Multiple Object Tracking Using Local Motion Patterns

Mehrsan Javan Roshtkhari
http://www.cim.mcgill.ca/~javan
Martin D. Levine
http://www.cim.mcgill.ca/~levine

Center For Intelligent Machines
Department of Electrical and Computer Engineering
McGill University
Montreal, QC, Canada

Abstract

This paper presents an algorithm for multiple-object tracking without using object detection. We concentrate on creating long-term trajectories for unknown moving objects by using a model-free tracking algorithm. Each individual object is tracked by modeling the temporal relationship between sequentially occurring local motion patterns. The algorithm is based on shape and motion descriptors of moving objects, obtained at two hierarchical levels from an event understanding system. By considering both local and global motion patterns, two sets of initial tracks, called linklets, are obtained. Then, a set of sparse tracks, referred to in the literature as tracklets, is produced by grouping linklets demonstrating similar motion patterns. This produces two sets of independent tracklets, referred to as the low- and high-level tracklets. We adopt Markov Chain Monte Carlo Data Association (MCMCDA) to estimate a varying number of trajectories given a set of tracklets as input. To this end, we formulate tracklet association as a Maximum A Posteriori (MAP) problem to create a chain of tracklets. The final output of the data association algorithm is a partition of the set of tracklets such that those belong to individual objects have been grouped. This yields individual tracks for each object in a video.

1 Introduction

Object tracking is, perhaps, the most fundamental task for any high-level video content analysis system. Decades of research on this topic have produced a diverse set of approaches and a rich collection of tracking algorithms. Readers can refer to [33] and [32] for a review of the state-of-the-art in object tracking and a detailed analysis and comparison of various representative methods. In the majority of the traditional approaches, only the object itself and/or its background are modeled. However, we observe that significant progress has been made in this case. This class of tracking methods is referred to as "object-centric" approaches [18]. On the other hand, detection cannot be performed when there is no prior knowledge about the specific objects being tracked. These methods are referred to as "generic object tracking" or "model-free tracking". Clearly, manually annotating sufficient numbers of object is often prohibitively expensive and impractical. Thus recently approaches for model-free tracking have received increased interest [18, 20]. Model-free tracking is a challenging task because there is little information available about the object to be tracked [20]. Another challenge is the presence of an unknown and ever-changing number of targets.

In this paper we concentrate on creating long-term trajectories for unknown moving objects by using a model-free tracking algorithm. As opposed to the tracking-by-detection algorithms [16, 31], no object detection is involved. Each individual object is tracked only by modeling the temporal relationship between sequentially occurring local motion patterns. This is achieved by constructing two sets of initial tracks that code local and global motion patterns in videos. These local motion patterns are obtained by analyzing spatially and temporally varying structures in videos. Initially, the video is densely sampled, spatio-temporal video volumes (STVs) are constructed, and similar ones are grouped to reduce the dimension of the search space. This is called the low-level codebook. Then, a large contextual region containing many STVs (in space and time) around each pixel is examined and their compositional relationships are approximated using a probabilistic framework. They are then employed to form yet another codebook, called the high-level codebook. Therefore, two codewords are assigned to each pixel, one from the low level and the other from the high level codebook. By examining pairs of sequential video frames, the matching codewords for each video pixel are transitively linked into distinct tracks, whose total number is unknown a priori and which we will refer to as linklets. The linking process is separately performed for both codebooks. This is done under the hard constraint that no two linklets may share the same pixel at the same time, i.e. the assigned codewords. The end result at this step is two sets of independent linklets obtained from the low- and high-level codebooks.

Subsequently, a set of sparse tracks, referred to as tracklets in the literature, are produced by grouping the linklets that indicate similar motion patterns (see Figure 1). This produces two sets of independent tracklets, referred to as low- and high-level tracklets. We adopt Markov Chain Monte Carlo Data Association (MCMCDA) to estimate an initially unspecified number of trajectories. To this end, we formulate the tracklet association problem as a Maximum A Posteriori (MAP) problem to produce a chain of tracklets. The final output of the data association algorithm is a partition of the set of tracklets such that those belonging to each individual object have been grouped together.

The main contribution of this paper is an approach capable of learning long-term trajectories of any moving object in a video without using any prior knowledge about the objects (object detection). This is achieved by creating local trajectories of regions that have similar motion patterns, while also considering their neighboring regions (contextual information). Therefore, this algorithm is a complete *bottom up* tracking method that only employs hierarchical codebooks to characterize local motion patterns as the *observations*. These hierarchical codebooks are obtained as described by the authors in [25]. In addition, by considering tracklets at two hierarchical levels, the data association algorithm is capable of easily handling missing information. Data association is accomplished by considering temporal continuity and motion consistency of both the low- and high-level tracklets, with the additional option of rejecting irrelevant tracklets.

2 Related Work

To date, most of the reported approaches for tracking rely on either robust motion or appearance models of each individual object or on object detection, i.e., they are object-centric. Thus a key assumption is that a reliable object detection algorithm exists [10, 16, 20, 23, 31]. This remains a challenge, particularly in complex and crowded situations. These methods use the detection response to construct an object trajectory. This is accomplished by using data association based on either the detection responses or a set of short tracks called tracklets

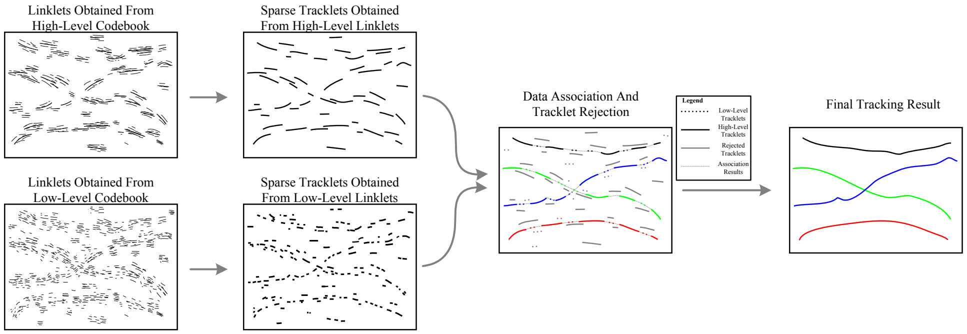

Linklets Obtained From High-Level Codebook
Sparse Tracklets Obtained From High-Level Linklets
Linklets Obtained From Low-Level Codebook
Sparse Tracklets Obtained From Low-Level Linklets
Data Association And Tracklet Rejection
Legend
Low-Level Tracklets
High-Level Tracklets
Rejected Tracklets
Association Results
Final Tracking Result

Figure 1: Overview of the algorithm. The goal is to estimate the trajectory of the moving objects in the video without invoking object detection. Initially two sets of linklets are constructed by chaining; the low-level considers small window fragments, while the high-level analyzes a larger region in order to impose a contextual influence. They are obtained by exploiting an activity understanding system. The resultant tracks (chains) are filtered and replaced by a set of sparse representative tracks, the so-called tracklets. Longer trajectories are then generated by using the Markov Chain Monte Carlo Data Association (MCMCDA) algorithm to solve the Maximum A Posteriori (MAP) problem using tracklet affinities. Thus this procedure uses low-level tracklets to connect high-level tracklets when there is a discontinuity in motion or time.

that are associated with each detected object [10, 20, 29]. Tracklets are mid-level features that provide more spatial and temporal context than raw sensor data during the process of creating consistent object trajectories. Subsequently, data association links these tracklets into multi-frame trajectories. The issue of associating tracklets across time, the so-called data association, is usually formulated as a MAP problem and has been solved using different methods. For example, network flow graphs and cost-flow networks are employed for data association in [9, 16, 19] to determine globally optimal solutions for an entire sequence of tracklets. Other data association approaches include the Hungarian algorithm [21], maximum weight independent sets [6], the Markov Chain Monte Carlo [4, 22, 29], and the iterative hierarchical tracklet linking methods [10].

On the other hand, there are other tracking algorithms, which are based on local spatio-temporal motion patterns in the scene. More closely related to our approach are those that construct motion models for the moving objects without performing any detection [7, 15, 18, 29, 30]. For example, in [17, 18], hidden Markov models are employed to learn local motion patterns that are subsequently used as prior statistics for a particle filter. Alternatively, other methods, such as those in [15] and [2], employ the global motion patterns of a crowd to learn local motion patterns of the neighboring local regions. Individual moving entities are detected by associating similar trajectories based on their features in [7] and [30]. These authors assume that objects move in distinct directions, and thus disregard possible and very likely local motion inconsistencies between different body parts. Thus a single pedestrian could be detected as a multiple target or multiple individuals as the same target. In order to overcome these difficulties, we analyze trajectories at two hierarchical levels, in which the second level accounts for the inconsistency between local motions of a single object.

Our proposed algorithm provides an alternative by strictly using only local motion patterns and contextual information within a data association framework. In contrast to the

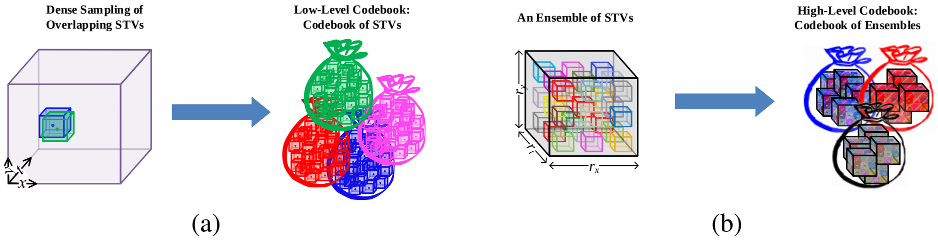

Dense Sampling of Overlapping STVs
Low-Level Codebook: Codebook of STVs
An Ensemble of STVs
High-Level Codebook: Codebook of Ensembles
y
t
x
r_y
r_t
r_x
(a)
(b)

Figure 2: Observations are represented by low- and high-level codebooks. First, the video is densely sampled scales to produce a set of overlapping STVs and subsequently, a two-level hierarchical codebook is created. (a) At the lower level of the hierarchy, similar video volumes are dynamically grouped to form a conventional fixed-size low-level codebook, $\mathbf{C}^{\mathcal{L}}$. (b) At the higher level, a much larger spatio-temporal 3D volume is created. It contains many STVs at and captures the spatio-temporal arrangement of the volumes, called an ensemble of volumes. Similar ensembles are grouped based on the similarity between arrangements of their video volumes and yet another codebook is formed, $\mathbf{C}^{\mathcal{H}}$ [25, 26].

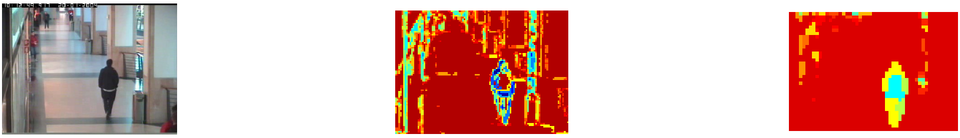
aforementioned approaches that attempt to track objects either by detection or learning an appearance model of the objects, our goal is to construct a hierarchical model for all moving objects in a scene.

3 Hierarchical Data Association And Tracking

3.1 Observations: Low- And High Level Codebooks Of Local Motions

Consider the overview in Figure 1 and assume that a system capable of producing the linklets (on the left) is available for event description. Our aim is to use the information produced by such a system to detect and track all moving objects in the scene. Here we adopt the hierarchical bag of video words framework developed in [25, 26] for short-term event description. In general, this on-line framework produces two sets of codebooks in real-time and assigns labels to local spatio-temporal video volumes (STVs) based on their similarity, while also considering their spatio-temporal relationships. The hierarchical algorithm dynamically codes a video as both a compact set of individual and ensembles of spatio-temporal volumes. These latter are used to construct a probabilistic model of video volumes and their spatio-temporal compositions (see Figure 2).

The first step is to represent a video by a meaningful low-level codebook. Using the framework developed in [25], we determine STVs using dense sampling and then cluster them at each frame based on similarity. We refer to the constructed low-level codebook at this level as $\mathbf{C}^{\mathcal{L}}$, as illustrated in Figure 2. The 3D STVs, $v_i \in \mathbb{R}^{n_x \times n_y \times n_t}$ are constructed by assuming a volume of size $n_x \times n_y \times n_t$ around each pixel. Each STV volume is then characterized by a descriptor vector, taken as a histogram of oriented gradients (HOG3D) within the STV. The HOG is constructed using the quantized spatial and temporal gradients converted to polar coordinates and weighted by the gradient magnitude [8, 25, 27]. The codebook is then created using online fuzzy clustering, which is capable of incrementally updating the cluster centers as new data are observed [14]. The clusters are used to produce a codebook of STVs and ultimately assign a label to each STV. Once a video clip has been processed by the first level of clustering as described in the previous section, we examine a



(a) (b) (c)

Figure 3: Codeword assignment for each pixel. (a) A sample video frame from the CAVIAR dataset [1]; (b) Color-coded low-level codewords assigned to every pixel in the video frame. In this case, there is a large number of low-level codewords; (c) High-level codewords, which represent compositions, are also assigned to every pixel in the video frame. This would generally produce a small number of codewords since it deals with objects in the scene. Each object might be represented by a large number of low-level codewords, while the high-level codebook assigns a few number of codewords to an objects, in most cases one or two.

large region, R, around each pixel. R contains many video volumes and thereby captures both local and more distant information in the video frames. Such a set is called an *ensemble* of volumes around the particular pixel (Figure 2). The *relative* spatio-temporal coordinates of the volume in each ensemble capture the spatio-temporal compositions of the video volumes, [24]. Each ensemble of STVs is represented by a *probability density function* of its spatio-temporal volume distribution, as described in [25]. This histogram becomes the descriptor for each ensemble and forms the second level codebook, called the high-level codebook of ensembles of volumes, $\mathbf{C}^{\mathcal{H}}$, as described in [25]. A sample video frame and the assigned codewords are illustrated in Figure 3.

3.2 Linklets And Tracklets

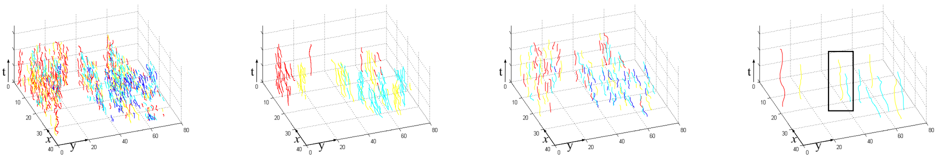
As indicated earlier, tracklets are obtained from both the low- and high-level codebooks, $\mathbf{C}^{\mathcal{L}}$ and $\mathbf{C}^{\mathcal{H}}$, constructed in section 3.1. Two codewords are assigned to each pixel $p(x,y)$ at time (t) in the video. Therefore, in a video sequence of temporal length T, a particular pixel $p(x,y)$ is represented by two sequences of assigned codewords[1]:

$$\begin{aligned} p\left(x,y\right) &= \left\{p(x,y) \leftarrow c_i : \forall t \in T,\, c_i \in \mathbf{C}^{\mathcal{L}}\right\} \\ p\left(x,y\right) &= \left\{p(x,y) \leftarrow c_i : \forall t \in T,\, c_i \in \mathbf{C}^{\mathcal{H}}\right\} \end{aligned} \tag{1}$$

Given the assigned codewords (labels) for each pixel, we obtain an over-segmented representation of the video (see Figure 3). In this over-segmented representation, each segment represents a set of pixels that are similar in terms of local motion patterns. Therefore, it is a simple task to create a short trajectory for each pixel by examining the temporal coherence of its assigned codewords. This is comparable to the concept of so-called "particles" [28]. Here we conservatively associate two responses only if they are in consecutive frames and are close enough in space and similar enough according to their assigned codewords. Thus we obtain, two sets of trajectories, called $\mathcal{X}^{\mathcal{L}}$ and $\mathcal{X}^{\mathcal{H}}$ (see Figure 4).

It is obvious that the number of linklets is generally more than the number of objects in the scene and that many trajectories might belong to a single object. In addition, we note that the number of linklets created by a single object is much smaller in $\mathcal{X}^{\mathcal{H}}$ than the ones in $\mathcal{X}^{\mathcal{L}}$. Ideally we are interested in obtaining a single trajectory for an object. Thus the linklets

[1] $\leftarrow$ symbolizes value assignment.



(a) (b) (c) (d)

Figure 4: Linklet and tracklet construction. (a) A set of linklets (short tracks) constructed using observations obtained from the low-level codebook, $\mathcal{X}^{\mathcal{L}}$. (b) A set of linklets constructed using observations obtained from the high-level codebook, $\mathcal{X}^{\mathcal{H}}$. (c) Low-level tracklets, $\mathbf{T}^{\mathcal{L}}$, obtained by grouping similar linklets in $\mathcal{X}^{\mathcal{L}}$. (d) High-level tracklets, $\mathbf{T}^{\mathcal{H}}$, obtained by grouping similar linklets in $\mathcal{X}^{\mathcal{H}}$. The black rectangle indicates the area in XYT-space occupied by a single person. It seems that a single person may produce more than a single trajectory. We expect this because our algorithm does not involve any person or object detection. We deal with this issue in the next section, which describes a data association process that rejects certain tracklets as false positives.

belonging to the same object must be merged in order to create a single representative track that describes the motion of the object. Here we follow the idea of clustering trajectories to create a representative object trajectory [8, 28].

Obviously non-informative linklets are removed before constructing clusters of trajectories. These are taken to be relatively motionless or those that carry little information about the motion. They are mainly related to the background or static objects. Similar to [28], we analyze the linklets within a temporal window of the length of T. Then, those trajectories with a small variance are removed[2]:

$$
\begin{aligned}
\mathbf{X}^{\mathcal{L}} &= \left\{\mathbf{x} \in \mathcal{X}^{\mathcal{L}},\ var\{\mathbf{x}\} \geq \varepsilon^{\mathcal{L}}\right\} \\
\mathbf{X}^{\mathcal{H}} &= \left\{\mathbf{x} \in \mathcal{X}^{\mathcal{H}},\ var\{\mathbf{x}\} \geq \varepsilon^{\mathcal{H}}\right\}
\end{aligned}
\tag{2}
$$

where $\varepsilon^{\mathcal{L}}$ and $\varepsilon^{\mathcal{L}}$ are two thresholds. Clearly, trajectories are not of the same temporal length. Therefore, in order to measure dissimilarity between two trajectories, we adopt the pairwise affinities between all trajectories as introduced in [8]. The distance between two trajectories $\mathbf{x}$ and $\mathbf{y}$, $D(\mathbf{x},\mathbf{y})$, is defined as: $D^2(\mathbf{x},\mathbf{y}) = \max_t\left\{d_t^2(\mathbf{x},\mathbf{y})\right\}$. $d_t^2(\mathbf{x},\mathbf{y})$ is the distance between two trajectories $\mathbf{x}$ and $\mathbf{y}$ at the time t and defined as follows:

$$
d^2(\mathbf{x},\mathbf{y}) = \|(\mathbf{x}-\mathbf{y})\|^2 \frac{\|\nabla_t(\mathbf{x}-\mathbf{y})\|^2}{5\sigma_t^2} \tag{3}
$$

The first factor on the right-hand-side of (3) is the average spatial Euclidean distance between the two trajectories. The second factor characterizes the motion of a point aggregated over 5 frames at time t. The normalization term, σ_t, accounts for the local variation in the motion [8]. Given the above distance measurement between two trajectories, clustering is performed using the k-means algorithm. Here we have invoked iterative clustering to determine the optimal number of clusters. In order to perform the merging, we use the Jensen-Shannon divergence measure to compute the actual difference between the resulting clusters. As reported in [28], this method achieves better results than others for trajectory clustering. Clustering produces two sets of low-level tracklets, which we refer to as $\mathbf{T}^{\mathcal{L}}$ and $\mathbf{T}^{\mathcal{H}}$.

[2] Other methods can be used to remove uninformative codewords, such as the one presented in [27].

As illustrated in Figure 4, the tracklets obtained after clustering are not quite reliable for long term object tracking, but do a relatively good job of encoding the moving object motions in the short term. The main advantage of constructing the tracklets based on the two codebooks is that no object detection is required. Although a set of representative trajectories is created for all moving objects in the video, there is no guarantee that an object would be represented by a single trajectory. Moreover, in crowded scenes, the representative trajectories may correspond to more than one object. However, if the motion pattern changes, then the trajectories would separate.

3.3 Data Association and High-Level Trajectory Construction

Given the resulting tracklets, high-level trajectories can be generated by linking them in space and time. We achieve this by formulating the data association required as a maximum a posteriori (MAP) problem and solve it with the Markov Chain Monte Carlo Data Association (MCMCDA) algorithm. The observations are taken to be the constructed tracklets in section 3.2:

$$\mathcal{O} = \left\{ \mathbf{T}^{\mathcal{L}}, \mathbf{T}^{\mathcal{H}} \right\} \tag{4}$$

Let $\mathbf{\Gamma}$ be a tracklet association result, which is a set of trajectories, $\Gamma_k \in \mathbf{\Gamma}$. Γ_k is defined as a set of the connected observations which is a subset of all observations, $\Gamma_k = \left\{ T_i^{\mathcal{L}}, T_j^{\mathcal{H}} \right\} \subseteq \mathcal{O}$. The goal is to find the most probable set of object trajectories, $\mathbf{\Gamma}$, which is formulated as a MAP problem:

$$\mathbf{\Gamma}^* = \arg\max_{\mathbf{\Gamma}} P\left(\mathbf{\Gamma}|\mathcal{O}\right) = \arg\max_{\mathbf{\Gamma}} P\left(\mathcal{O}|\mathbf{\Gamma}\right) P\left(\mathbf{\Gamma}\right) \tag{5}$$

The likelihood, $P\left(\mathcal{O}|\mathbf{\Gamma}\right)$ indicates how well a set of trajectories matches the observations and the prior, $P\left(\mathbf{\Gamma}\right)$ indicates how correct the data association is. By assuming that the likelihoods of the tracklets are conditionally independent, we can rewrite the likelihood, $P\left(\mathcal{O}|\mathbf{\Gamma}\right)$, in (5) as follows:

$$P\left(\mathcal{O}|\mathbf{\Gamma}\right) = \prod_{\substack{T_i^{\mathcal{L}} \in \mathbf{T}^{\mathcal{L}} \\ T_j^{\mathcal{H}} \in \mathbf{T}^{\mathcal{H}}}} P\left(T_i^{\mathcal{L}}, T_j^{\mathcal{H}}|\mathbf{\Gamma}\right) \prod_{\Gamma_k \in \mathbf{\Gamma}} P\left(\Gamma_k\right) \tag{6}$$

First we consider the encoding of the likelihood of tracklets in (6). The observations, that is, the tracklets, can be either true or false trajectories of the object. Therefore, the likelihood of a tracklet, given the set of trajectories, $\mathcal{S}$, can be modeled by a Bernoulli distribution:

$$P\left(T|\mathbf{\Gamma}\right) \sim Bern(p) = \begin{cases} p^{|T|} & : T \in \Gamma_k,\ \Gamma_k \in \mathbf{\Gamma} \\ (1-p)^{|T|} & : T \notin \Gamma_k,\ \Gamma_k \in \mathbf{\Gamma} \end{cases} \tag{7}$$

where $|T|$ denotes how good a tracklet is. Since the tracklets are taken to be clusters of small trajectories constructed in section 3.2, $|T|$ is defined as the size of the cluster. Here we assume that the two sets of tracklets, $\mathbf{T}^{\mathcal{L}}$ and $\mathbf{T}^{\mathcal{H}}$, are independent[3]. Therefore, we can write the likelihood in (6) as follows:

$$P\left(T_i^{\mathcal{L}}, T_j^{\mathcal{H}}|\mathbf{\Gamma}\right) = P\left(T_i^{\mathcal{L}}|\mathbf{\Gamma}\right) P\left(T_j^{\mathcal{H}}|\mathbf{\Gamma}\right) \tag{8}$$

where $P\left(T_i^{\mathcal{L}}|\mathbf{\Gamma}\right) \sim Bern(p^{\mathcal{L}})$ and $P\left(T_j^{\mathcal{H}}|\mathbf{\Gamma}\right) \sim Bern(p^{\mathcal{H}})$ as described in (7). This formulation makes it possible to exclude some tracklets from the final data association by assuming

[3] The independency assumption is valid here because the consistency between tracklets and observations, i.e., the suitability of the tracklets, is independent of the relationship between trajectories.

that any tracklet can belong to at most one trajectory in the data association process. This is achieved simply by rejecting them as false object tracklets.

Next we consider the encoding of the prior of tracklets in (6), $P(\Gamma_k)$. Similar to [10], we model these priors as a Markov chain:

$$P(\Gamma_k) = \prod_{\Gamma_k^t \in \Gamma_k} P\left(\Gamma_k^t | \Gamma_k^{t-1}\right) = P_i\left(\Gamma_k^0\right) P_l\left(\Gamma_k^1 | \Gamma_k^0\right) \dots P_l\left(\Gamma_k^n | \Gamma_k^{n-1}\right) P_t\left(\Gamma_k^n\right) \tag{9}$$

where Γ_k^t is the trajectory of the object at a time instant t. The chain consists of an initialization term, P_i, a probability to link the tracklets, P_l, and a termination probability, P_t, to terminate the trajectory. It is assumed that a trajectory can only be initialized or terminated using the tracklets obtained from the high-level codebook, $\mathbf{T}^{\mathcal{H}}$. Therefore, the probabilities of initializing and terminating a trajectory are written as follows:

$$P_i\left(\Gamma_k^0\right) = P_i\left(T_j^{\mathcal{H}}\right) \qquad , \qquad P_t\left(\Gamma_k^n\right) = P_i\left(T_j^{\mathcal{H}}\right) \tag{10}$$

The probability of linking two tracklets can be written as:

$$P_l\left(\Gamma_k^t | \Gamma_k^{t-1}\right) = P_l\left(T_{j_t}^{\mathcal{H}}, T_{i_t}^{\mathcal{L}} | T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) = P_l\left(T_{j_t}^{\mathcal{H}} | T_{i_t}^{\mathcal{L}}, T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) P_l\left(T_{i_t}^{\mathcal{L}} | T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) \tag{11}$$

Two tracklets are linked if they are consistent in the time domain and show similar motion patterns. We assume independency and decompose the probability of linking the tracklets into two probabilities. Therefore (11) is rewritten as:

$$\begin{aligned} P_l\left(\Gamma_k^t | \Gamma_k^{t-1}\right) =& P_{\mathcal{T}}\left(T_{j_t}^{\mathcal{H}} | T_{i_t}^{\mathcal{L}}, T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) P_{\mathcal{M}}\left(T_{j_t}^{\mathcal{H}} | T_{i_t}^{\mathcal{L}}, T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) \\ & P_{\mathcal{T}}\left(T_{i_t}^{\mathcal{L}} | T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) P_{\mathcal{M}}\left(T_{i_t}^{\mathcal{L}} | T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) \end{aligned} \tag{12}$$

where the temporal consistency probability, $P_{\mathcal{T}}$, is taken to be the hyper-exponential distribution of the temporal gap between the tracklets:

$$\begin{aligned} P_{\mathcal{T}}\left(T_{j_t}^{\mathcal{H}} | T_{i_t}^{\mathcal{L}}, T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) &= \sum_n \alpha_n P_n(\tau_n) \\ P_n(\tau_n) \sim Exp(\lambda_n) &= \begin{cases} \lambda_n e^{(\lambda_n \tau_n)} & : \tau_n \geq 0 \\ 0 & : \tau_n < 0 \end{cases} \end{aligned} \tag{13}$$

where τ_n is the temporal distance between the end of a tracklet and the start of its immediate successor. The motion consistency probability, $P_{\mathcal{M}}$, is modeled by assuming that the trajectories follow a constant velocity model and obey a Gaussian distribution.

3.4 Markov Chain Monte-Carlo Data Association And Parameter Estimation

The combinatorial solution space of $\mathbf{\Gamma}$ in (5) is extremely large and finding good tracklet associations is extremely challenging. Here we follow the MCMCDA sampling approach similar to [4, 12] and simultaneously estimate the parameters and $\mathbf{\Gamma}^*$. Figure 5 shows how the low- and high-level tracklets can be used for constructing long trajectories in a data association framework.

MCMC is a general method for generating samples from a distribution, p, by constructing a Markov chain in which the states are $\mathbf{\Gamma}$. At any state $\mathbf{\Gamma}$, a new proposal is introduced

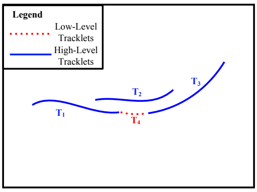


Figure 5: Data association and tracklet rejection. Formulating the likelihood as described in (8) makes it possible to reject some trajectories by considering them as false positives. Here, T_2 is a rejected tracklet. A low-level tracklet, T_4 is used to connect T_1 and T_3 based on motion consistency and temporal continuity.

using the distribution, $q\left(\mathbf{\Gamma}|\mathbf{\Gamma}'\right)$. Following [4], we consider three types of association as a result of the sampling process. The first randomly selects one tracklet and one trajectory. This affects the current state of the tracklet by associating it to the selected trajectory. The second, called swapping, postulates that, all tracklets constructing the two trajectories be swapped at a randomly chosen time. Finally, the third proposes a change of trajectory type. We decide which of the three $\mathbf{\Gamma}'$ should be accepted by employing the Metropolis-Hastings acceptance function [13] which defines the likelihood by:

$$A\left(\mathbf{\Gamma},\mathbf{\Gamma}'\right) = \min\left\{\frac{p\left(\mathbf{\Gamma}'\right)q\left(\mathbf{\Gamma}|\mathbf{\Gamma}'\right)}{p\left(\mathbf{\Gamma}\right)q\left(\mathbf{\Gamma}'|\mathbf{\Gamma}\right)},1\right\} \tag{14}$$

In addition, in order to estimate the model parameters described in section 3.3, we follow the approach presented in [12]. The latter uses MCMCDA sampling followed by an additional Metropolis-Hastings update for the parameters.

In summary, we have described a method to construct tracklets, given online observations. Then the probability of a tracklet being part of an actual track has been calculated by formulating the data association problem as a MAP estimation. Initial observations are taken to be the low- and high-level codebooks obtained by an event detection system. The low-level codebook codes the local motion patterns, while the high-level codebook codes global motion patterns in videos while considering the scene context. They are then tracked in consecutive frames, which produces two sets of dense tracks of small temporal length, called linklets. These dense linklets are then grouped to produce a small number of representative object tracklets. The representative tracklets are then linked to form long-term object trajectories. The data association framework we have adopted has two main advantages:1) It can reject certain tracklets by considering them as parts of false trajectories, and 2) It uses low-level tracklets as supportive information for filling the gaps between high-level tracklets, thereby producing smooth trajectories.

4 Experimental Results

The algorithm has been tested using the TUD [3] and CAVIAR datasets [1]. All parameters have been set experimentally, but most have remained identical for all sequences. In all cases, we have used the suggested parameters in [25] for codebook construction. We show quantitative comparisons with state-of-art methods, as well as visual results of our approach (see supplemental videos). We follow the same evaluation metrics as those in [19, 29, 31,

Table 1: Comparison of different tracking methods for the CAVIAR [1] and TUD dataset [3].

Dataset	Method	MT	ML	ID	FRAG	FAF
CAVIAR	Hao et al. [11]	84.6	0.7	11	18	0.085
	Yuan et al. [34]	84.6	1.4	11	17	0.157
	Li et al. [19]	85.7	35.7	15	20	-
	Song et al. [29]	84	4	8	6	-
	Ours	84.3	6.4	18	16	0.237
TUD	Yang et al. [31]	70	0	0	1	0.184
	Hao et al. [11]	60	0	0	3	0.014
	Ours	60	10	1	4	0.281

34]. These are Mostly Tracked (MT), which is the percentage of the trajectories covered by the tracker output more than 80% of the time; Mostly Lost (ML) which is the percentage of the trajectories covered by the tracker output less than 20% of the time; ID Switch (ID) which is the number of times that a trajectory changes its matched ground truth identity; fragments (FRAG), which is the number of times that a ground truth trajectory is interrupted (i.e., each time it is lost by the current hypothesis); and average False Alarms per Frame (FAF). The results indicate that although the correct detections we obtain with our algorithm are comparable to the state of the art, they include more false positives (see Table 1). Perhaps one can expect this, since no object detection is employed in our algorithm. Recall that the scene observations that we use are motion descriptors and do not incorporate object appearance, as do object-centric trackers.

5 Conclusions And Future Work

In this paper, we have introduced the use of motion descriptors obtained by an event detection algorithm for multiple object tracking. We have shown how pure motion descriptors for event detection could be employed to build a tracker without requiring an object model. Thus, each individual object is tracked by modeling only the temporal relationships between sequentially occurring local motion patterns. The algorithm is based on the descriptors of moving objects, obtained at two hierarchical levels. By considering both local and global motion patterns, two sets of initial tracks, called linklets, are obtained. Then, a set of sparse tracks, referred to as tracklets, was created by grouping linklets showing similar motion patterns. We then developed associations between them in order to produce longer trajectories.

Although our algorithm possesses no information regarding either an object's color pattern or a human body model, it achieves promising results on challenging data sets. As stated previously, the major drawback of our algorithm is the number of false positives and some problems in maintaining the trajectory identity when objects have similar shape and motion. Further improvements would include incorporating color information to reduce the number of ID switches.

References

[1] Caviar dataset, http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1.

[2] Saad Ali and Mubarak Shah. Floor fields for tracking in high density crowd scenes. In *Computer Vision - ECCV 2008*, volume 5303, pages 1–14. Springer Berlin Heidelberg, 2008.

[3] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1265–1272, 2011.

[4] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3457–3464, 2011.

[5] Marco Bertini, Alberto Del Bimbo, and Lorenzo Seidenari. Multi-scale and real-time non-parametric approach for anomaly detection and localization. *Compt. Vis. Image Und.*, 116(3):320–329, 2012.

[6] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1273–1280, 2011.

[7] G. J. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on*, volume 1, pages 594–601, 2006.

[8] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *Computer Vision - ECCV 2010*, volume 6315, pages 282–295. Springer Berlin Heidelberg, 2010.

[9] A. A. Butt and R. T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1846–1853, 2013.

[10] Huang Chang, Li Yuan, and R. Nevatia. Multiple target tracking by learning-based hierarchical association of detection responses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(4):898–910, 2013.

[11] Kuo Cheng-Hao, Huang Chang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 685–692, 2010.

[12] Weina Ge and Robert T Collins. Multi-target data association by tracklets with unsupervised parameter estimation. In *British Machine Vision Conference - BMVC*, volume 2, page 5, 2008.

[13] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall, 1998.

[14] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami. Fuzzy c-means for very large data. *IEEE Trans. Fuzzy Syst.*, PP(99):1–1, 2012.

[15] Haroon Idrees, Nolan Warner, and Mubarak Shah. Tracking in dense crowds using prominence and neighborhood motion concurrence. *Image and Vision Computing*, 32 (1):14–26, 2014.

[16] Liu Jingchen, P. Carr, R. T. Collins, and Liu Yanxi. Tracking sports players with context-conditioned motion models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1830–1837, 2013.

[17] L. Kratz and K. Nishino. Tracking with local spatio-temporal motion patterns in extremely crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 693–700, 2010.

[18] L. Kratz and K. Nishino. Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):987–1002, 2012.

[19] Zhang Li, Li Yuan, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8, 2008.

[20] Zhang Lu and L. van der Maaten. Structure preserving object tracking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1838–1845, 2013.

[21] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and Hu Wensheng. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 666–673, 2006.

[22] Yu Qian, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*, pages 1–8, 2007.

[23] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Computer Vision - ECCV 2012*, pages 343–356. Springer Berlin Heidelberg, 2012.

[24] Mehrsan Javan Roshtkhari and Martin D. Levine. A multi-scale hierarchical codebook method for human action recognition in videos using a single example. In *Computer and Robot Vision (CRV), 2012 Ninth Conference on*, pages 182–189, 2012.

[25] Mehrsan Javan Roshtkhari and Martin D. Levine. Online dominant and anomalous behavior detection in videos. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2609–2616, 2013.

[26] Mehrsan Javan Roshtkhari and Martin D. Levine. An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions, computer vision and image understanding. *Computer Vision and Image Understanding*, 117(10):1436–1452, 2013.

[27] Mehrsan Javan Roshtkhari and Martin D. Levine. Human activity recognition in videos using a single example. *Image and Vision Computing*, 31(11):864–876, 2013.

[28] Wu Shandong, Brian E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2054–2060, 2010.

[29] Bi Song, Ting-Yueh Jeng, Elliot Staudt, and Amit K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *Computer Vision - ECCV 2010*, pages 605–619. Springer-Verlag, 2010.

[30] D. Sugimura, K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1467–1474, 2009.

[31] Bo Yang and Ramakant Nevatia. Multi-target tracking by online learning a crf model of appearance and motion patterns. *International Journal of Computer Vision*, 107(2): 203–217, 2014.

[32] Hanxuan Yang, Ling Shao, Feng Zheng, Liang Wang, and Zhan Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011.

[33] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

[34] Li Yuan, Huang Chang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 2953–2960, 2009.