Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

Article

A Hierarchical Association Framework for Multi-Object Tracking in Airborne Videos

Ting Chen [1,2,*], Andrea Pennisi [3,2], Zhi Li [1], Yanning Zhang [1] and Hichem Sahli [3,2,1]

1 School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072, China
2 Dept. Electronics and Informatics, AVSP Lab, Vrije Universiteit Brussel, Brussels, Belgium
3 Interuniversity Microelectronics Center, Leuven, Belgium
* Correspondence: chentingnwpu@mail.nwpu.edu.cn; Tel.: +86-15102959004.

Abstract: Multi-object tracking (MOT) in airborne videos is a challenging problem due to the uncertain airborne vehicle motion, vibrations of the mounted camera, unreliable detections, size, appearance and motion of the moving objects as well as occlusions due to the interaction between the moving objects and with other static objects in the scene.To deal with these problems, this work proposes a four-stage Hierarchical Association framework for multiple object Tracking in Airborne video (HATA). The proposed framework combines data association-based tracking (DAT) methods and target tracking using a Compressive Tracking approach, to robustly track objects in complex airborne surveillance scenes. In each association stage, different sets of tracklets and detections are associated to efficiently handle local tracklet generation, local trajectory construction, global drifting tracklet correction and global fragmented tracklet linking. Experiments with challenging airborne video datasets show significant tracking improvement compared to existing state-of-art methods.

Keywords: Multiple object tracking, Airborne video, Tracklet confidence, Hierarchical association framework

1. Introduction

The goal of Multi-object tracking (MOT) in airborne videos is to estimate the state of multiple objects and conserving their identities under appearance and motion variations over time [1–4]. This is a challenging problem because of the uncertain motion of the airborne vehicle, the vibration of the non-stationary camera and the partial occlusions of the objects [5]. Much attention has been paid in data association-based tracking (DAT) methods [6] along with the improvement of object detection methods, which provide reliable detections even in complex scenarios [7,8]. To produce the final trajectories for each tracked object, most of the DAT approaches rely on the detection accuracy [9] and the used affinity model [10,11], integrating multiple visual cues (*i.e.* appearance and motion), to find the linking probabilities between detection responses and tracklets in the subsequent frames [7,8].

Existing object detectors can be roughly categorized into off-line and on-line methods. The off-line detectors use a pre-defined strategy to learn the patterns representing the object's appearance by using various kind of features. They are widely used in MOT because they are less sensitive to image noise [12–14]. In the aerial surveillance domain, the several types of targets, their fine grained size and and appearance differences (due to their own movement as well as motion of the UAV) make such methods difficult to train for achieving reasonable detection performance. For such reasons, on-line detectors using motion compensation-based models [10,11,15–18] are more popular in airborne videos analysis. The objects with different motion and appearance cues compared to the background can be automatically detected without any prior information. Moreover, the low computational complexity of such algorithms makes them suitable for embedded platforms on-board of unmanned aerial vehicles (UAV).

Generally speaking, the performance of the existing compensation-based detectors are usually a tradeoff between the detection rate and the false alarm rate. This is because an accurate estimation of the

© 2018 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

camera's motion model cannot be computed and is time consuming. Most of the compensation-based algorithms assume a simple camera model such as the affine or projective camera model [19]. To reduce false detections, Yin *et al.* [20] adopted a detection method based on the forward-backward motion history images (MHI) to localize moving objects. The required forward motion history makes this method not suitable for real-time applications. To analyze the long-term object motion pattern, Yu *et al.* [7] used a Tensor Voting computational framework to detect and segment moving objects. This method might be impractical in many real-world applications because it requires the full image sequence for the global analysis step. Considering the errors which could arise from motion compensation, Kim *et al.* [21] proposed a spatio-temporal distributed Gaussian model, while a dual-model single Gaussian model (SGM) has been adopted by Yi *et al.* [22]. These approaches reduce many false detections and achieve real-time performance with a low computation complexity, but they produce miss detections and still show unsatisfactory performance in complex scenes. In [19], the authors combined the spatio-temporal properties of moving objects and the SGM background model to reduce miss detections and false detections.

Occlusions are the main problem of both off-line and on-line detectors [8,23–25]. To overcome such problems, some recently proposed tracking algorithms recover the trajectories of all targets via a two stages association framework [14,24,26]. In a first stage, a set of reliable short tracklets are locally generated by linking the detections to tracklets. In a second stage, to build longer tracklet and deal with frequent occlusions, a global optimal solution is obtained by solving a maximum a posteriori problem (MAP) problem using various optimization algorithms. This two-stage DAT approach can be applied for time critical applications since they sequentially build trajectories based on a frame-by-frame association. However, DAT can not be directly adopted in airborne videos as both the local and global association stages require efficient object detection with accurate object's location and size [24,25].

In this paper, we take into account most of the limitations of the previous methods and propose an efficient Hierarchical Association framework for multiple object Tracking in Airborne platforms (HATA). We adopt the SGM [22] as on-line object detector, and motivated by the works of Bae *et al.* [14] and Ju *et al.* [27] we formulate the MOT problem as a Hierarchical DAT based on tracklet confidence. The proposed hierarchical association framework consider a four stage approach for data association: a local tracklet generation stage, followed by a local trajectory construction step, than a global drifting tracklet correction stage, and finally a global fragmented tracklet linking step. To this end, the tracklets and the detections are divided into several groups depending on the tracklet confidence and association results. Furthermore, for each tracklet we maintain a Kalman Filter tracker and an appearance-based tracker, build upon Compressive Tracking[28,29] to deal with (i) target's changes in appearance, (ii) occlusions, as well as (iii) motion-less tracklets. Moreover, the appearance-based tracker is used to update the tracklets state for managing unreliable associations.

In our algorithm, we define two types of occlusions: *Occlusion-I* when two tracked objects overlaps, and *Occlusion-II* when the object is occluded by a static obstacle within the environment (*e.g.* trees). The *Occlusion-I* is handled via the detection-tracklet association. The *Occlusion-II* cases are more challenging because of the lack of hard temporal (frame-to-frame) constraints. For such cases, we apply an object re-identification approach, based on tracklet-to-tracklet matching using a set of appearance and motion features extracted around the target objects [30]. The proposed MOT framework robustly tracks multiple objects in complex scenes and can be fully implemented for real-time applications.

2. Related Works

In this section, we give an overview of state-of-art methods for MOT in airborne surveillance, the main data association-based tracking (DAT) approaches on which we based our work, and finally basic object re-identification methods.

MOT in airborne videos: A number of methods for detecting and tracking objects from airborne platforms have been developed in the last decades [2–4,8,31,32]. Early approaches adopt optical flow [33] or feature points [5,7] to detect and estimate the trajectories of the moving objects. Yu and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

Medioni, in [7], estimated the motion flow in each frame based on a cross-correlation methods, and then a Tensor Voting approach is used to analyze the optical flow to segment moving objects. The motion history image (MHI) method [20] is used to generate the initial segmentations, and the tracklets are generated by using the appearance similarity and flow dynamics between the segmented regions. The mean-shift algorithm is applied to predict the location in the motion field. The end (entry and exit) information of a flow is imposed as environmental constraints when associating tracklets. However, in their tracking framework, a relatively long sequence is needed to detect motion patterns, which causes tracking delays and hence not practical for real-time tracking. In [34], the Kanade Lucas Tomasi (KLT) features and a temporal differencing method have been used to separate moving vehicles from the background. Local features are clustered to establish different motion layers for vehicle tracking. This method is robust to partial occlusion. However, it fails locating vehicles when the background is highly cluttered. In order to solve this problem, they proposed in [35] a novel tracking framework based on particle filter method. An estimate of the vehicle's ego motion is incorporated into the particle filter framework to guide particles moving towards the target position.

Prokaj *et al.* [10] presented a method for vehicle tracking in an aerial surveillance context. First, the moving object detection was done using background subtraction. The background is modeled as the mode of a (stabilized) sliding window of frames [10]. Then, they formulated the data association problem as an inference in a set of Bayesian networks using motion and appearance consistency. Such an approach avoids the exhaustive evaluation of data association hypotheses and provides a confidence estimate of the solution. Moreover, it handles split-merge observations. In [36], a collaborative framework consisting of a two-level tracking process has been adopted to track objects as groups. The higher-level process builds a relevance network and divides objects into different groups, where the relevance is calculated based on the information obtained from the lower level processes. In [16], Prokaj *et al.* handled the missed detections by generating virtual detections. Any time a detection in frame t does not have an object to link to in frame $t+1$, a virtual detection is generated by predicting the location and appearance of the target in the next frame. This procedure is also recursive, so that when a newly added virtual detection does not have nearby detections in the next frame, the process is repeated. In [18], Prokaj *et al.* presented a multiple target tracking approach that does not exclusively rely on background subtraction and is better to track targets through stops. It accomplishes this by effectively running two trackers in parallel: one based on detections from background subtraction providing target initialization and reacquisition, and one based on a target state regressor providing frame to frame tracking. The detection based tracker provides accurate initialization by inferring tracklets over a short time period (5 frames). The initialization period is then used to learn a non-parametric regressor based on target appearance templates, which can directly infer the true target state from a given target state sample in every frame. When the regressor based tracker fails (loses a target), it falls back to the detection based tracker for re-initialization. However, the regressor's output would be meaningless when the target is not visible without information.

Two-stage DAT: Xing *et al.* [24] proposed to combine local linking and global association as a two-stage data association-based tracking (DAT) framework. They produce locally optimized tracklets by associating observations with tracklets, and global tracklets by associating fragmented tracklets. They used a greedy method for local association, and a predefined appearance model. Similarly, Bae *et al.* [26] proposed a Bayesian data association approach in which a tracklet existence probability is used during the local stage to assign the detections to tracks. Such an approach handles partial occlusions. The tracklet-to-tracklet global association stage is made by using an adjusted tracklet management system to link fragmented tracklets under long-term occlusions. In a more recent work, Bae *et al.* [14] formulated the multi-object tracking problem as a two-stage DAT based a tracklet confidence. The tracklets with a high confidence are sequentially grown with the provided detections. The fragmented tracklets, with low confidence, are linked to the other tracklets and detections without any iterative and expensive association. However, the long-term occlusions have not been considered by the authors. To improve upon the approach of [14], recently, Ju *et al.* [27] proposed a four-stage hierarchical association

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

framework based on an on-line matching strategy and tracklet confidence. The tracklets and detections are divided into several groups depending on several cues obtained from the matching results and a proposed tracklet confidence. In each matching stage, different sets of tracklets and detections are associated to handle frequent and prolonged occlusions, abrupt motion change of objects, and unreliable detections. In our framework, we follow the four stages ideas of [27], however using an on-line detection approach and the involvement of multiple appearance-based trackers.

Re-identification: Object re-identification (Re-ID) has been an active research topic in the past few years. It has been intensively studied for stationary inter-camera target associations [37] for long-term object tracking. A typical Re-ID algorithm is based on appearance modeling and matching [38,39]. The appearance modeling often uses low-level features such as color, texture, gradient, or a combination of them, to build more discriminative appearance descriptors [37,38]. Many successful Re-ID algorithms have been proposed for special target Re-ID systems [37–40], such as pedestrians and vehicles. Liu *et al.* [37] exploit a spatio-temporal body-action model by using fisher vector learning to solve the large appearance variation of a pedestrian. Zapletal *et al.* [38] proposed an approach based on a linear regression model using color histograms and histograms of oriented gradients for vehicle re-identification in a multiple cameras scenario. Liu *et al.* [39] proposed a fusion model of low-level features and high-level semantic attributes for vehicle Re-ID. In our framework we follow the ideas of object matching using appearance and motion cures for object re-identification after long term occlusion.

3. Framework Overview

3.1. Framework Overview

We follow the notations defined in [14]. An object i appearing in a frame t is denoted as present using a binary function $\phi_t^i = 1$; otherwise $\phi_t^i = 0$. When $\phi_t^i = 1$ the state of the object i is represented as $\mathbf{x}_t^i = (\mathbf{p}_t^i, w_t^i, h_t^i, \mathbf{v}_t^i)$, where $\mathbf{p}_t^i = (p_t^i(x), p_t^i(y)), w_t^i, h_t^i$ and $\mathbf{v}_t^i = (v_t^i(x), v_t^i(y))$ are, the object's center location, width and height of its bounding box, and its velocity, respectively. We then define the tracklet T_t^i of the object i as a set of states up to frame t, and denote it as $T_t^i = \{\mathbf{x}_k^i | \phi_t^i = 1 \leq t_s^i \leq k \leq t_e^i \leq t\}$, where t_s^i and t_e^i are the start- and end-frame of the tracklet. In addition we denote by $\mathbb{T}_t = (\mathbf{x}_t^1, \mathbf{x}_t^2, \cdots, \mathbf{x}_t^{n_x})$ the states of all the n_x objects in the t-th frame, and by $\mathbb{T}_{1:t} = \{T_t^1, T_t^2, \cdots, T_t^{n_x}\}$ the set of tracklets of all the n_x objects up to frame t. Correspondingly, we denote by $\mathbf{d}_t^j = (\mathbf{p}_d, w_d, h_d)_t^j$ a j-th detected observation at frame t, with $\mathbf{p}_d, w_d$ and h_d, the position of the center location (given by its coordinates $(p(x), p(y))$), width and hight of the detected blob, respectively. We also define $\mathbb{D}_t = \{\mathbf{d}_t^j; 1 \leq j \leq n_d\}$ the set of the n_d detected blobs (observations) at frame t. All the observations associated to object i up to frame t are referred to $d_{1:t}^i = \{\mathbf{d}_1^i, \cdots, \mathbf{d}_t^i\}$, and $\mathbb{D}_{1:t} = \{d_{1:t}^1, \cdots, d_{1:t}^{n_d}\}$ the set of all observation up to frame t. Following the approach of [14], the objective of MOT is to find the optimal $\mathbb{T}_{1:t}$ by maximizing the posterior probability for a given $\mathbb{D}_{1:t}$ as

$$\mathbb{T}_{1:t}^* = \arg\max_{\mathbb{T}_{1:t}} p(\mathbb{T}_{1:t} | \mathbb{D}_{1:t}). \tag{1}$$

Using a tracklet confidence, $\Omega(T_t^i) \in [0, 1]$, estimated as the affinity between a tracklet and and its associated detections, Bae and Yoon [14] formulated the above problem as

$$\begin{aligned} \mathbb{T}_{1:t}^* &= \arg\max_{\mathbb{T}_{1:t}} p\left(\mathbb{T}_{1:t} | \mathbb{T}_{1:t}^{(h)}, \mathbb{T}_{1:t}^{(l)}\right) \times p\left(\mathbb{T}_{1:t}^{(h)}, \mathbb{T}_{1:t}^{(l)} | \mathbb{D}_{1:t}\right) \\ &= \arg\max_{\mathbb{T}_{1:t}} p\left(\mathbb{T}_{1:t} | \mathbb{T}_{1:t}^{(h)}, \mathbb{T}_{1:t}^{(l)}\right) \times \underbrace{p\left(\mathbb{T}_{1:t}^{(l)} | \mathbb{T}_{1:t}^{(h)}, \mathbb{D}_{1:t}\right)}_{UA} \underbrace{p\left(\mathbb{T}_{1:t}^{(h)} | \mathbb{D}_{1:t}\right)}_{RA} d\mathbb{T}_{1:t}^{(h)} d\mathbb{T}_{1:t}^{(l)} \end{aligned} \tag{2}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

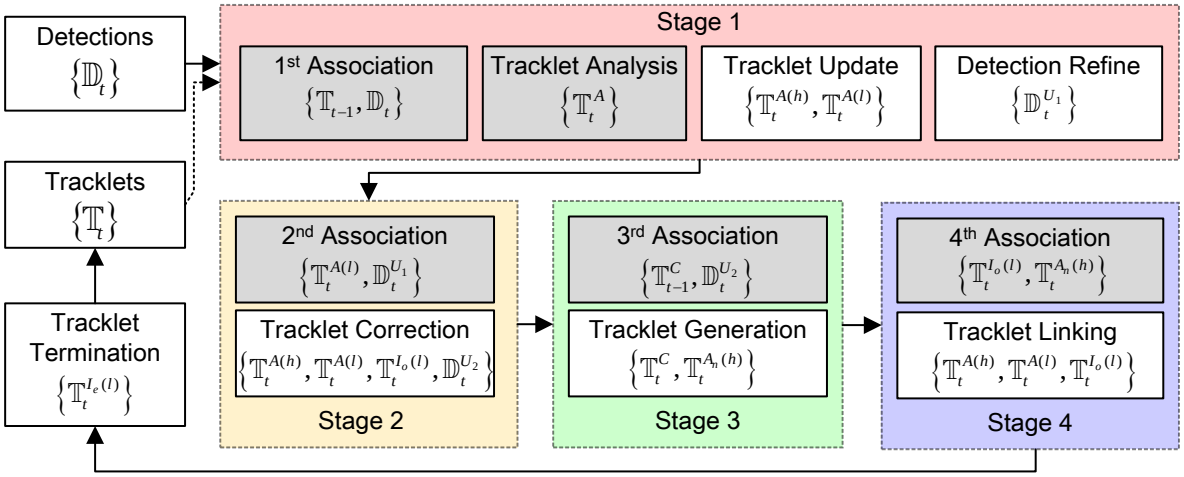


Figure 1. The framework of the proposed algorithm. The symbols in the gray bounding box are the input to the processing stage and the symbols in the white bounding box are the output.

where $\mathbb{T}_{1:t}^{(h)}$ and $\mathbb{T}_{1:t}^{(l)}$ represent a set of tracklets with high confidence (*i.e.* $\Omega(T^i) > th_\Omega$ with $th_\Omega = 0.5$), and a set of tracklets with low confidence. In the above equation, the tracking problem is solved in two phases. In a first phase, tracklets with high confidence are locally associated with provided detections (part denoted by RA), while tracklets with low confidence, which are more likely to be fragmented, are globally associated with other tracklets and detections in a second global phase (UA).

In our framework, we follow the same ideas, though, we use the four-stage hierarchical association concept proposed in [27] to find the optimal assignments for the local tracklet-to-detection or global tracklet-to-tracklet. However, we extend the approach of [27] by considering an appearance-based tracker associated to each tracked object, to better characterize motion-less or occluded objects, along with a detection refinement process to manage inaccurate detections. The flowchart of the proposed method is shown in Figure 1.

At each stage, the tracklet-to-detection or tracklet-to-tracklet assignment is solved by using the Hungarian algorithm approach [41]. For each frame, we first apply a motion compensation-based object detector to detect objects of interest (see Section 3.3). After the local tracklet-to-detection association of Stage-1, a tracklet state analysis, involving an appearance-based tracker (Section 3.5) and a Kalman Filter tracker (Section 3.6), is used to characterize motion-less or occluded objects (Section 4.1.2), and a detection refinement process is used to manage inaccurate detections which have not been associated to tracklets (Section 4.1.3). After a first global tracklet-to-detection association of Stage-2, the un-matched detections are used to generate new tracklets in Stage-3. Some of these new tracklets are used to re-link the lost tracklets during the global tracklet-to-tracklet association of Stage-4. Stage-4 also handles tracklets termination. All the symbols used in Figure 1 are introduced in the following.

3.2. Hierarchical groups of detections and tracklets

We follow the ideas of Ju *et al.* [27] and define hierarchical groups of tracklets and detections as follows. In each frame t an object detector (see Section 3.3) detects objects of interest and produces the set $\mathbb{D}_t$ of detections, the elements of which are associated to tracklets during the first two association stages. During the association process the set $\mathbb{D}_t$ is decomposed in 4 sets: $\mathbb{D}_t^{M_1}$ and $\mathbb{D}_t^{U_1}$ being the matched and un-matched detections during Stage-1, respectively, and $\mathbb{D}_t^{M_2} \subset \mathbb{D}_t^{U_1}$) and $\mathbb{D}_t^{U_2} \subset \mathbb{D}_t^{U_1}$ the matched and un-matched detections during Stage-2, respectively.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

During the hierarchical association process, the set of tracklets in the t-th frame, $\mathbb{T}_t$, will be decomposed into three disjoint subsets:

$$\mathbb{T}_t = \mathbb{T}_t^A \cup \mathbb{T}_t^C \cup \mathbb{T}_t^I \tag{3}$$

with $\mathbb{T}_t^A$ the active tracklet set, $\mathbb{T}_t^C$ the candidate tracklet set, and $\mathbb{T}_t^I$ the inactive tracklet set.

- The active tracklets set, $\mathbb{T}_t^A$, includes the tracklets corresponding to the current existing objects, composed of three disjoint subsets:

$$\mathbb{T}_t^A = \mathbb{T}_t^{A_n(h)} \cup \mathbb{T}_t^{A(h)} \cup \mathbb{T}_t^{A(l)} \tag{4}$$

  with $\mathbb{T}_t^{A_n(h)}$ the new active tracklet (recently generated tracklet) set with high confidence, $\mathbb{T}_t^{A(h)}$ the reliable active tracklet set with a high confidence, and $\mathbb{T}_t^{A(l)}$ the un-reliable active tracklet set with low confidence. They are formally defined as follows.

$$\mathbb{T}_t^{A_n(h)} = \{T_t^i | L(T_t^i) \leq th_L\} \tag{5}$$

$$\mathbb{T}_t^{A(h)} = \{T_t^i | L(T_t^i) > th_L, \Omega(T_t^i) \geq th_\Omega\} \tag{6}$$

$$\mathbb{T}_t^{A(l)} = \{T_t^i | L(T_t^i) > th_L, \Omega(T_t^i) < th_\Omega\} \tag{7}$$

  where th_L is a threshold on the tracklet length $L(\cdot)$ for distinguishing new tracklets from old ones, th_Ω is a threshold on the tracklet confidence $\Omega(\cdot)$ for characterizing whether the tracklet is reliable or un-reliable (*e.g.*, likely to drift or lost).
- The candidate tracklet set $\mathbb{T}_t^C$ includes the tracklets waiting for enough matched detections in the third stage before being added as new active tracklets.
- The inactive tracklet set $\mathbb{T}_t^I$ includes two disjoint subsets:

$$\mathbb{T}_t^I = \mathbb{T}_t^{I_o(l)} \cup \mathbb{T}_t^{I_e(l)} \tag{8}$$

  where $\mathbb{T}_t^{I_o}$ and $\mathbb{T}_t^{I_e}$ represent the lost tracklet set and the terminated tracklet set, respectively. $\mathbb{T}_t^{I_o}$ includes tracklets corresponding to the temporary lost objects, due to long-term occlusions, while the terminated tracklet set $\mathbb{T}_t^{I_e}$ includes the disappeared objects. Each subset is defined as:

$$\mathbb{T}_t^{I_o(l)} = \{T_t^i | L(T_t^i) > th_L, \Omega(T_t^i) < th_I, t - t_e^i < th_e\} \tag{9}$$

$$\mathbb{T}_t^{I_e(l)} = \{T_t^i | L(T_t^i) > th_L, \Omega(T_t^i) < th_I, t - t_e^i \geq th_e\} \tag{10}$$

  where th_I is a threshold for distinguishing active and non-active tracklets, t_e^i is the last frame of the active tracklet, and th_e is a threshold to terminate the tracklet.

Figure 2 illustrates tracklet's status changes in time according to the tracklet confidence. The overall process is summarized here after. In Stage-1, we determine the best associations between the previous set of active tracklets $\mathbb{T}_{t-1}^A$ and the detection set $\mathbb{D}_t$ at frame t. Then, the states of the matched tracklets are updated based on the associated detections and the appearance-based predictions. For the un-matched tracklets, a tracklet analysis (Section 4.1.2), using the appearance-based predictions, is performed to update their states. According to the tracklet analysis, some tracklets are updated using the appearance-based prediction and others are updated using a motion-based prediction. Then, the tracklet confidence values are estimated using the associated detections. Based on the confidence value, a tracklet is assigned to the sub-sets $\mathbb{T}_t^{A(h)}$ or $\mathbb{T}_t^{A(l)}$. The inaccurate detections from the un-matched detection set, $\mathbb{D}_t^{U_1}$, which are overlapping with active tracklets, are deleted or resized via a detection refinement process (see Section 4.1.3).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

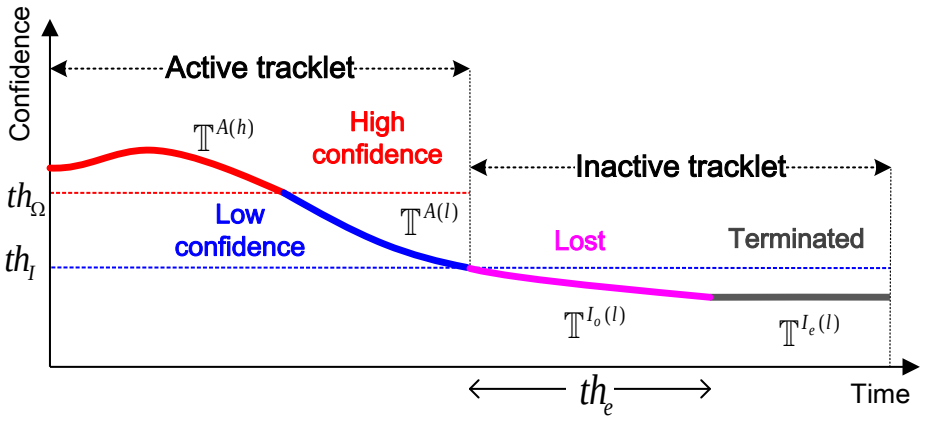


Figure 2. Tracklet Status.

In Stage-2, the association between the unreliable tracklets, $\mathbb{T}_t^{A(l)}$, and the un-matched detections $\mathbb{D}_t^{U_1}$, is performed to handle drifting targets caused by frequent occlusions. The states of the tracklets which have been matched with detections are updated using the associated detections and assigned to $\mathbb{T}_t^{A(h)}$. The un-matched (to detections) tracklets are moved to the inactive tracklets set, $\mathbb{T}_t^{I_o(l)}$, when their confidence $\Omega(T_t^i)$ is lower than a given threshold th_I (i.e $\Omega(T_t^i) < th_I$). Then, in Stage-3, the association between candidate tracklets, $\mathbb{T}_{t-1}^{C}$, and the remaining un-matched detections, $\mathbb{D}_t^{U_2}$, is performed to update the set of candidate tracklets, $\mathbb{T}_t^{C}$, or generate new active tracklets in $\mathbb{T}_t^{A_n(h)}$.

Finally, in Stage-4, the association between the lost tracklets, $\mathbb{T}_t^{I_o(l)}$, in the inactive tracklets set, and new tracklets is performed to merge fragmented tracklets of the same object after long-term occlusions. The inactive tracklets which are not associated to new tracklets within $t - t_e^i \geq th_e$ are terminated and included to the set $\mathbb{T}_t^{I_e(l)}$ after the fourth stage. The 4 stages are detailed in Section 4.

3.3. On-line detection

In our framework, we use the method described in [19,22] as on-line detector. The detector models the background through a dual-mode SGM and compensates the motion of the camera by mixing neighbor models. Modeling through a dual-mode SGM prevents the background model from being contaminated by the foreground pixels, while still allows the model to be adaptive to the changes of the background. After the detection step, a post processing step, consisting of dilation and erosion, is performed to merge scattered detections. Finally, a bounding box is estimated around every detected blob. The detector achieves real-time performances with low computation complexity, while it produces miss detections and false detections.

The detection results are illustrated in Figure 3. Most of the missed detections and false detections are caused by occlusions or motion-less objects. Figure 3(a) shows a reliable detection bounding box, which perfectly encloses the object. However, in cases of slowly moving objects, the bounding box can cover part of the object (Figure 3(b)). The detector can also provide 2 (or more than 2) bounding boxes for a single object (Figure 3(c)). In the following, the above cases are denoted as *Motion-I* type detection. Furthermore, motion-less objects cannot be detected with the used algorithm, and we denote such cases as *Motion-II* type detection, as shown in Figure 3(d).

In our algorithm, we define two occlusion cases: *Occlusion-I* and *Occlusion-II*. *Occlusion-I* include all occlusions caused by other tracked objects. We define the more front object as *occluder* and the occluded object as *occluded*. In general, a good detection bounding box can be obtained for the occluder object. However, when two objects (or more than 2) are very close, only one detection is obtained (Figure 3(e)) and the size of the bounding box matches one of the two objects (Figure 3(f). The *Occlusion-II* case includes the occlusions which are caused by static objects (obstacles) within

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

the environment (*e.g.*, trees and buildings). This case is more challenging because of the lack of hard temporal (frame-to-frame) constraints and not reliable object representation form the detected bounding boxes, and so the obtained bounding boxes do not match the object size, as shown in Figures 3(g). Also, *Occlusion-II* case includes objects that are fully occluded by the environment (Figure 3(h)).

To deal with the above described unreliable detections, we implemented a detection refinement process (see Section 4.1.3) in which the states of the current tracklets are used to analyze and refine unreliable detections for further tracklets-to-detections associations.

3.4. Tracklet confidence

The tracklet confidence, $\Omega(T_t^i)$, expresses how well the constructed tracklet matches the real trajectory of the target. In our framework it is defined as:

$$\Omega(T_t^i) = \begin{cases} \Omega_\Lambda(T_t^i) \cdot \Omega_o(T_t^i) & \text{,if } \phi_t^i = 1 \\ \Omega(T_{t-1}^i) \cdot w_p^i & \text{,if } \phi_t^i = 0 \end{cases} \tag{11}$$

$$\Omega_\Lambda(T_t^i) = \frac{1}{L_T} \sum_{k \in [t_s^i, t_e^i], \phi_k^i = 1} \Lambda^J \left(T_t^i, \mathbf{d}_k^i \right) \tag{12}$$

$$\Omega_o(T_t^i) = 1 - \exp\left(-w^d \cdot \sqrt{L(T_t^i) - L_M} \right) \tag{13}$$

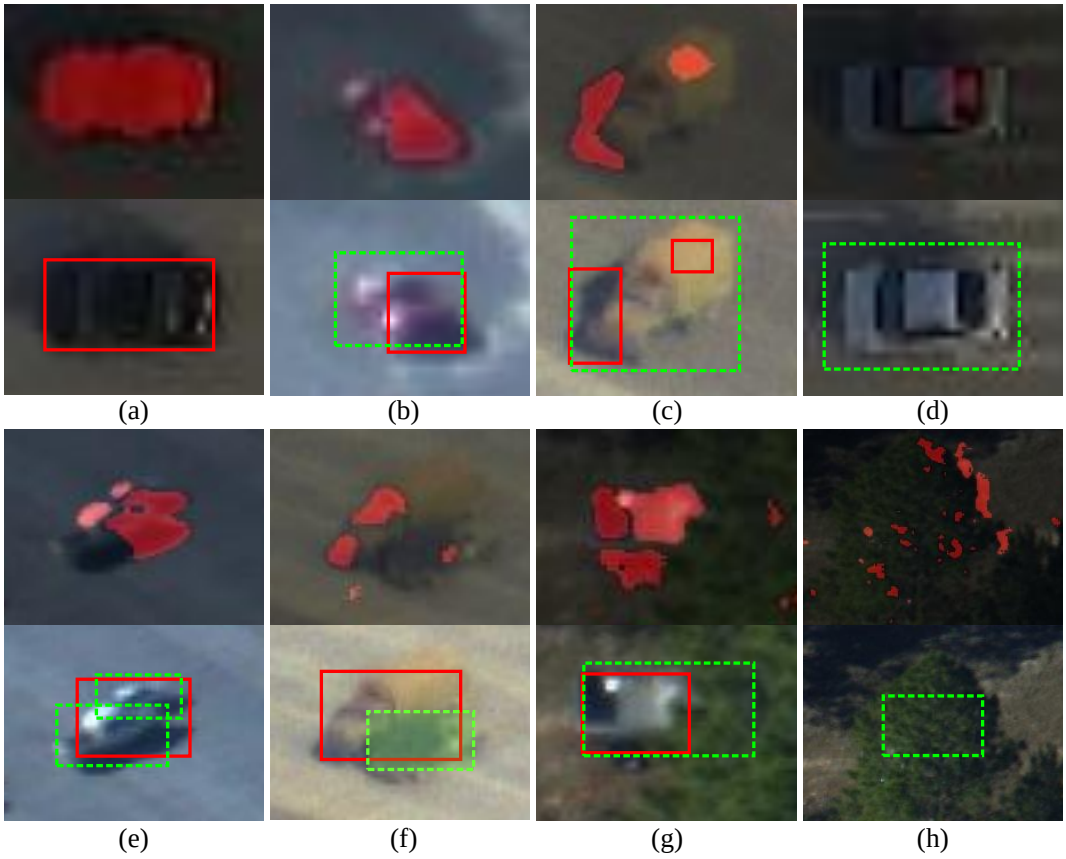


Figure 3. Illustration of detection results. The bounding boxes with the red color and dotted green color are the detection results and the ground truth, respectively.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

where $\Omega_\Lambda(T_t^i)$ and $\Omega_o(T_t^i)$ are the affinity and observation confidence terms, respectively. Depending on the association stage, $J \in [1,4]$, the affinity confidence term $\Omega_\Lambda(T_t^i)$ is calculated using an affinity model $\Lambda^J\left(T_t^i, \mathbf{d}_k^i\right)$ involving the appearance, shape and motion of the objects. The used affinity models are defined in Section 4. The observation confidence term $\Omega_o(T_t^i)$ is computed using the tracklet length $L(T_t^i)$ and $L_M = (t_e^i - t_s^i + 1 - L_T)$, while, w^d is a control parameter relying on the performance of the detection and it will be discussed in the experimental Section 5.2.1. w_p^i is a control parameter relying on the performance of the i-th tracklet prediction and defined in Eq. (24) Section 4.1.2. The observation confidence $\Omega_o(T_t^i)$ decreases rapidly if the detection responses of the tracklet T_t^i are missing over L_M frames (heavily occluded tracklet). A tracklet is considered as a reliable tracklet, $T_t^{i(h)} \in \mathbb{T}_t^{A(h)}$, if it has a high confidence, *i.e.* $\Omega(T^i) > th_\Omega$ (th_Ω is set to 0.5 in our experiment); otherwise it is considered as a fragmented tracklet with low confidence, $T_t^{i(l)} \in \mathbb{T}_t^{A(l)}$.

3.5. Appearance based prediction

Object appearance modeling is important in our framework for both tracklet state analysis and detection refinement processes. To maintain reliable appearance model of the tracklets we make use of the discriminative appearance model of the compressive tracking (CT) algorithm of [28,29]. For each object i we associate a Fast-CT tracker (FCT) as proposed in [29].

We summarize here after the main components of the CT algorithm being (1) Naïve Bayes classifier update, and (2) target detection, the reader is refereed to [28,29] for the algorithmic details.

1. Naïve Bayes classifier update: CT samples some positive samples near the current target location, and negative samples far away from the object center. To represent the sample $\mathbf{z} \in \mathbb{R}^{w \times h}$, CT uses a set of rectangle features and extracts the features with low dimensionality using a very sparse measurement matrix $R \in \mathbb{R}^{n \times m}$, $\mathbf{a} = R\mathbf{b}$, with $\mathbf{b} \in \mathbb{R}^m$ ($m = (w \times h)^2$) the high-dimensional image feature, formed by concatenating the convolved target images (represented as column vectors) with rectangle filters; $\mathbf{a} \in \mathbb{R}^n$ the lower-dimensional compressive features with $n \ll m$. Each element a_i in the low-dimensional feature $\mathbf{a}$ is a linear combination of spatially distributed rectangle features at different scales. A simple Bayesian model is used to construct a classifier based on the positive ($y = 1$) and negative ($y = 0$) sample features. The compressive sensing algorithm assumes that all the lower-dimensional samples of the target are independent of each other, $H(\mathbf{a}) = \sum_{k=1}^{n} \log\left(\frac{p(a_k|y=1)}{p(a_k|y=0)}\right)$. The parameters of the Naïve Bayes classifier are incrementally updated according to the four parameters of the classifier's Gaussian conditional distribution $(\mu^1, \sigma^1, \mu^0, \sigma^0)$ and a update rate $\lambda > 0$.
2. Target detection: The candidate region corresponding to the maximum $H(\mathbf{a})$ is regarded as the tracking target location:

$$\mathbf{l}_t^* = \arg\max_{\mathbf{a}} H(\mathbf{a}). \tag{14}$$

See [28] for the detailed implementation. The overall performance, in terms of speed and tracking accuracy, of the CT algorithm has been significantly improved by the fast compressive tracking (FCT) presented in [29]. While the CT samples in a fixed rectangular region in single pixel steps, the FCT improves upon this by introducing a coarse-to-fine search strategy to reduce the computational complexity in the detection procedure.

In our implementation, for each new active tracklet $T_t^{i(h)} \in \mathbb{T}_t^{A_n(h)}$, the latest object state $\mathbf{x}_t^i = \left(\mathbf{p}_t^i, w_t^i, h_t^i, \mathbf{v}_t^i\right)$ is used to initialize an FCT-based tracker and keep the four parameters of its appearance model $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$. At each new frame t, the coarse-to-fine sampling strategy [29] is used to crop a set of candidate samples around the previous location of the target. The sample which obtains the maximal classifier response (Eq(14)) is selected as the current appearance-based prediction of the target's location,$\mathbf{lc}_t^i$. The FCT-tracker outputs a target-state denoted as $\mathbf{c}_t^i = (\mathbf{lc}_t^i, wc_t^i, hc_t^i)$, with wc_t^i and hc_t^i the width and hight of the corresponding bounding box, respectively. In our implementation of the FCT algorithm, we use a dynamic learning rate defined as $\lambda = \Omega(T_t^i)$ to update the target's

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

appearance. The parameters of the appearance model are re-initialized at each 5 frames to avoid large scale variation in both x and y directions. For the tracklet $T_t^{i(l)} \in \mathbb{T}_t^{A(l)}$, we set $\lambda = 0$ to stop the update. For the tracklet $T_t^{i(l)} \in \mathbb{T}_t^{I_e(l)}$, we delete the appearance model.

3.6. Motion based prediction

The motion model describes the dynamic movement of tracked objects, which can be used to predict the potential position of the objects in the future frames, especially under occlusion. In most cases, it is assumed that a given object moves smoothly in the world and hence, their image apparent motion is also smooth [9]. A linear motion model based on Kalman Filter (KF) is the most used model in MOT [24,42,43]. Given the motion model of a moving object, KF provides an optimal estimate of its position at each time step.

In our framework, we use KF to predict the position and velocity of a target object. For each tracked object $\mathbf{x}_t^i = (\mathbf{p}_t^i, w_t^i, h_t^i, \mathbf{v}_t^i)$, we maintain a Kalman Filter state $\mathbf{xk}_t^i = \left(\mathbf{pk}_t^i, \mathbf{vk}_t^i\right)$. We use the propagation equation of the KF to predict the object's state when it is not associated to any detection, and use the update equation of the KF to update the state of the object when it is associated to a detection. In such case, the observation vector is the center location of the associated detected blob given by its coordinates $\mathbf{p}_d = (p(x), p(y))$. The state transition matrix is defined as $A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, and the observation matrix defined as $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$.

4. Four stages hierarchical association framework

In this section we describe the different stages of the proposed framework for sequentially and robustly tracking multiple objects.

4.1. Stage-1: Local progressive trajectory construction

The first association stage solves the assignment problem between the active tracklets, $\mathbb{T}_{t-1}^A$, and the current detections, $\mathbb{D}_t$, to progressively build object trajectories. The input pairs for this stage are $\{(T_{t-1}^i, \mathbf{d}_t^j) | \forall T_{t-1}^i \in \mathbb{T}_{t-1}^A, \forall \mathbf{d}_t^j \in \mathbb{D}_t\}$, and the association is evaluated using the following affinity model:

$$\Lambda^1\left(T_{t-1}^i, \mathbf{d}_t^j\right) = \Lambda_a^1\left(T_{t-1}^i, d_t^j\right) \Lambda_s^1\left(T_{t-1}^i, \mathbf{d}_t^j\right) \Lambda_m^1\left(T_{t-1}^i, \mathbf{d}_t^j\right) \tag{15}$$

where $\Lambda_a^1\left(T_{t-1}^i, \mathbf{d}_t^j\right)$, $\Lambda_s^1\left(T_{t-1}^i, \mathbf{d}_t^j\right)$ and $\Lambda_m^1\left(T_{t-1}^i, \mathbf{d}_t^j\right)$ are the appearance affinity, the shape affinity and the motion affinity, respectively. They are defined in the following section.

4.1.1. First association via the affinity score

To rapidly evaluate the affinity appearance for real-time applications, a template matching-based approach is used. Each active tracklet maintains the latest template and the historical template set consisting of N_H^a templates (N_H^a=10 in our experiments). The templates of the detections and tracklets are obtained using a 24-bin Red-Green-Intensity histogram extracted from the image patches within the bounding box. All patches are resized to 64×64 pixels to be invariant to object scaling. Let $\chi_{\mathbf{d}^j}$ be the template of a detection $\mathbf{d}_t^j$, $\chi_{T^i}^L$ be the latest template of the tracklet T_{t-1}^i, and $H_{T^i} = \{\chi_{T^i}^k, k \in [1, N_H^a]\}$ be the historical template set of the tracklet T_{t-1}^i, The Bhattacharyya distance is used to evaluate the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

similarity between two templates, and we define the appearance affinity, Λ_a^1 in Eq.(15), of a tracklet T_{t-1}^i and a detection $\mathbf{d}_t^j$ as:

$$\Lambda_a^1\left(T_{t-1}^i, \mathbf{d}_t^j\right) = \omega_a \cdot \rho(\chi_{T^i}^L, \chi_{\mathbf{d}^j}) + (1-\omega_a) \max_k \rho(\chi_{T^i}^k, \chi_{\mathbf{d}^j}) \tag{16}$$

where $\rho(\cdot,\cdot)$ is the Bhattacharyya distance, and $\omega_a = \Omega(T_{t-1}^i)$.

The shape affinity, Λ_s^1 in Eq.(15), between the tracklet and the detection is defined as:

$$\Lambda_s^1\left(T_{t-1}^i, \mathbf{d}_t^j\right) = \exp\left(-\left\{\frac{h^i - h_d^j}{h^i + h_d^j} + \frac{w^i - w_d^j}{w^i + w_d^j j}\right\}\right) \tag{17}$$

where (w^i, h^i) and (w_d^j, h_d^j) are the widths and the heights of the bounding boxes of the tail of tracklet T_{t-1}^i and the detection $\mathbf{d}_t^j$, respectively.

The motion affinity, Λ_m^1 in Eq.(15), is evaluated between the tail of the history of the tracklet T_{t-1}^i and the detection $\mathbf{d}_t^j$ based on a linear motion assumption [14]:

$$\Lambda_m^1\left(T_{t-1}^i, \mathbf{d}_t^j\right) = \mathcal{N}\left(\tilde{\mathbf{p}}^i; \mathbf{p}_d^j, m^F\right) = \exp\left(-0.5 \cdot (\tilde{\mathbf{p}}^i - \mathbf{p}_\mathbf{d}^\mathbf{j})^\top \cdot (\mathbf{m}^\mathbf{F})^{-\mathbf{1}} \cdot (\tilde{\mathbf{p}}^\mathbf{i} - \mathbf{p}_\mathbf{d}^\mathbf{j})\right) \tag{18}$$

where $\tilde{\mathbf{p}}^i = \mathbf{p}_{tail}^i + \mathbf{v}_F^i \Theta_t$, $\mathbf{p}_{tail}^i$ and $\mathbf{p}_d^j$ represent the position of the target T_{t-1}^i and detection $\mathbf{d}_t^j$, respectively, $\mathbf{v}_F^i$ is the forward velocity of T_{t-1}^i, estimated via the associated Kalman filter (KF) using the latest N_v^F (N_v^F =4 in our experiments) states of tracklet T_{t-1}^i, and $\mathcal{N}(\cdot)$ is a Gaussian distribution function.

Then, an association score matrix S^1 is used to express the affinity score between the detections and tracklets,

$$S^1 = [s_{ij}]_{n_h \times n_d}, \quad s_{ij} = -\ln\left(\Lambda^1(T_{t-1}^i, \mathbf{d}_t^j)\right). \tag{19}$$

The Hungarian algorithm [41] is used to determine the tracklet-detection pairs with the lowest affinity value in S^1. A detection $\mathbf{d}_t^j$ is associated with T_{t-1}^i when the association cost s_{ij} is less than a pre-defined threshold θ [14].

4.1.2. Tracklet analysis and update based on prediction

Once a tracklet is associated with a detection, the state (position, velocity and size) of the object is updated with the associated detection. However, as the detection's bounding box does not always fully represent the object (see Figure 3(b),3(c),3(g)) the location, width and hight of the state vector $\mathbf{x}_t^i$ of the tracklet T_t^i is estimated using the FCT tracking results $\mathbf{c}_t^i$ and the detection $\mathbf{d}_t^j$ as follows:

$$\mathbf{x}_t^i = w_f \cdot \mathbf{d}_t^j + (1 - w_f) \cdot \mathbf{c}_t^i \tag{20}$$

with $w_f = Area(\mathcal{B}(\mathbf{d}_t^i) \cap \mathcal{B}(\mathbf{c}_t^i)) / Area(\mathcal{B}(\mathbf{d}_t^i) \cup \mathcal{B}(\mathbf{c}_t^i))$, $\mathcal{B}(\cdot)$ is the bounding box of $\mathbf{d}_t^j$ or $\mathbf{c}_t^i$, and $\cap$ and $\cup$ are the intersection and union operators between bounding boxes, respectively. The velocity $\mathbf{v}_\mathbf{t}^\mathbf{i}$ of the state vector $\mathbf{x}_t^i$ is updated using the KF output.

In our framework, the detector acts as an unbiased observation model while the FCT tracker refines the results in an adaptive way. This fusion strategy can efficiently solve inaccurate detections as shown in Figures 4(a)-4(c), especially for objects of *Motion-I* type.

For the un-matched objects (tracklets not associated to detections), the FCT-based prediction, $\mathbf{c}_t^i$, is used to analysis their occlusion state using the following constraint:

$$\zeta(\mathbf{c}_t^i, T_{\tilde{t}}^i) = \zeta_a(\mathbf{c}_t^i, T_{\tilde{t}}^i) \cdot \exp(-\zeta_p(\mathbf{c}_t^i, \mathbb{D}_t^{M_1})) \tag{21}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

with $\zeta_a(\mathbf{c}_t^i, T_{\widetilde{t}}^i)$ is the appearance similarity between the FCT-tracker prediction $\mathbf{c}_t^i$ and the templates history of the object i (tracklet T^i) at time $\widetilde{t}$ being the latest time the object i has been updated with an associated detection. It is defined as:

$$\zeta_a(\mathbf{c}_t^i, T_{\widetilde{t}}^i) = \frac{1}{N_H^a}\sum_k \rho(\chi_{\mathbf{c}^i}, \chi_{T^i}^k) \tag{22}$$

where $\chi_{\mathbf{c}^i}$ is the template of $\mathbf{c}_t^i$, $\chi_{T^i}^k$ is the k-th template of the tracklet T_t^i, and $\rho(\cdot,\cdot)$ is the Bhattacharyya distance.
And $\zeta_p(\mathbf{c}_t^i, \mathbb{D}_t^{M_1})$ is the bounding box overlap ratio between $\mathbf{c}_t^i$ and the matched detections $\mathbf{d}_t^k \in \mathbb{D}_t^{M_1}$ in the first stage. It is defined as:

$$\zeta_p(\mathbf{c}_t^i, \mathbb{D}_t^{M_1}) = \sum_{\mathbf{d}_t^k \in \mathbb{D}_t^{M_1}} \frac{Area(\mathcal{B}(\mathbf{c}_t^i) \cap \mathcal{B}(\mathbf{d}_t^k))}{Area(\mathcal{B}(\mathbf{c}_t^i) \cup \mathcal{B}(\mathbf{d}_t^k))} \tag{23}$$

$\zeta_a(\mathbf{c}_t^i, T_{\widetilde{t}}^i)$ is used to distinguish the motion-less objects from the ones occluded by obstacles, and $\zeta_p(\mathbf{c}_t^i, \mathbb{D}_t^{M_1})$ is adopted to suppress objects drift when the FCT-based prediction overlaps with a matched detection (tracklet).

In our experiments, we assume that an object is motion-less of *Motion-II* type when $\zeta(\mathbf{c}_t^i, T_{\widetilde{t}}^i) > th_o$ (th_o=0.5), otherwise, it is an occluded object ($\zeta(\mathbf{c}_t^i, T_{\widetilde{t}}^i) \leq th_o$). As shown in Figure 4(d), the motion-less object obtains reliable appearance cues, while both the appearance and motion cues are un-reliable for the occluded objects in Figure 4(e)-Figure 4(h).

After the tracklet state analysis, the FCT-based prediction $\mathbf{c}_t^i$ is used to update the state of a motion-less object (*Motion-II*). While, the state of the occluded objects (both *Occlusion-I* and *Occlusion-II*) are updated using the KF prediction. Indeed, to reduce the drifting effect of occluded object, we assume the targets do not change their motion abruptly and use KF to predict their next position.

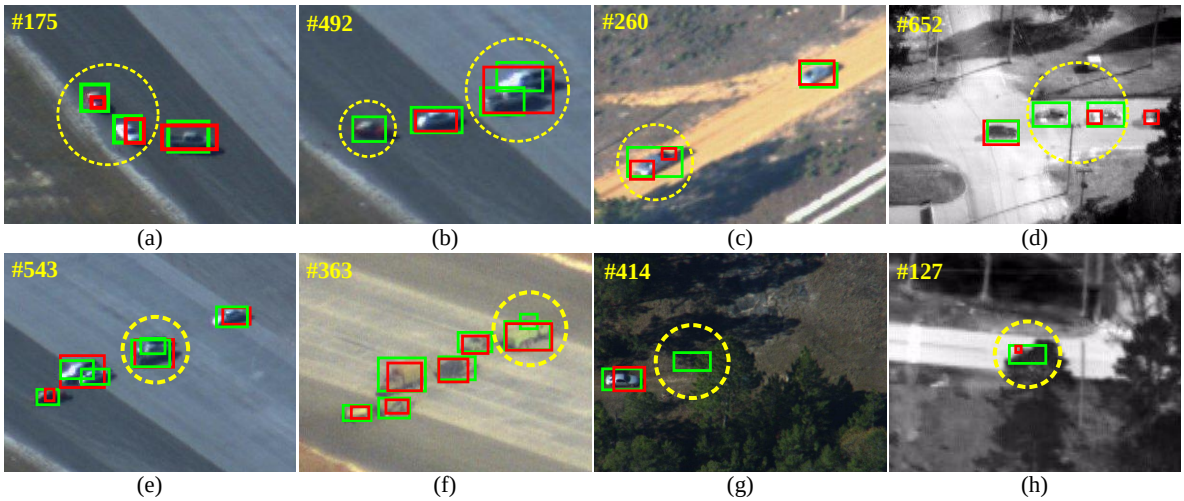


Figure 4. Illustration of Stage-1 association. The bounding boxes with the red color are the detection results. The bounding boxes with the green color are appearance-based predictions as results of the FCT tracker. The un-matched objects are marked with yellow dotted circle and yellow color. (a)-(d): matched objects having a high tracklet confidence; (e)-(h): matched objects having a low tracklet confidence.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

After the state update, the tracklet's confidence (Eq.(11)) of the matched tracklets are updated using the affinity Eq.(15), and w_p^i defined as:

$$w_p^i = \begin{cases} \zeta_a(\mathbf{c}_t^i, T_{\tilde{t}}^i), & \text{if } \zeta(\mathbf{c}_t^i, T_{\tilde{t}}^i) > th_o \\ 0.4, & \text{if } \zeta(\mathbf{c}_t^i, T_{\tilde{t}}^i) \leq th_o \end{cases} \tag{24}$$

Consequently, according to the confidence level, $\Omega(T_t^i) >= th_\Omega$, they are added to the set $\mathbb{T}_t^{A(h)}$ or $\mathbb{T}_t^{A(l)}$.

In estimating the confidence level, $w_p^i = \zeta_a(\mathbf{c}_t^i, T_{\tilde{t}}^i)$ is used to slowly reduce the tracklet confidence of the motion-less objects according to the appearance similarity, and $w_p^i = 0.4$ is used to reduce the value of the tracklet confidence of the occluded objects to change them to unreliable tracklets, $\mathbb{T}_t^{A(l)}$, for input to Stage-2 for occlusion analysis.

4.1.3. Detection refinement

In Figure 3, we illustrated some inaccurate detections caused by two (or more) spatially close objects, which might increase the object's identity switch and false alarms. Therefore, we propose a detection refinement process to solve these problems. For the un-matched detection $\mathbf{d}_t^j \in \mathbb{D}_t^{U_1}$ after Stage-1, we delete inaccurate detections from $\mathbb{D}_t^{U_1}$ when their bounding box is overlapping with more than 2 un-matched objects updated by the FCT appearance-based prediction. Thus, the inaccurate detections in Figure 3(b),(c),(e),(f),(g) are deleted if they do not associate to any tracklets. After this step of detection refinement, all remaining un-matched detection $\mathbf{d}_t^j \in \mathbb{D}_t^{U_1}$ are used in Stage-2, along with the un-reliable tracklets in $\mathbb{T}_t^{A(l)}$.

4.2. Stage-2: Handling drifting tracklets

In complex situations of airborne videos, where objects are occluded while the mounted camera changes its motion, conventional on-line tracking methods, based on a simplified motion model (*e.g.*, the used KF-based constant velocity model), are prone to produce drifting problems [25,44]. If the object continue drifting, it is difficult to re-assign it to detections or re-appearing objects (*Occlusion-I* and *Occlusion-II*). In the proposed framework, the second association stage solves the reassignment problem between un-reliable tracklets $\mathbb{T}_t^{A(l)}$ and un-matched detections $\mathbb{D}_t^{U_1}$ not associated during the first stage. An un-reliable tracklet in $\mathbb{T}_t^{A(l)}$ is converted into a reliable tracklet in $\mathbb{T}_t^{A(h)}$ if it can be re-associated with a detection, otherwise, it maintains the same state or converted to an inactive tracklet in $\mathbb{T}_t^{I_o(l)}$ after state update.

There are two aspects which are considered in this stage: (1) if the object is occluded by an occluder, it might re-appear again around the occluder. The un-matched detection near the occluder has the high possibility to be re-associated to the re-appeared object after occlusion; (2) if the object has been occluded by environmental obstacles, it might re-appear at any position in the image. We assume that the occluded object might re-appear in a limited region around the occluder. The more time it disappears, the larger search region should be.

4.2.1. Second association via the affinity score

For the current frame t, the input pairs of this association stage are $\{(T_t^i, \mathbf{d}_t^j) | \forall T_t^i \in \mathbb{T}_t^{A(l)}, \forall \mathbf{d}_t^j \in \mathbb{D}_t^{U_1}\}$. The affinity of the second association is defined as:

$$\Lambda^2\left(T_t^i, \mathbf{d}_t^j\right) = \begin{cases} \Lambda_a^1(T_t^i, \mathbf{d}_t^j) \cdot \exp\left(\Omega(T_t^k)\right), & \text{if } \zeta_s^2(T_t^i) = T_t^k, dist(\mathbf{d}_t^j, T_t^k) \leq \Delta_t^{i(l)} \\ \Lambda_a^1(T_t^i, d_t^j), & \text{if } \zeta_s^2(T_t^i) = \emptyset, dist(\mathbf{d}_t^j, T_t^i) \leq \Delta_t^{i(h)} \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

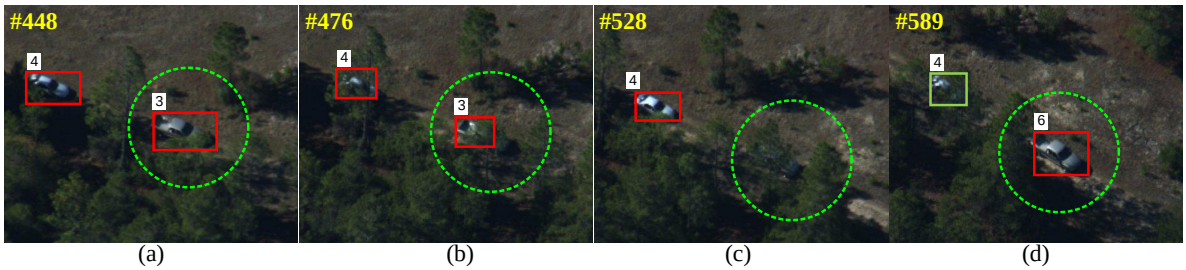
In the above equation, $\zeta_s^2(T_t^i)$ is an operator which returns a possible occluder tracklet T_t^k or $\emptyset$ to indicate that the occluder is an environmental obstacle. A tracklet T_t^k is defined as an occluder of T_t^i if the overlap ratio, $\zeta_p(\mathbf{c}_t^i, T_t^k)$ (defined in Eq.(23)), between the bounding box of the FCT-based tracker $\mathbf{c}_t^i$ of T_t^i and the bounding box of the tracklet T_t^k is less than a given overlapping threshold th_o, i.e. $\zeta_p(\mathbf{c}_t^i, T_t^k) \geq th_o$. The function $dist(\mathbf{d}_t^j, T_t^k)$ is the Euclidean distance between the location of a detection $\mathbf{d}_t^j$ and the tracklet T_t^k. $\Delta_t^{i(l)} = \sqrt{(\frac{w_t^i + w_t^k}{2})^2 + (\frac{h_t^i + h_t^k}{2})^2}$ is the maximum allowed distance for an acceptable detection near the occluder tracklet, T_t^k to be associated to T_t^i, with (w_t^i, h_t^i) and (w_t^k, h_t^k) the width and height of the bounding box of tracklets T_t^i and T_t^k, respectively. $\Delta_t^{i(h)} = \sqrt{(w_t^i)^2 + (h_t^i)^2} \cdot L_M \cdot (1 - \Omega(T_t^i))$ is the maximum allowed distance of an acceptable detection to be associated to be associated to T_t^i, where $\Omega(\dot{})$ is the tracklet confidence, and L_M is the number of frames in which the i-th object is missing due to occlusion or un-reliable detection (defined in Eq. (11)).

4.2.2. Tracklet correction

The second association allows us to re-assign drifting tracklets to the detections of re-appearing objects in a limited time. An association score matrix S^2, same as in Eq.(19), is used to express the affinity score between the detections and the tracklets, and the Hungarian algorithm [41] is used to determine the tracklet-detection pairs with the lowest affinity value in S^2. After association, the state and the confidence values of the associated tracklets are updated with the associated detections using Eq.(20) and Eq.(11), respectively. Here, To update the state of the re-appeared tracklet we only use the matched detection and set $w_f = 1$ in Eq.(20). Finally, the trajectory within the drifting interval is corrected via a linear interpolation between the previous location of the tracklet and the updated one.

4.3. Stage-3: New active tracklet generation

The third association stage solves the assignment problem between the candidate tracklets $\mathbb{T}_{t-1}^C$ from the previous frame and the remaining un-matched detections $\mathbb{D}_t^{U_2}$ to generate new active tracklets $\mathbb{T}_t^{A_n(h)}$. The input pairs of this association in the current frame t are $\{(T_{t-1}^i, \mathbf{d}_t^j) | \forall T_{t-1}^i \in \mathbb{T}_{t-1}^C, \forall \mathbf{d}_t^j \in \mathbb{D}_t^{U_2}\}$. The affinity, $\Lambda^3(T_{t-1}^i, \mathbf{d}_t^j)$, and the association score matrix, S^3, are the same as the ones used in the first stage (Stage-1). When the candidate tracklet is associated in th_I consecutive frames (th_I=5 frames in our experiments), it is converted into a new tracklet, for which we initialize an FCT appearance-based tracker. The matched to detection candidate tracklets are maintained in the candidate tracklet set $\mathbb{T}_t^C$ if the tracklet length is less than th_I, while the un-matched candidate tracklets, which are considered as false-alarms, are removed from the candidate tracklet set.



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

Figure 5. Fragmented tracklet under long-term occlusions. (a) Two tracked objects ID-3 and ID-4; (b) the object ID-3 is partially occluded, and (c) heavily occluded by trees; (d) the lost object ID-3 is switched to ID-6 when it reappears again after the occlusion.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

4.4. Stage 4: Globally linking fragmented tracklets

In challenging situations, where the objects are constantly occluded by other objects or obstacles for a long-time, tracklet fragmentation is likely to occur and the same object can be divided into two or more tracklets, as illustrated in Figure 5. Motivated by the works in object re-identification [38,39] to build long-term object trajectories based on appearance modeling and matching, the fourth association stage of the proposed framework solves the assignment problem between the lost tracklets $\mathbb{T}_t^{I_o(l)}$ and the new tracklets $\mathbb{T}_t^{A_n(h)}$ to link these fragmented tracklets, re-identify the lost objects and thereby building longer trajectories. Due to the fact that targets in airborne videos have very similar appearance, false tracklet linking might occur if only based on the appearance modeling. Thus, both the appearance and motion terms are considered in the fourth stage.

4.4.1. Fourth association via the affinity score

The input pairs of the forth association in the current frame t is the set $\{(T_t^i, T_t^j)|\forall T_t^i \in \mathbb{T}_t^{I_o(l)}, \forall T_t^j \in \mathbb{T}_t^{A_n(h)}\}$. The affinity of the fourth association is defined as,

$$\Lambda^4\left(T_t^i, T_t^j\right) = \Lambda_a^4\left(T_t^i, T_t^j\right)\Lambda_m^4\left(T^i, T^j\right) \tag{26}$$

where $\Lambda_a^4\left(T_t^i, T_t^j\right)$ and $\Lambda_m^4\left(T_t^i, T_t^j\right)$ are the appearance and motion affinity score, respectively.

The appearance affinity $\Lambda_a^4\left(T_t^i, T_t^j\right)$ is defined as:

$$\Lambda_a^4\left(T_t^i, T_t^j\right) = \max\left\{\frac{1}{N_H^i}\sum_{l\in[1,N_H^i]}\varsigma(\chi_{T^i}^l, T_t^j), \frac{1}{N_H^j}\sum_{m\in[1,N_H^j]}\varsigma(\chi_{T^j}^m, T_t^i)\right\} \tag{27}$$

where N_H^i and N_H^j are the number of templates of the tracklet T_t^i and T_t^j, respectively. $\chi_{T^i}^l$ is the l-th template of tracklet T_t^i, $\chi_{T^j}^m$ is the m-th template of tracklet T_t^j. And $\varsigma(\chi_{T^i}^a, T_t^b) = \frac{1}{N_H^b}\sum_{b\in[1,N_H^b]}\rho(\chi_{T^i}^a, \chi_{T^j}^b)$, for $(a,b) = (l,j)$ and $(a,b) = (m,i)$. The motion affinity $\Lambda_m^4\left(T_t^i, T_t^j\right)$ is evaluated between the tail of the history of the trackelt T_t^i and the head of the tracklet T_t^j with the time gap Θ_t [14] based on a linear motion assumption:

$$\Lambda_m^4\left(T^i, T^j\right) = \mathcal{N}\left(\tilde{\mathbf{p}}_i; \mathbf{p}_j^{head}, m^F\right)\mathcal{N}\left(\tilde{\mathbf{p}}_j; \mathbf{p}_i^{tail}, m^B\right) \tag{28}$$

where $\tilde{\mathbf{p}}_i = \mathbf{p}_i^{tail} + \mathbf{v}_i^F\Theta_t$ and $\tilde{\mathbf{p}}_j = \mathbf{p}_j^{head} + \mathbf{v}_j^B\Theta_t$, $\mathbf{p}_i^{tail}$ and $\mathbf{p}_j^{head}$ represent the position of T_t^i and T_t^j, $\mathbf{v}_i^F$ is the forward velocity of T_t^i and $\mathbf{v}_j^B$ is the backward velocity of T_t^j estimated using the KF with the latest and first N_v^B states of the tracklet T_t^i and T_t^j, respectively. $\mathcal{N}(\cdot)$ is a Gaussian distribution function.

4.4.2. Object re-identification via tracklet Linking

The association score matrix $S^4 = [s_{ij}]_{n_i^4\times n_j^4}$ with $s_{ij} = -\ln\left(\Lambda^4(T_t^i, T_t^j)\right)$ is used to express the affinity score between tracklets in the fourth stage. The Hungarian algorithm [41] is used to determine the (i,j) pairs of tracklets with the maximum affinity in S^4. The tracklet T_t^j is associated with T_t^i when the association cost s_{ij} is less than a pre-defined threshold θ [14]. If a lost tracklet T_t^i and a new tracklet T_t^j are associated, they are considered as the same object and merged and their trajectory are linked with an linear interpolation. We assign to the new tracklet T_t^j the ID of lost one T_t^i. Thus, the lost objects are re-identified using the above described tracklet linking process.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

The remaining inactive tracklets which have not been reassigned to new tracklets are either terminated if $t - t_e^i \geq th_e$, or kept in the inactive tracklets set $\mathbb{T}_t^{I_o(l)}$.

5. Experiments

5.1. Datasets

We evaluated our approach on two datasets, the VIVID dataset [45] and the SAIIP dataset. Figure 6 illsutrates few images from the used datasets. The first dataset includes 5 visible data sequences and 3 thermal IR data sequences. The second dataset includes 4 sequences which are captured using an UVA belonging to the Northwestern Polytechnical University. Table 1 lists the different sequences along with their main challenging situations, including Illumination variation (IV), Scale Variation (SV), Occlusion (OCC), Background Occlusion (BOC), Motion Variation (MV), Image Blurring (IB) and Shadow Interference (SI).

In the *EgTest01* sequence, the vehicles loop around a runway and then drive straight. Some vehicles have very similar appearance. In the *EgTest02* sequence, two sets of three vehicles pass by each other on a runway. Changes of scaling occur because the airborne camera circles the scene. The data association for the *EgTest02* sequence is more difficult than for the *EgTest01* sequence due to severe occlusions. This also happens in the *EgTest03* sequence, where two sets of three vehicles pass by each other on a runway. In the *EgTest04* sequence, a line of vehicles travels down a red dirt road. In the *EgTest05* sequences, a vehicle is moving along a dirt road in a wooded area. Occlusion and illumination variations occur when the vehicle passes in and out of tree shadows.

The sequences of *PkTest01*, *PkTest02* and *PkTest03* are thermal IR data. In the *PkTest01* sequence, the vehicles are frequently occluded by the trees. In the *PkTest02* sequence, the vehicles stop at an intersection then continue. The main issues are: occlusion, the shadows and camera auto-gain problems. The thermal IR contains a line of vehicles in a stop and go scenario in the *PkTest03* sequence. As in the previous sequence, occlusions, shadows and camera auto-gain are prevalent in this sequence. Moreover, the vehicles are small, and the camera viewpoint is nearly nadir.

All the sequences from the SAIIP dataset (*SpTest01*, *SpTest02*, *SpTest03* and *SpTest04*) are captured over a provincial road. There are less occlusions because the camera pointed at the road to take the videos and most of the vehicles are moving with a high speed while keeping a safety distance between each other. However, several targets have very similar appearance, and some of them are stopping at the crossroad. There are also some auto-trucks with a long body size which might be detected as two separated objects.

Table 1. Used benchmark sequences.

Sequence	Image Size	Dataset	IV	SV	OCC	BOC	MV	IB	SI
EgTest01	640 × 480	VIVID	√	√	×	×	√	×	√
EgTest02	640 × 480	VIVID	√	√	√	×	√	×	√
EgTest03	640 × 480	VIVID	√	√	√	×	√	×	√
EgTest04	640 × 480	VIVID	√	×	×	√	√	√	√
EgTest05	640 × 480	VIVID	√	√	×	√	√	×	√
PkTest01	320 × 256	VIVID	√	√	√	√	×	×	×
PkTest02	320 × 256	VIVID	√	√	×	√	√	×	×
PkTest03	320 × 256	VIVID	√	×	×	√	√	×	×
SpTest01	1920 × 1080	SAIIP	√	×	×	×	×	×	×
SpTest02	1920 × 1080	SAIIP	√	√	×	×	√	×	×
SpTest03	1920 × 1080	SAIIP	√	√	×	×	√	×	√
SpTest04	1920 × 1080	SAIIP	√	√	×	√	√	×	√

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

5.2. Parameters Setting

The proposed MOT framework has been implemented on a PC with an Intel Core 2.40 GHz CPU with 32GB RAM. In the following we describe the parameters setting of each module of the framework.

5.2.1. Parameters of the detector

We first compare three recently used motion compensation-based detectors, and then analysis the parameters setting of the used detector. The three compensation-based detectors include the basic compensation-based detector (BCD) [20], the MHI detector [20] and the SGM detector [22]. All the source codes have been provided by the authors. For a fair comparison, the same parameter setting used by the authors in their original publication are consisted. Both BCD and MHI detectors assume a pre-defined threshold, T_θ=20, to determine the detections in each image. The SGM detector relies on the parameter of the grid size $T_\theta \times T_\theta$ with T_θ=10 [22] for determining the detections.

For the quantitative evaluation of the detectors performance, we use the detection ratio (DTR) $r_D = N_O^D / N_O^T$ and the false-alarm ratio (FAR) $r_F = (N_O^A - N_O^T)/N_O^A$, where N_O^D represents the effective number of the detect object, N_O^T represents the number of the true objects, and N_O^A represents the total number of detections. A detection with bounding box B_D is considered successful, if $SR = \frac{Area(B_D \cap B_{GT})}{Area(B_D \cup B_{GT})} \geq T_{SR}$ (in our experiments $T_{SR} = 0.5$) for a ground truth bounding box B_{GT}. To well analyze the influence of the threshold, T_θ, of the considered motion compensation-based detectors, we define different values of $T_\theta^v = 10 \cdot \theta_v$, with $\theta_v = \{0.5, 0.75, 1, 1.25, 1.5\}$. As shown in Figure 7, the MHI-based apparoach can efficiently reduce FAR compared to the BCD- and SGM-based approaches. However, the required forward motion history is not suitable for practical applications. In our implementation, we select the SGM-based detector which has comparable DTR and FAR to the MHI-based approach while it performs in a real-time.

using motion-based compensation approaches, the detection performance depends on the velocity of the tracked objects and the complexity of the background. As such, a single fixed determining threshold T_θ is not suitable for all the test sequences. Table 2 lists the DTR ratio and FAR ratio, along

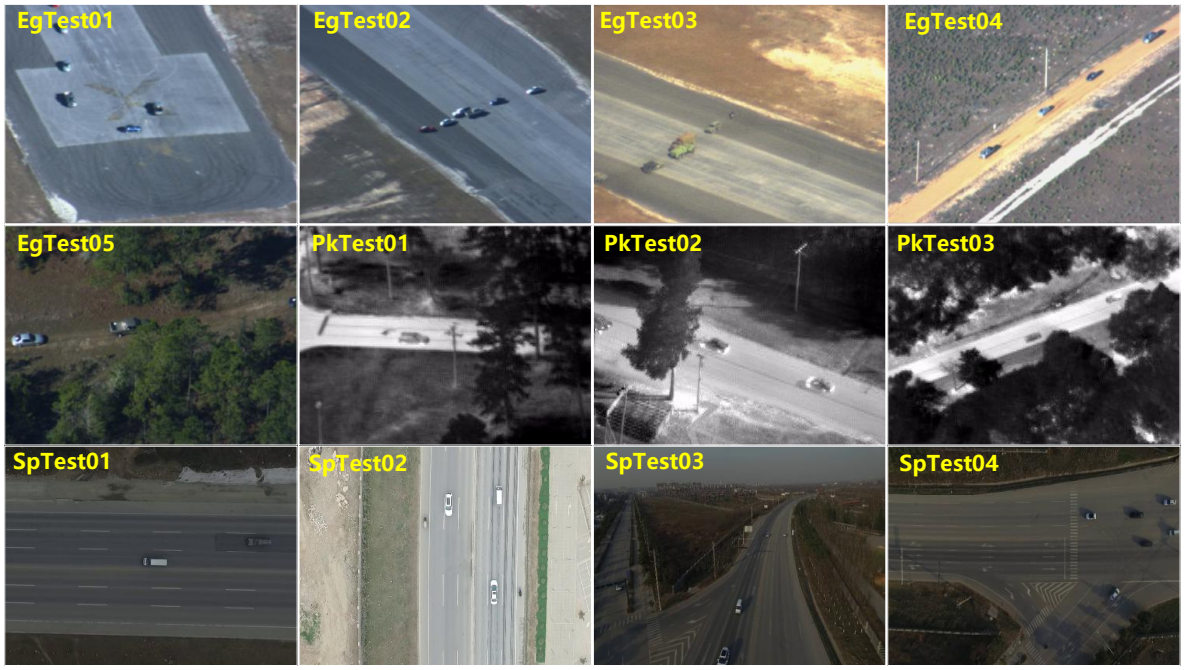


Figure 6. Scenes from the public DARPA VIVID dataset (first two rows) and the SAIIP dataset (last row).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

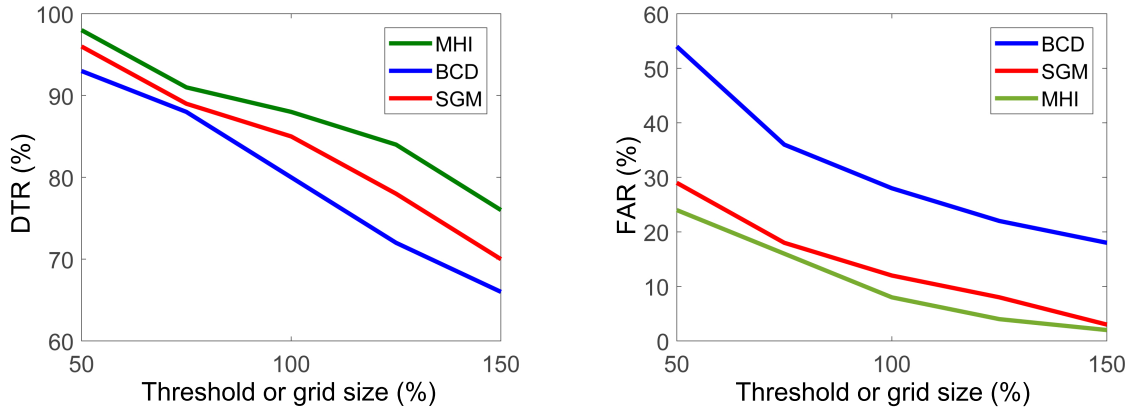


Figure 7. Performance comparison of different motion compensation-based detectors.

Table 2. Comparison of detection results with different detection thresholds T_θ^v.

Threshold	VIVID			SAIIP		
	DTR%	FAR%	FPS	DTR%	FAR%	FPS
T_θ^1	91.7	36.7	18	97.3	12.8	9
T_θ^2	85.6	28.4	22	94.4	10.3	12
T_θ^3	81.3	18.6	28	91.7	8.7	16
T_θ^4	72.9	14.2	32	88.5	6.6	20
T_θ^5	68.4	10.5	37	86.9	5.9	27

with the computational cost in terms of frames per second (FPS), of the SGM-based detector with different determining thresholds on the VIVID dataset and SAIIP dataset, respectivley. As it can be noticed, on the VIVID dataset, with increasing values of the determining threshold, both the DTR and FAR ratios reduce. The obtained results on the SAIIP dataset are very similar while with less computation when the determining threshold is increased. The computation cost on the SAIIP dataset is higher than on the VIVID dataset due to the larger image size.

For the experiments reported in the following sections, we set $w^d = 0.5$ in Eq. (11) and $T_\theta = 10$ for the 5 visible data sequences and $T_\theta = 5$ 3 thermal IR data sequences of the VIVID dataset. For the and the SAIIP dataset we set $T_\theta = 15$ and $w^d = 0.7$. Note that, w^d is set to a large value when the detector provides high accuracy [14].

5.2.2. Parameters of hierarchical framework

All parameters of the tracking framework have been set empirically, and remained unchanged for all datasets.

- For the affinity models of Eq. (15) and Eq. (26), the parameters m^F and m^B are set to diag $[30^2\ 75^2]$.
- The same threshold $\theta = 0.4$ is used for the association score matrices S^1, S^2, S^3 and S^4 to determine the association results.
- For the FCT trackers, in our experiments, the search radius for drawing positive samples in the on-line appearance-based classifier is set to $\alpha = 4$, to generate 45 positive samples. The inner and outer radius for the negative samples are set to $\beta = 8$ and $\zeta = 30$, respectively, to randomly select 50 negative samples. The initial learning rate λ of the classifier is set to 0.9. The size of the random matrix is set to 100.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

Table 3. Evaluation metrics [46].

Name	Definition
PR	Correctly matched objects / total output objects. (Frame-based)
FA/Frm	Number of false alarms per frame. *The smaller the better*. (Frame-based)
GT	Number of groundtruth trajectories.
MT	Mostly tracked: Percentage of GT trajectories which are covered by the tracker's output for more than 80% in length.
ML	Mostly lost: Percentage of GT trajectories which are covered by the tracker's output for less than 20% in length. *The smaller the better*.
PT	Partially tracked: 1.0-MT-ML.
Frag	Fragments: The total number of times that a groundtruth trajectory is interrupted during tracking. *The smaller the better*.
IDS	ID switches: The total of number of times that a tracked trajectory changes its matched GT identity. *The smaller the better*.

Table 4. Comparison of tracking results on sequence *EgTest02* with different detection thresholds T_θ^v ($\theta_1 = 0.5, \theta_3 = 1, \theta_5 = 1.5$).

Method	MT(%)			ML(%)			IDS		
	T_θ^1	T_θ^3	T_θ^5	T_θ^1	T_θ^3	T_θ^5	T_θ^1	T_θ^3	T_θ^5
$\mathbf{S}_1$	86.6	80.6	76.3	3.8	8.6	16.4	24	20	27
$\mathbf{S}_2$	**92.1**	**86.1**	**80.5**	**2.1**	**6.8**	**10.7**	**12**	**9**	**13**

- For the Kalman Filter model, the process (Q) and measurement (R) noise covariance matrices are set as $Q = \begin{bmatrix} 0.0025 & 0 & 0.0025 & 0 \\ 0 & 0.0025 & 0 & 0.0025 \\ 0.0025 & 0 & 0.0025 & 0 \\ 0 & 0.0025 & 0 & 0.0025 \end{bmatrix}$, and $R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$, respectively.

5.3. Comparison to state-of-art frameworks

To demonstrate the tracking performance of our proposed framework, we compared it to the MOT approaches of [14] and [10] on the selected datasets. All the approaches, including ours, adopt the same detection configuration, and a window size of 5 frames is defined to remove unreliable shorter tracklet. For both [14] and [10], we used the publicly available codes provided by the authors.

5.3.1. Evaluation metrics

The popular evaluation metrics defined in [46], as listed in Table 3, are used for evaluating the performance. The precision of the intersection area over the union area of bounding boxes (PR), the number of the trajectories in the ground-truth (GT), the ratio of the mostly tracked trajectories (MT), the ratio of the mostly lost trajectories (ML), the ratio of partially tracked trajectories (PT), and identity switches (IDS).

5.3.2. Comparison of data association

The qualitative comparison between different versions of the proposed system on sequence *EgTest02* is given in Table 4. The two considered versions, $\mathbf{S}_1$ and $\mathbf{S}_2$ are defined as follows:

- $\mathbf{S}_1$: correspond to the framework without tracklets analysis and detection refinement. It corresponds to the approach of [14]. It adopts the method of [14] to estimate the tracklet state. The position and the velocity of the matched tracklets are updated with the associated detection while the un-matched tracklets are updated using the KF motion-based predictions. The size of the object is updated by averaging the associated detection of the recent past frames.
- $\mathbf{S}_2$: full proposed HATA framework as illustrated in Figure 1.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

Comparing the results of the frameworks $\mathbf{S}_1$ and $\mathbf{S}_2$, one can notice the effect of the tracklet analysis and detection refinement processes in the proposed framework $\mathbf{S}_2$. Notice from Table 4, the system $\mathbf{S}_1$ performs well for the MT and ML measures, while, the high false alarm rate and unreliable detections introduce high IDS measure, due to inaccurate location and size of the detections, which affects the association between tracklets and detections. As expected, the proposed framework $\mathbf{S}_2$ improves the performance for most metrics, and it efficiently reduces the IDS measure compared to $\mathbf{S}_1$. Figure 8 illustrates the tracking results of $\mathbf{S}_1$ and $\mathbf{S}_2$ using the threshold T_θ^3 on sequence *EgTest02*. As shown in Figure 8, the targets with ID-2 and ID-3 in frame #390 have an accurate location and size using the framework $\mathbf{S}_2$ even with inaccurate detections input. This is due to the use of the FCT tracker to correct the state of the tracklet (see Eq. (20)). Similarly, $\mathbf{S}_2$ performs well in frame #460 with the help of the tracklet analysis and detection refinement process, which efficiently avoid the false new tracklet generation (ID-11 in system $\mathbf{S}_1$). This also happens in frame #532.

5.3.3. Comparisons to other MOT frameworks

A quantitative comparison between our proposed framework and state-of-the-art algorithms is given in Table 5. Both [14] and [10] achieve good results with the available detections, while with poor performance under inaccurate detection. Instead, our algorithm improves the performance for the used evaluation metrics (ML, MT and IDS). The qualitative tracking results of our approach are shown in Figure 9 and Figure 10.

Results using the VIVID dataset: Figure 9 illustrates the tracking results using the 8 sequences from the VIVID dataset. For the *EgTest01* sequence, all considered approaches perform well because of the reliable detections. Our proposed framework achieves the best results when the appearance and the motion of the vehicles vary during the loop around period (frame #28, #172 and #323). In the

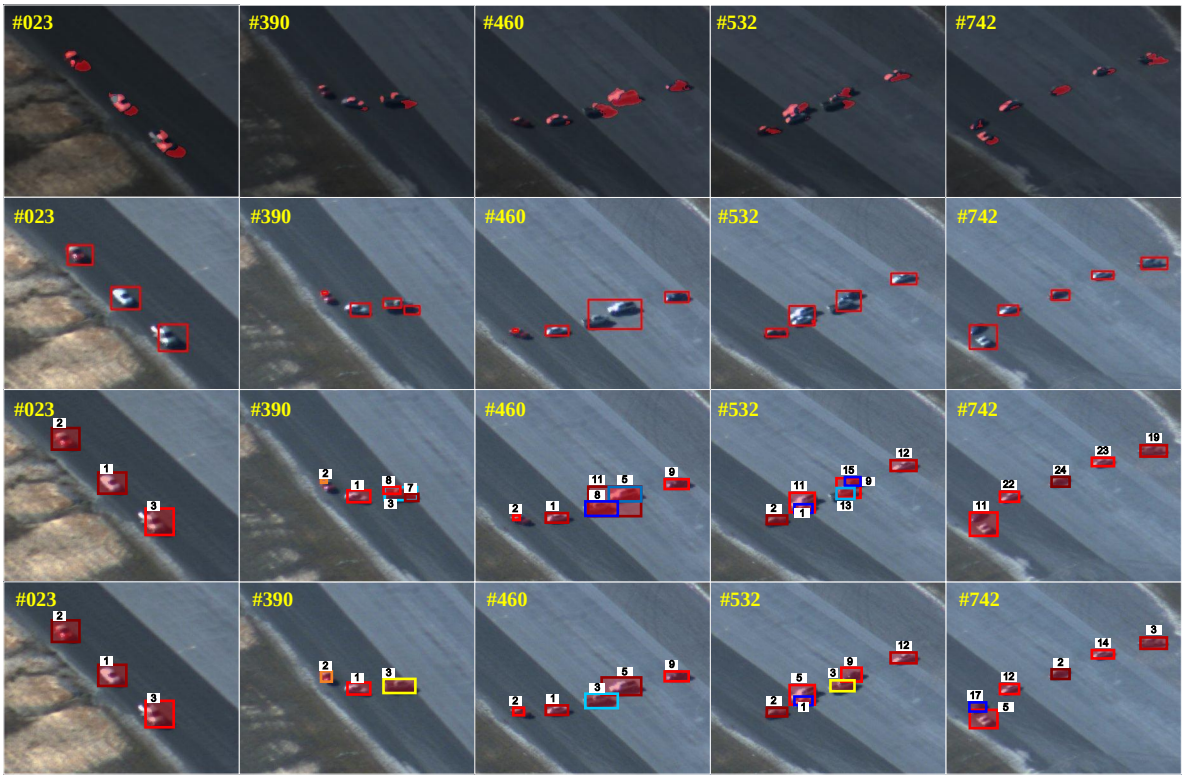


Figure 8. Detection and tracking results. First row: the detection results. Second row: the bounding box for each detection. Third row: the tracking results using the framework $\mathbf{S}_1$. Fourth row: the tracking results using the framework $\mathbf{S}_2$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

EgTest02 sequence, two sets of vehicles pass by each other on a runway and one set is occluded by the other set between frame #443, #482 and #670. Both [14] and [10] produce ID switches with most of the tracked targets, while HATA identifies well most of the tracklets. HATA also performs well in the *EgTest03* sequence. In the *EgTest04* sequence, only HATA solves the ID switches problem when the vehicle with ID-3 has been occluded by the trees in frame #721. In the *EgTest05* sequence, HATA handles well the occlusion in frame #590 and #701 and the illumination changes when the targets pass in and out of the shadowed wooded area.

Figure 9(f)-(g) illustrate the tracking results using the thermal IR sequences *PkTest01*, *PkTest02* and *PkTest03*. In the *PkTest01* sequence, only HATA identifies well the vehicle which is frequently occluded by the tress between frames #128 and #278. Our algorithm constantly keeps on tracking the vehicles which stop at the intersection in frame #561 and restart moving after frame #654 in the *PkTest02* sequence. Like for visible data, HATA solves the occlusion and illumination variation problems in IR data, as shown in frames #833 and #1229. In the *PkTest03* sequence, the vehicles are frequently occluded by trees after frame #298, and HATA can robustly save the correct ID for each tracked target in frame #374 and frame #386.

Table 5. Tracking results on the selected datasets. The best performing method is shown in **Bold**.

Sequence	GT	Method	PR(%)	MT(%)	ML(%)	PT(%)	IDS
EgTest01	6	Bae *et al.* [14]	90.7	94.4	3.6	2.0	**2**
		Prokaj *et al.* [10]	88.6	93.6	3.2	3.2	4
		Proposed HATA	**94.8**	**96.8**	**2.9**	**0.3**	**2**
EgTest02	6	Bae *et al.* [14]	78.8	80.6	8.6	11.8	28
		Prokaj *et al.* [10]	70.5	69.3	5.4	25.3	41
		Proposed HATA	**84.4**	**86.1**	**6.8**	**7.1**	**13**
EgTest03	6	Bae *et al.* [14]	82.6	80.7	6.8	12.5	20
		Prokaj *et al.* [10]	77.8	74.3	5.4	20.3	29
		Proposed HATA	**87.1**	**83.6**	**4.7**	**11.7**	**11**
EgTest04	5	Bae *et al.* [14]	82.9	78.9	4.9	16.2	19
		Prokaj *et al.* [10]	76.4	73.2	6.6	20.2	28
		Proposed HATA	**85.3**	**81.8**	**5.6**	**12.6**	**12**
EgTest05	4	Bae *et al.* [14]	68.9	75.2	6.7	18.1	42
		Prokaj *et al.* [10]	70.8	81.2	5.3	13.5	60
		Proposed HATA	**78.6**	**86.4**	**5.7**	**7.9**	**23**
PkTest01	5	Bae *et al.* [14]	79.6	82.3	5.3	12.4	20
		Prokaj *et al.* [10]	74.3	78.7	10.2	11.1	36
		Proposed HATA	**88.8**	**89.1**	**2.1**	**8.8**	**14**
PkTest02	12	Bae *et al.* [14]	76.9	73.8	5.9	20.3	23
		Prokaj *et al.* [10]	72.9	69.7	7.2	23.1	38
		Proposed HATA	**83.4**	**79.4**	**5.1**	**15.5**	**15**
PkTest03	7	Bae *et al.* [14]	72.9	78.6	6.4	15.0	29
		Prokaj *et al.* [10]	68.4	74.5	8.2	17.3	42
		Proposed HATA	**79.1**	**81.9**	**5.8**	**12.3**	**16**
SpTest01	37	Bae *et al.* [14]	97.6	94.7	0.9	5.4	5
		Prokaj *et al.* [10]	93.3	92.6	2.8	7.6	9
		Proposed HATA	**98.5**	**96.4**	**0.5**	**3.1**	**2**
SpTest02	42	Bae *et al.* [14]	88.9	83.8	9.8	6.4	18
		Prokaj *et al.* [10]	82.9	77.9	12.2	9.9	22
		Proposed HATA	**93.5**	**91.4**	**6.2**	**3.4**	**7**
SpTest03	29	Bae *et al.* [14]	87.2	85.6	10.8	3.6	17
		Prokaj *et al.* [10]	84.6	82.6	13.5	3.9	29
		Proposed HATA	**89.8**	**91.2**	**6.9**	**1.9**	**11**
SpTest04	46	Bae *et al.* [14]	89.3	87.9	8.4	3.7	26
		Prokaj *et al.* [10]	81.6	81.3	13.6	5.1	31
		Proposed HATA	**91.7**	**93.4**	**4.1**	**2.5**	**12**

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

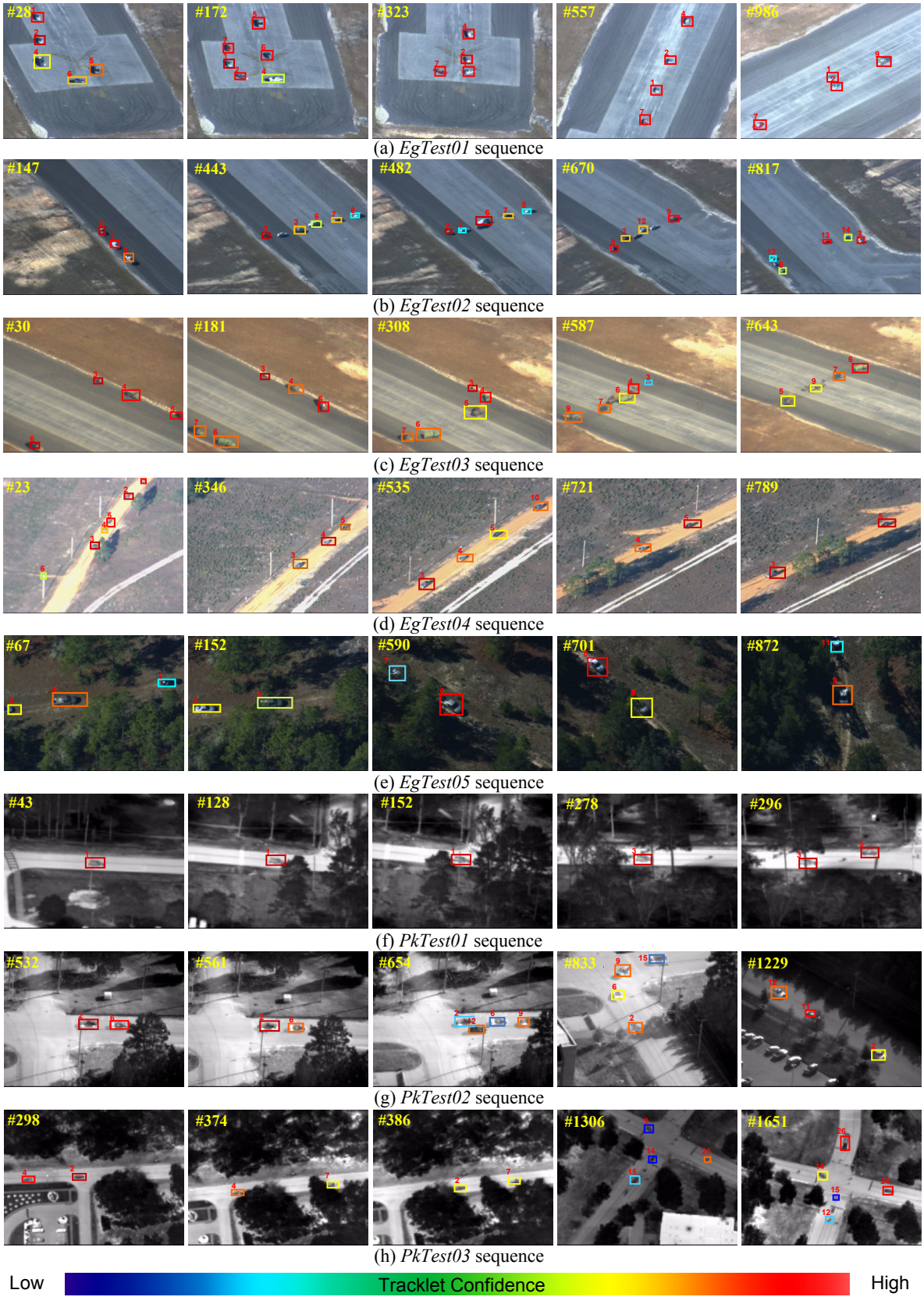


Figure 9. The results on 8 sequences from the VIVID dataset.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

Results using the SAIIP dataset: Figure 10 illustrates the tracking results using the SAIIP dataset. For the *SpTest01* sequence, all the moving objects are well detected (Figure 10(a)). HATA efficiently tracks all the detected objects. The false alarms are removed when the bounding boxes size are smaller than a pre-defined threshold $T_{fal} = 5 \times 5$. This strategy is also adopted for the sequences *SpTest02*, *SpTest03* and *SpTest04*. The *SpTest02* sequence is more challenging than *SpTest01* sequence when the vehicles slowdown their motion. HATA solves the motion-less problem, as shown in frames #564 and #709 of Figure 10(b). Both *SpTest03* sequence and *SpTest04* sequence are captured around a crossroad where the vehicles are slowing down, stopping or changing directions. In the *SpTest03* sequence, as shown in Figure 10(c), HATA identifies well the object with ID-4 when it changes direction in frame #122. Moreover, HATA achieves long-term trajectory for the object with ID-1 in frame #245. In the *SpTest04* sequence, a lot of vehicles are passing through the crossroad. As shown in Figure 10(d), HATA identifies well the objects with ID-3 and ID-7 in frame #98, and the objects with ID-3 and ID-10 in frame #119.

The proposed method was implemented using MATLAB on a PC with an Intel Core 2.40 GHz CPU with 32GB RAM without parallel and GPU processing. The average speed of the proposed method using the VIVID dataset is about 18 FPS and 15 FPS for SAIIP dataset, excluding the detection step. The results show the improved performance of the proposed method, compared to state-of-the-art methods. However, it is worth noting the limitations of our proposed algorithm. Generally, the

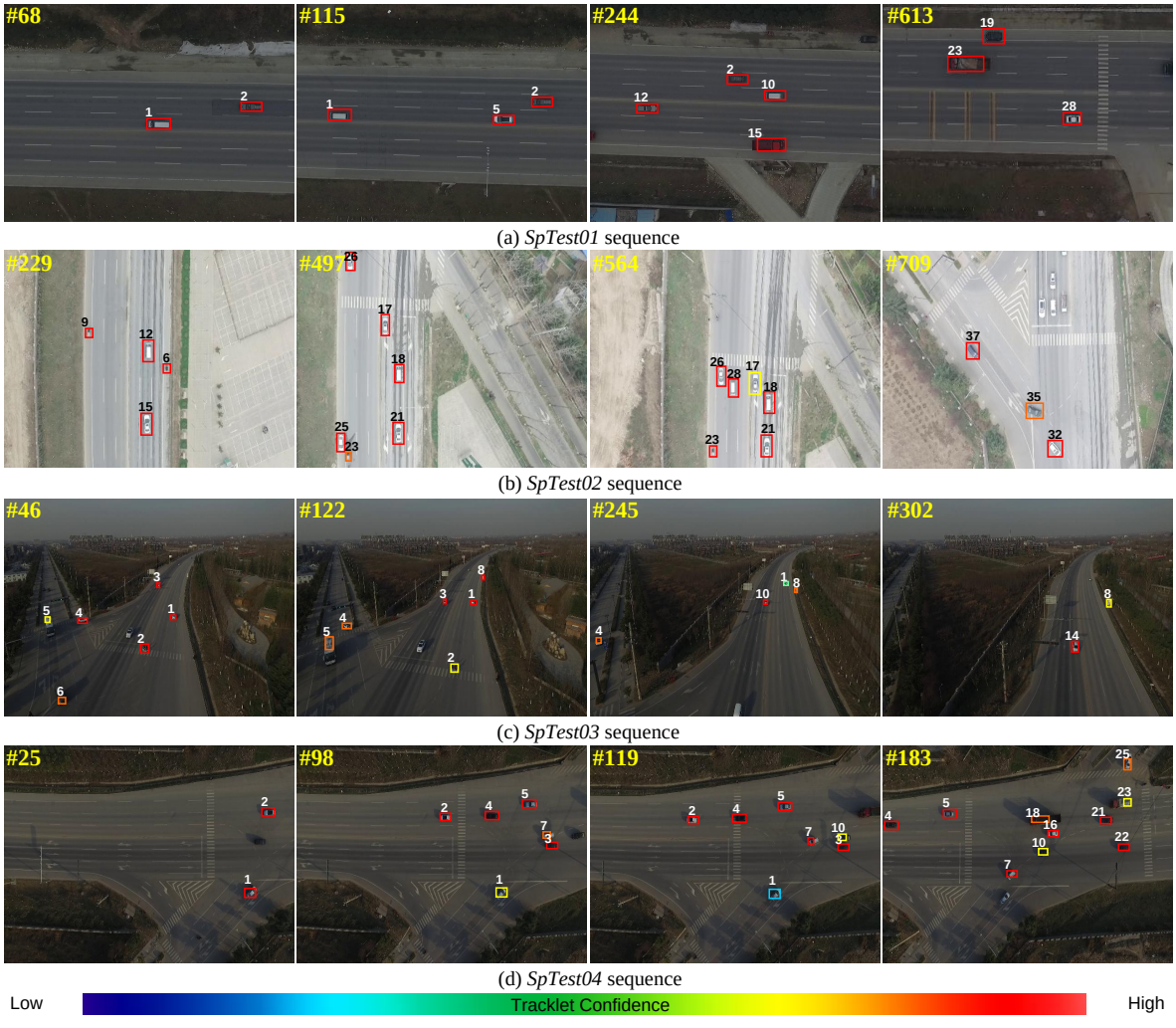


Figure 10. The results on 4 sequences from the SAIIP dataset.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

limitation comes from three aspects: (i) the unreliable object initialization caused by motion-less objects occluded objects; (ii) reduced performance of the FCT-based tracker when objects changes abruptly their appearance which causes unreliable tracklet state analysis, and hence unreliable matching; (iii) the fixed parameters of the detector, which are not suitable for other type of datasets.

6. Conclusions

In this paper, an on-line multi-object tracking method has been proposed for airborne videos to solve the association problem caused by unreliable object detections. To robustly track objects in complex scenarios, we proposed an efficient Hierarchical Association framework based on the tracklet confidence and an FCT-based appearance tracking for multiple object tracking in airborne videos. The proposed framework can handle well tracklet generation, progressive trajectory construction, tracklet drifting and fragmentation. Each association stage of the hierarchical framework solves different assignment problems achieving reliable performance with 16 frames per second in MATLAB. The obtained results demonstrated the effectiveness of our framework compared to state-of-the-art methods. In the future we will seek for reproaches combining the proposed motion compensation based detector with an on-line multi-object detection approach to reduce the false alarm rate of detections, as well as consider a deep learning approach for a better objects re-identification after long-term occlusion.

Author Contributions: Ting Chen and Hichem Sahli contributed to the idea. Ting Chen designed the algorithm and wrote the source code, compared the work with other systems and wrote the manuscript. Andrea Pennisi revised the entire manuscript. Zhi Li contributed to the acquisition and annotation of the SAIIP data set. Yanning Zhang provided suggestions on the experiment. Hichem Sahli provided most of the equations of the algorithm and meticulously revised the entire manuscript.

Acknowledgments: This work is supported by the National Natural Science Foundation of China (Nos. 61672429, 61272288, 61231016, 61303123), CSC-VUB scholarship (No. 201406290121), ShenZhen Science and Technology Foundation (No. JCYJ20160229172932237), the Northwestern Polytechnical University (NPU) New AoXiang Star (No. G2015KY0301), the Fundamental Research Funds for the Central Universities (No. 3102015AX007), and the Research Foundation Flanders (FWO) through the CHIST-ERA COACHES project (No. GA.018.14N).

Conflicts of Interest: The authors declare no conflict of interest.

1. Wan, M.; Gu, G.; Qian, W.; Ren, K.; Chen, Q.; Zhang, H.; Maldague, X. Total Variation Regularization Term-Based Low-Rank and Sparse Matrix Representation Model for Infrared Moving Target Tracking. *Remote Sensing* **2018**, *10*, 510.
2. Skoglar, P.; Orguner, U.; Törnqvist, D.; Gustafsson, F. Road Target Search and Tracking with Gimballed Vision Sensor on an Unmanned Aerial Vehicle. *Remote Sensing* **2012**, *4*, 2076–2111.
3. Leitloff, J.; Rosenbaum, D.; Kurz, F.; Meynberg, O.; Reinartz, P. An Operational System for Estimating Road Traffic Information from Aerial Images. *Remote Sensing* **2014**, *6*, 11315–11341.
4. Cao, Y.; Wang, G.; Yan, D.; Zhao, Z. Two Algorithms for the Detection and Tracking of Moving Vehicle Targets in Aerial Infrared Image Sequences. *Remote Sensing* **2015**, *8*, 28.
5. Dey, S.; Reilly, V.; Saleemi, I.; Shah, M. Detection of independently moving objects in non-planar scenes via multi-frame monocular epipolar constraint. In *ECCV*; 2012; pp. 860–873.
6. Yang, B.; Nevatia, R. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. CVPR, 2012, pp. 1918–1925.
7. Yu, Q.; Medioni, G. Motion pattern interpretation and detection for tracking moving vehicles in airborne video. CVPR, 2009, pp. 2671–2678.
8. Cao, X.; Wu, C.; Lan, J.; Yan, P. Vehicle Detection and Motion Analysis in Low-Altitude Airborne Video Under Urban Environment. *IEEE Transactions on Circuits Systems for Video Technology* **2011**, *21*, 1522–1533.
9. Luo, W.; Zhao, X.; Kim, T.K. Multiple object tracking: A review. *arXiv preprint arXiv:1409.7618* **2014**.
10. Prokaj, J.; Duchaineau, M.; Medioni, G. Inferring tracklets for multi-object tracking. CVPRW, 2011, pp. 37–44.
11. Reilly, V.; Idrees, H.; Shah, M. Detection and tracking of large number of targets in wide area surveillance. In *Computer Vision–ECCV 2010*; Springer, 2010; pp. 186–199.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

12. Berclaz, J.; Fleuret, F.; Turetken, E.; Fua, P. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2011**, *33*, 1806–1819.
13. Pirsiavash, H.; Ramanan, D.; Fowlkes, C.C. Globally-optimal greedy algorithms for tracking a variable number of objects. CVPR, 2011, pp. 1201–1208.
14. Bae, S.H.; Yoon, K.J. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2014, pp. 1218–1225.
15. Xiao, J.; Cheng, H.; Sawhney, H.; Han, F. Vehicle detection and tracking in wide field-of-view aerial video. CVPR, 2010, pp. 679–684.
16. Prokaj, J.; Zhao, X.; Medioni, G. Tracking many vehicles in wide area aerial surveillance. CVPRW, 2012, pp. 37–43.
17. Pollard, T.; Antone, M. Detecting and tracking all moving objects in wide-area aerial video. CVPRW, 2012, pp. 15–22.
18. Prokaj, J.; Medioni, G. Persistent Tracking for Wide Area Aerial Surveillance. CVPR, 2014, pp. 1186–1193.
19. Yun, K.; Choi, J.Y. Robust and fast moving object detection in a non-stationary camera via foreground probability based sampling. IEEE International Conference on Image Processing, 2015, pp. 4897–4901.
20. Yin, Z.; Collins, R. Moving object localization in thermal imagery by forward-backward MHI. Conference on Computer Vision and Pattern Recognition Workshop. IEEE, 2006, pp. 133–133.
21. Kim, S.W.; Yun, K.; Yi, K.M.; Kim, S.J.; Choi, J.Y. Detection of moving objects with a moving camera using non-panoramic background model. *Machine vision and applications* **2013**, *24*, 1015–1028.
22. Moo Yi, K.; Yun, K.; Wan Kim, S.; Jin Chang, H.; Young Choi, J. Detection of moving objects with non-stationary cameras in 5.8 ms: Bringing motion detection to your mobile device. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 27–34.
23. Bae, S.H.; Yoon, K.J. Robust Online Multiobject Tracking With Data Association and Track Management. *IEEE Transactions on Image Processing* **2014**, *23*, 2820–2833.
24. Xing, J.; Ai, H.; Lao, S. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. CVPR. IEEE, 2009, pp. 1200–1207.
25. Breitenstein, M.D.; Reichlin, F.; Leibe, B.; Koller-Meier, E.; Van Gool, L. Online multiperson tracking-by-detection from a single, uncalibrated camera. *PAMI* **2011**, *33*, 1820–1833.
26. B., L. Robust Online Multiobject Tracking With Data Association and Track Management. *IEEE Transactions on Image Processing* **2014**, *23*, 2820–33.
27. Ju, J.; Kim, D.; Ku, B.; Han, D.K.; Ko, H. Online Multi-object Tracking Based on Hierarchical Association Framework. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 34–42.
28. Zhang, K.; Zhang, L.; Yang, M.H. Real-time compressive tracking. In *European Conference on Computer Vision*; Springer, 2012; pp. 864–877.
29. Zhang, K.; Zhang, L.; Yang, M.H. Fast compressive tracking. *IEEE transactions on pattern analysis and machine intelligence* **2014**, *36*, 2002–2015.
30. Gray, D.; Tao, H. *Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features*; Springer Berlin Heidelberg, 2008; pp. 262–275.
31. Ali, S.; Shah, M. COCOA: tracking in aerial imagery. Defense and Security Symposium. International Society for Optics and Photonics, 2006, pp. 62090D–62090D.
32. Alatas, O.; Yan, P.; Shah, M. Spatio-temporal regularity flow (SPREF): Its Estimation and applications. *IEEE Transactions on Circuits and Systems for Video Technology* **2007**, *17*, 584–589.
33. Yalcin, H.; Hebert, M.; Collins, R.; Black, M.J. A flow-based approach to vehicle detection and background mosaicking in airborne video **2005**. *2*, 1202–1202.
34. Cao, X.; Lan, J.; Yan, P.; Li, X. Vehicle detection and tracking in airborne videos by multi-motion layer analysis. *Machine Vision and Applications* **2012**, *23*, 921–935.
35. Cao, X.; Gao, C.; Lan, J.; Yuan, Y.; Yan, P. Ego motion guided particle filter for vehicle tracking in airborne videos. *Neurocomputing* **2014**, *124*, 168–177.
36. Cao, X.; Shi, Z.; Yan, P.; Li, X. Tracking vehicles as groups in airborne videos. *Neurocomputing* **2013**, *99*, 38–45.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2018 doi:10.20944/preprints201807.0238.v1

Peer-reviewed version available at Remote Sens. 2018, 10, 1347; doi:10.3390/rs10091347

37. Liu, K.; Ma, B.; Zhang, W.; Huang, R. A spatio-temporal appearance representation for viceo-based pedestrian re-identification. Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3810–3818.
38. Zapletal, D.; Herout, A. Vehicle Re-Identification for Automatic Video Traffic Surveillance. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 25–31.
39. Liu, X.; Liu, W.; Ma, H.; Fu, H. Large-scale vehicle re-identification in urban surveillance videos. 2016 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2016, pp. 1–6.
40. Liu, X.; Liu, W.; Mei, T.; Ma, H. A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance. European Conference on Computer Vision. Springer, 2016, pp. 869–884.
41. Ahuja, R.K.; Magnanti, T.L.; Orlin, J.B. Network Flows: Theory, Algorithms, and Applications **1993**.
42. Kuo, C.H.; Huang, C.; Nevatia, R. Multi-target tracking by on-line learned discriminative appearance models. CVPR, 2010, pp. 685–692.
43. Qin, Z.; Shelton, C.R. Improving multi-target tracking via social grouping. CVPR, 2012, pp. 1972–1978.
44. Yamaguchi, K.; Berg, A.C.; Ortiz, L.E.; Berg, T.L. Who are you with and Where are you going? Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011, pp. 1345–1352.
45. Collins, R.; Zhou, X.; Teh, S.K. An open source tracking testbed and evaluation web site. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2005, pp. 17–24.
46. Li, Y.; Huang, C.; Nevatia, R. Learning to associate: Hybridboosted multi-target tracker for crowded scene. CVPR, 2009, pp. 2953–2960.