETS-S2.L2,2 PETS-S2.L3 (see footnote 2) and TUD-Stadtmitte (see footnote 1), which guaranteed the off-line learned human detector responses to be the same as used in the compared approaches. On the other hand, since there is no single established metric, we conduct experiment evaluations and performance comparisons using one of the most comprehensive metric sets introduced in [27], and MOTA, MOTP from CLEAR-metrics [54] as well. Thus the final evaluation metrics are listed in Table 1. $\uparrow$ means higher is better, as opposed to ↓. MOTA (Multi-Object Tracking Accuracy) synthesizes all false positives, false negatives and identity switches into a single number. MOTP (Multi-Object Tracking Precision) averages the bounding box overlap between the ground truth and the tracker output as a measure of localization accuracy. Furthermore, since there exists a conflict between the rate of recall and precision, we introduced the $F_1$-measure index to measure these two metrics comprehensively:

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

which is the harmonic mean of precision and recall.

6.3. Evaluation results

The PETS09 datasets consider crowd image analysis and include crowd count and density estimation, tracking of individual(s) within a crowd, and detection of separate flows and specific crowd events. The datasets we used in this paper contain different scenarios containing three levels of difficulty with low (S2.L1), medium (S2.L2) and high person densities (S2.L3). For the three sequences the datasets provide multiple views from different camera angles, however, we only use the first view of each sequence in all our experiments.

The sequence S2.L1 is the most widely used in multi-target tracking literature. People are tracking in a sparse crowd. The sequence is 795 frames long, and shows up to 8 people, whose motion directions change frequently. Table 2 shows the quantitative evaluation results for the dataset. Many state-of-the-art methods in the literature [42,44,6,55] also used this dataset in their experiment evaluations, and the evaluation results reported in their papers were also included in the table for fair comparison. The table shows that our approach achieves fairly good recall and precision, and the comprehensive $F_1$-measure is the best among all methods. Furthermore, our method generates no ID switches,

---


---

Please cite this article as: S. Zhang, et al., Multi-target tracking by learning local-to-global trajectory models, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.08.013
few fragments, and the best MOTA and MOTP compared with other methods. Some visual results are shown in the first row of Fig. 6. The tails behind the people illustrate that our approach is able to keep correct identities and trajectories when targets are occluded, such as person 19. Targets with similar appearance can be correctly distinguished, such as person 19 and 22. Besides the widely used PETS-S2.L1 sequence, we extend our test data with two more challenging scenarios with high crowd density (S2.L2 and S2.L3) from the PETS09 dataset. Both the sequences have crowded pedestrians and more inter-object occlusions, making tracking significantly more challenging. The sequence S2.L2 contains 436 frames and shows 33 targets at most in one frame, while S2.L3 contains 240 frames and shows 42 targets at most. Table 2 shows the quantitative results on both sequences. Notice that our experiments are evaluated only in 2D image space, while the listed state-of-the-art tracking method based on the continuous energy minimization algorithm [56] was performed in the world coordinate of 3D space. From the table we can see that our method achieves good tracking performance even with 2D image information only, such as lower fragments, lower FAF and higher precision. In these denser crowds, there are many missed detections due to severe overlaps and occlusions so that some occluded targets are not rightly detected. Thus the trajectory models deleted those detections that only appear in a few frames, which leads to the relatively low recall and MOTA. Since we only used 2D information for each sequence, achieving a comparable performance with those methods based on 3D information can demonstrate the robustness and effectiveness of the proposed method. Some visual results of S2.L2 are shown in the second row of Fig. 6.

Table 2 presents the evaluation results of the proposed method and three other state-of-the-art methods [41,55,58] that used the TUD-Stadtmitte dataset in performance evaluations. The TUD-Stadtmitte dataset shows walking pedestrians in a city environment and contains only 179 frames, but it is one of the most challenging benchmark datasets in the literature because it is recorded by a camera with a low viewing angle, which results in frequent occlusions among targets and a large variation in their sizes. In the third row of Fig. 6, person 1, 4 and 5 (pointed by the arrows) are heavily occluded through the process. Nonetheless, our proposed method is still able to locate the targets and track them reliably without fragments. We achieved the best MT, PT (both are 10% higher than the state-of-the-art method), MOTA and MOTP among the results listed in the table. The F1-measure is almost the same as the state-of-the-art method proposed by [58], which requires more complex computing of the MAP solution of the proposed CRF model. Notice that even though we do not explicitly handle occlusions, our algorithm is able to connect the tracklets across occlusion gaps in most cases.

In the CAVIAR dataset a number of video clips were recorded acting out the different scenarios of interest. These include people walking alone, meeting with others, window shopping, entering and exiting shops, fighting and passing out and leaving a package in a public place. In accordance with the previous results reported in [55,27], we select 20 sequences of the dataset for comprehensive testing. The sequences include 143 ground truth trajectories,
including 77,296 pedestrian labels in 29,283 frames (2.6 labels per frame, 204 labels per track on average). Table 4 shows the result of performance evaluations on this dataset. We can see from the table that our method achieves the best Precision, F1 and FAF measures among all the methods, and reduces the ID switches to 6. The last row of Fig. 6 shows some visual tracking results in our method. We notice that most fragments were generated when people went into or went off the scene only with the upper body, thus our method made errors to generate fragments, such as person 15 in the second picture. Generally speaking, the proposed iterative algorithm enables us to pinpoint the targets, and to handle complex situations appropriately. More visual demonstrations of our experimental results can be seen from our project website.3

6.4. Effectiveness of iterations

Table 5 shows the intermediate results of our proposed algorithm on dataset PETS-S2.L1. At the beginning of the tracking task, the trajectory models \( \Gamma \) were initialized with the detection result from the first frame of the video stream, which were not too

### Table 3

<table>
<thead>
<tr>
<th>Method</th>
<th>Recall (%)</th>
<th>Precision (%)</th>
<th>F1 (%)</th>
<th>FAF</th>
<th>GT (%)</th>
<th>MT (%)</th>
<th>PT (%)</th>
<th>ML (%)</th>
<th>Frac</th>
<th>IDS</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andriyenko et al. [41]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>9</td>
<td>67</td>
<td>33</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>60.5</td>
<td>65.8</td>
</tr>
<tr>
<td>Kuo et al. [55]</td>
<td>81.0</td>
<td>99.5</td>
<td>89.30</td>
<td>0.028</td>
<td>10</td>
<td>60</td>
<td>30</td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Yang et al. [58]</td>
<td>87.0</td>
<td>96.7</td>
<td>91.59</td>
<td>0.184</td>
<td>10</td>
<td>70</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ours</td>
<td>85.8</td>
<td>98.1</td>
<td>91.54</td>
<td>0.100</td>
<td>10</td>
<td>80</td>
<td>80</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>84.2</td>
<td>86.5</td>
</tr>
</tbody>
</table>

Results of the method in [55] are provided by Ref. [58].

Please cite this article as: S. Zhang et al., Multi-target tracking by learning local-to-global trajectory models, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.08.013
assigned wrong target IDs. This is re
the rest ones were either treated as false positive detections or
assigned appropriate labels and correctly linked together, while
Multi-target tracking results after each iteration on the PETS-S2.L1 dataset.

Table 4
Quantitative comparison with other state-of-the-art methods on the CAVIAR dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Recall (%)</th>
<th>Precision (%)</th>
<th>F1 (%)</th>
<th>FAF</th>
<th>GT</th>
<th>MT (%)</th>
<th>PT (%)</th>
<th>ML (%)</th>
<th>Frag</th>
<th>IDS</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kuo et al. [55]</td>
<td>88.1</td>
<td>96.6</td>
<td>92.2</td>
<td>0.082</td>
<td>143</td>
<td>86.0</td>
<td>13.3</td>
<td>0.7</td>
<td>17</td>
<td>4</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Li et al. [27]</td>
<td>89.0</td>
<td>94.1</td>
<td>91.5</td>
<td>0.157</td>
<td>143</td>
<td>84.6</td>
<td>14.0</td>
<td>1.4</td>
<td>17</td>
<td>11</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Huang et al. [25]</td>
<td>86.3</td>
<td>–</td>
<td>–</td>
<td>0.186</td>
<td>143</td>
<td>78.3</td>
<td>14.7</td>
<td>7</td>
<td>54</td>
<td>12</td>
<td>80.0</td>
<td>–</td>
</tr>
<tr>
<td>Ours</td>
<td>89.2</td>
<td>97.8</td>
<td>93.3</td>
<td>0.048</td>
<td>143</td>
<td>85.3</td>
<td>13.2</td>
<td>1.5</td>
<td>32</td>
<td>6</td>
<td>87.2</td>
<td>76.3</td>
</tr>
</tbody>
</table>

Results of the method in [25] are provided by Ref. [27].

Table 5
Multi-target tracking results after each iteration on the PETS-S2.L1 dataset.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Recall (%)</th>
<th>Precision (%)</th>
<th>F1 (%)</th>
<th>FAF</th>
<th>GT</th>
<th>MT (%)</th>
<th>PT (%)</th>
<th>ML (%)</th>
<th>Frag</th>
<th>IDS</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iter1</td>
<td>52.1</td>
<td>94.7</td>
<td>67.22</td>
<td>0.17</td>
<td>19</td>
<td>5</td>
<td>79</td>
<td>16</td>
<td>107</td>
<td>36</td>
<td>48.4</td>
<td>88.9</td>
</tr>
<tr>
<td>Iter2</td>
<td>94.3</td>
<td>98.5</td>
<td>96.35</td>
<td>0.09</td>
<td>19</td>
<td>95</td>
<td>5</td>
<td>0</td>
<td>17</td>
<td>8</td>
<td>92.7</td>
<td>88.2</td>
</tr>
<tr>
<td>Iter3</td>
<td>97.0</td>
<td>98.6</td>
<td>97.79</td>
<td>0.08</td>
<td>19</td>
<td>95</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>95.6</td>
<td>91.6</td>
</tr>
</tbody>
</table>

accurate. The detections that fitted the estimations well by \( \Gamma \) were
assigned appropriate labels and correctly linked together, while
the rest ones were either treated as false positive detections or
assigned wrong target IDs. This is reflected by the poor metrics in
the first row of Table 5, such as Recall, FAF, MT and FM. For the
subsequent iteration, \( \Gamma \) was updated by the confident tracklets
generated by the previous iteration, which resulted in a better \( \Gamma \).
A better \( \Gamma \) defined a better MRF model which led to more accurate
tracklets. From Table 5, we find that most evaluation metrics in the
second iteration are obviously superior to those of the first
iteration. Similarly, all the metrics continue to improve in the
third iteration. These intermediate results demonstrate the effec-
tiveness of the proposed inference algorithm.

7. Conclusion

We propose a unified framework of the MRF model to simul-
taneously track multiple targets by learning the local-to-global
trajectory models. The MRF model formulates the tracking pro-
blem as the target identity association problem and generates the
tracklets by the LBP algorithm. We select the reliable tracklets to
update the trajectory model parameters, and then iteratively
update the MRF model to generate longer tracklets. The trajectory
models are further updated by the longer tracklets, as a result, the
output results are further refined. Our experiments on the stan-
dard PETS-S2.L1, CAVIAR dataset and the challenging TUD-
Stadmitte datasets show that our framework achieves signi-
cant improvements in tracking recall and precision over those in
the previous state-of-the-art works [41,42,55,6,44].

Conflict of interest

None declared.

Acknowledgement

This work is supported by National Natural Science Foundation
of China under Grant no. 61102100, and the key program from
National Natural Science Foundation of China under Grant no.
61332018.

References


Please cite this article as: S. Zhang, et al., Multi-target tracking by learning local-to-global trajectory models, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.08.013
Shun Zhang received the B.S. degree in Electronic Engineering from Xi’an Jiaotong University, Xi’an, China, in 2009. He is currently pursuing the Ph.D. degree in Institute of Artificial Intelligence and Robotics at Xi’an Jiaotong University. His research interests include Machine Learning, Computer Vision and Human-Computer Interaction, with a focus on visual tracking, object detection, image classification, feature extraction and sparse representation.

Jinjun Wang received the B.E. and M.E. degrees from Huazhong University of Science and Technology, China, in 2000 and 2003. He received the Ph.D. degree from Nanyang Technological University, Singapore, in 2006. From 2006 to 2009, he was with NEC Laboratories America, Inc. as a postdoctoral research scientist, and in 2010, he joined Epson Research and Development, Inc. as a senior research scientist. His research interests include pattern classification, image/video enhancement and editing, content-based image/video annotation and retrieval, semantic event detection, etc.

Zelun Wang is a fourth-year undergraduate student, enrolled in the School of Electronics & Information Engineering, Xi’an Jiaotong University. He is now doing his bachelor thesis in the Institute of Artificial Intelligence and Robotics. His research interests include artificial intelligence, computer vision and human-computer interaction.

Yihong Gong received the B.S., M.S., and Ph.D. degrees in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1987, 1989, and 1992, respectively. In 1992, he joined Nanyang Technological University, Singapore, where he was an Assistant Professor with the School of Electrical and Electronic Engineering for four years. From 1996 to 1998, he was a Project Scientist with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. He was a Principal Investigator for both the Informedia Digital Video Library Project and the Experience-On-Demand Project funded by the National Science Foundation, the Defense Advanced Research Projects Agency, the National Aeronautics and Space Administration, and other government agencies. From 1999 to 2009, he was the Head of the Department of Information Analysis and Management, NEC Laboratories America, Inc., Cupertino, CA. He is currently a professor with Xi’an Jiaotong University (XJTU), China. His research interests include image and video analysis, multimedia database systems, and machine learning.

Yuehu Liu received his B.E. and a M.E. degrees in Computer Science and Engineering at Xi’an Jiaotong University, China, in 1984 and 1989, respectively, and the Ph.D. degree in electrical engineering from Keio University, Japan, in 2000. He is currently a professor and the vice director of the Institute of Artificial Intelligence and Robotics at Xi’an Jiaotong University. His research interests include computer vision, pattern recognition, computational intelligence, image processing.