Integrating Geometric, Motion and Appearance Constraints for Robust Tracking in Aerial Videos

Mahmudul Hasan, Elliot Staudt, and Amit K. Roy-Chowdhury

Abstract

The analysis of videos from aerial platforms remains a challenging and important problem. The most fundamental task in this regard is to be able to detect and track objects reliably from a moving platform. In this paper, we address the problem of multi-target detection and tracking in unconstrained aerial videos. Generally, aerial videos are very unstable due to air turbulence and targets of interest have few discriminating features, which impose strong challenges in tracking objects such as humans and vehicles. In our proposed approach, we stabilize an unstable aerial video using homography transformation. We estimate the homography between two frames of an unstable video by utilizing the geometric constraint of the ground plane. In order to detect targets in a stabilized video frame, we detect motion regions and then identify targets of interest around the motion regions using appearance based pre-trained classifiers. We devise a finite state machine (FSM) that incorporates both motion detection and target classification into a Kalman filter (KF) based tracking-by-detection framework for robustly tracking humans and vehicles across the aerial video frames. Finally, we associate the tracklets by using overlap and appearance based bipartite graph matching and homography projection of the tracklets. We conduct extensive experiments on challenging aerial video datasets, which prove the robustness of our approach compared to other state-of-the-art tracking approaches.

I. Introduction

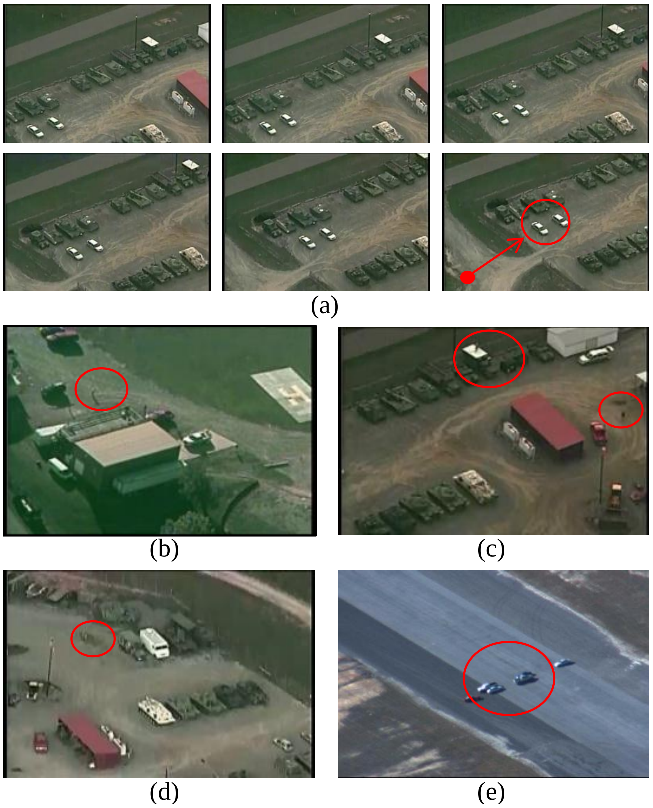
Unmanned aerial vehicles (UAV) are very popular in various applications like law enforcement, firefighting, aerial surveillance, oil and gas exploration, transportation, and scientific research. They are generally equipped with global positioning system (GPS) and image capturing and processing devices for collecting videos. These aerial videos are important sources of information and can be exploited in human and vehicle detection and tracking [1]–[6], scene understanding, mosaicking, and human-object interactions. One of the vital tasks in aerial video analysis is the detection and tracking of moving objects such as humans and vehicles.

Tracking moving objects in aerial videos is much more difficult than in ground-based videos captured by either fixed or moving cameras, as illustrated in Figure 1. Aerial videos are generally captured by UAVs or aerial platforms, which are subjected to a host of complicating factors. Strong vibration and air turbulence as well as rapid changes of velocity and movement perturb the video. As a result, captured videos are unstable, noisy, and blurred. Several preliminary processing steps such as video stabilization, noise removal, and deblurring are necessary. In addition to this, frequent changes in relative motion between the aerial platform and the moving object make it more difficult to perform detection and tracking. Moreover, aerial videos generally suffer from low resolution and low contrast, and the target of interest is usually only a few pixels in size with few discriminating features. Aerial videos are also subjected to occlusion and clutter. Because of these challenges, conventional visual tracking algorithms do not provide good results on aerial videos.

In this paper, we address the issue of multi-target tracking problem in unconstrained aerial videos. We aim to detect and track moving objects such as humans and vehicles. The overall framework of our proposed approach is illustrated in Figure 2. Since the overall algorithm integrates geometric, motion and appearance constraints, we term it as the GMAC aerial video tracker. GMAC tracker begins with stabilizing the video using homography transformation, which is estimated by utilizing the geometric constraint of the ground plane. We divide the video sequence into a number of equal length segments, which we refer to as the video segments. The first frame of each video segment is used as the reference frame. All other

Mahmudul Hasan is with the Department of Computer Science and Engineering, University of California at Riverside, CA-92521, USA.

Elliot Staudt, and Amit K. Roy-Chowdhury are with the Department of Electrical Engineering, University of California at Riverside, CA-92521, USA, (Email: amitrc@ee.ucr.edu).



(a)

(b)

(c)

(d)

(e)

Fig. 1: Figures in (a) show six consecutive frames of an aerial video scene, which illustrate the necessity of video stabilization. Due to the significant motion of the camera, two white stationary cars in the image move from the edge to the center in a fraction of a second. Figures (b, c, d, and e) illustrate (in red circle) some challenges associated with the aerial video tracking such as noise, targets with few pixels, low contrast, and occlusion respectively. These images are taken from VIRAT [7] and VIVID [8] datasets.

frames in the segment are stabilized with respect to this reference frame. We use SURF [9] to detect keypoints in the frames of each video segment. Good matching keypoints between the reference frame and the frame to be stabilized are used to estimate the corresponding homography matrix with RANSAC [10].

After video stabilization, we use an adaptive background subtraction algorithm to identify motion regions in the video frames. These motion regions might correspond to moving objects or static 3D structures due to stabilization error. We filter out the spurious motion regions using a median filter and connected motion regions are formed by applying morphological dilation and erosion. However, motion regions can present an incorrect description of the moving objects. Often motion will be detected at areas of greatest contrast such as the front and back ends of a vehicle. To resolve this problem, in addition to motion detection, we apply a Haar appearance feature based cascaded classifier surrounding the motion regions to recognize humans and vehicles. We devise a finite state machine (FSM) that incorporates both motion detection and target classification into a Kalman filter (KF) based tracking-by-detection framework for robustly tracking humans and vehicles across the aerial video frames. During tracking, detections in a frame are associated with the currently tracked targets using a bipartite graph matching scheme, that uses two metrics: overlapping bounding box and appearance similarity. Later, we use a two stage tracklet association algorithm. In the intra-segment tracklet association, we use the bipartite graph matching based Hungarian algorithm to associate tracklets inside a video segment, whereas, in the inter-segment tracklets association, we use homography projection to associate tracklets between two video segments.

In this research work, our main contributions are as follows: (a) We introduce a new scheme during video stabilization to deal with the problem of stabilization error. (b) We simultaneously apply both motion detection and target classification during target detection, which is very effective, particularly for

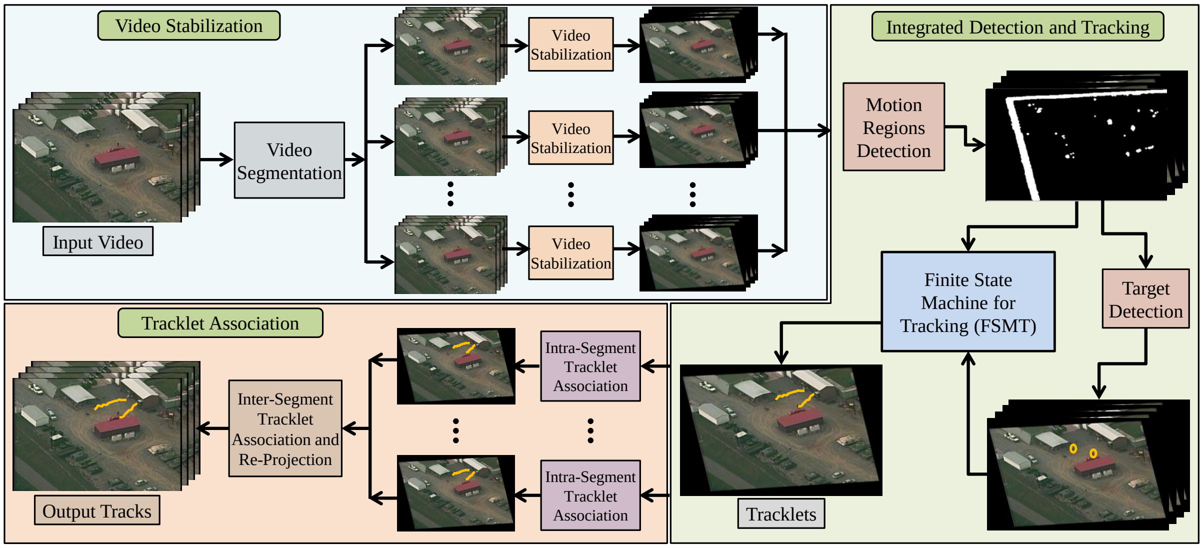

Video Stabilization
Input Video
Video Segmentation
Video Stabilization
Video Stabilization
Video Stabilization
Integrated Detection and Tracking
Motion Regions Detection
Finite State Machine for Tracking (FSMT)
Target Detection
Tracklets
Tracklet Association
Intra-Segment Tracklet Association
Intra-Segment Tracklet Association
Inter-Segment Tracklet Association and Re-Projection
Output Tracks

Fig. 2: The overall framework of the proposed GMAC aerial video tracking algorithm. After segmenting and stabilizing the video, we find the motion regions using an adaptive background subtraction algorithm. An offline trained Haar feature based boosted cascaded classifier is used to recognize humans and vehicles around these motion regions. Motion and target detection are integrated with the Kalman filter. Finally, we stitch the tracklets and re-project the tracks to the original plane.

aerial videos where the targets of interest are very small and have few discriminating features. (c) We devise a FSM that incorporates both motion detection and target classification into a Kalman filter based tracking-by-detection framework.

The rest of the paper is organized as follows. Some related works of state-of-the-art tracking and aerial video tracking are discussed in Section II. Video stabilization and target detection methodology are described in Section III, whereas target tracking methodology are described in Section IV. Detailed experiments and results are presented in Section V. Some concluding remarks are presented in Section VI along with discussion and future research direction.

II. Related Works

In this Section, we provide an overview of the recent advances in visual tracking research both in conventional videos and aerial videos. A comprehensive review on visual tracking can be found in [11], which presents a detailed analysis of various tracking approaches. A survey of more recent advances and state-of-the-art visual tracking methods are discussed in [12]. We would like to refer to these two articles for elaborate reviews on tracking.

A well-known approach of tracking is feature tracking, where an object is represented as a set of distinguishing features. Tracking is performed by finding the correspondence of object features across the frames. Among many types of features, gradient feature such as shape/contour [13], edgelets [14], SIFT [15], SURF [9], HOG [16] etc. are widely used to represent objects. In deterministic feature tracking [17], features between two consecutive frames are associated by using optimal assignment methods such as Hungarian algorithm and greedy search, whereas, in probabilistic feature tracking, state space approaches such as Kalman filters [18] and particle filters [19], [20] are used to take care of the uncertainty in measurement and system model. Tracking multiple objects requires a joint solution of data association and state estimation. Joint Probability Data Association Filtering (JPDAF) [21] and Multiple Hypothesis Tracking (MHT) [22] are two popular techniques for data association. However, appearance of a target may not remain constant throughout the video; it may change for several reasons such as pose and shape variation, different illuminations, change of camera motion and viewpoint, occlusion, etc.

Recently developed tracking approaches take care of appearance variations by incrementally updating the target appearance model online using various techniques such as online subspace learning [23], online adaptive learning of appearance models [24], online multiple instance learning [25], online boosting [26], joint segmentation and tracking [27], tracking-by-detection [28], etc. Some recent methods integrate the context information such as auxiliary objects around the targets [29], relative size of the targets [28], and contour of the object [30] into the tracking framework. Advantages of online and offline learning are combined in [31] in order to provide more information to the tracker. Unlike single target tracking, multi-target tracking approaches pay more attention on data association techniques such as learning based hierarchical data association [32], path estimation in network flow for matching observations [33], etc.

Some visual tracking approaches have been developed specially for use on aerial videos. In [1], the problem of noise in video is handled by an adaptive target appearance model, which is updated online. Video stabilization is performed through frame to frame affine transformations of the target and a particle filter is used for target tracking. However, it works only for single target tracking, while our proposed method is developed for multi-target tracking. [2] presents a vehicle detection scheme in an urban environment, where an attention focusing algorithm is developed to reduce the search area in the frame into a smaller salient region based on color and motion orientation. Then, an adaboost classifier is used for the actual vehicle detection. However, this method is developed for working only in urban environments and relies on traffic patterns to detect vehicles, whereas our proposed approach allows targets to have any motion pattern. Research in [3] addresses the issue of parallax during aerial video stabilization. It uses two geometric constraints such as epipolar and 3D scene structure constraints for filtering out these parallax regions and JPDAF for tracking moving objects. However, their approach works for low altitude aerial videos, whereas our proposed approach is capable of tracking in both of the low and the high altitude aerial videos. In [5], a multi-vehicle tracking method in a wide area scene is proposed, where the motion pattern of the vehicles are learned to deal with the problem of limited appearance information and to reduce false alarms. A motion pattern distribution is incorporated in the probabilistic tracking model. However, it is not always possible to learn the motion pattern in all scenarios. A human detection method in aerial video is proposed in [4]. It uses the geometric constraints from the camera's meta data to find the relationship between a human and its shadow, which is further used to find human blobs and their initial locations. Wavelet based features and SVMs are used to classify these blobs as human or not. However, this method is not applicable in all situations, where camera parameter metadata and shadow of the objects are not available.

III. Target Detection Methodology

In this Section, we provide a detailed description of the target detection methodology of GMAC aerial video tracker, which is comprised of stages such as video stabilization, motion detection, and target classification.

A. Video Stabilization Using Geometric Constraint

Recall from Section I, since, aerial videos are unstable due to the rapid movement of the aerial platform and turbulence, it is important to stabilize aerial videos before applying the target detector and the tracking algorithm. Research work in [34] addresses the issue of video stabilization in turbulence. In this work, we stabilize the aerial videos using homography transformation, which is estimated from the video using the geometric constraint of the ground plane. A homography is the invertible mapping of points and lines in the projective plane. In video registration, a homography is the process of projecting an image frame to the plane of a reference image frame. A homography matrix is a non singular 3×3 homogeneous matrix with eight degrees of freedom. Generally, homographies are estimated by finding feature correspondences between two images. Interest point feature correspondences are commonly used. In this work, we use the SURF [13] keypoint detector to find keypoints in both of the reference frame and the frame to be stabilized. Each keypoint belonging to the reference frame is matched to the keypoints in the frame to be stabilized. Good matching keypoints are used to compute the homography matrix. Since a homography matrix has eight degrees of freedom and each point correspondence provides two equations, four point correspondences are sufficient to compute the homography matrix. The only restriction is that no three points can be collinear.

Video stabilization through homography transformation works well for planar objects, but 3D objects such as buildings, towers, and most importantly shapes of the humans and vehicles become distorted because of this transformation. This shape distortion becomes much worse if the frame to be stabilized is far away from the reference frame. Thus, it becomes increasingly difficult for the target detector and the tracker to achieve better performance on these distorted targets. In order to solve this problem, we divide the whole video sequence into a number of equal length segments. These video segments are stabilized separately. The first frame of each video segment is used as the reference frame and all other frames are stabilized with respect to this reference frame. However, the length of the video segment has significant effect on stabilization and tracking performance. If a video segment is too lengthy, human and vehicle shapes will be distorted more towards the end of the video segment and thus, target detection and tracking will inefficient. On the other hand, if a video segment is too small, the length of the tracklets generated by the tracker will also be small and thus, tracklets association will be more difficult. Algorithm 1 details overall video stabilization process.

Algorithm 1: Algorithm for stabilizing the unstable aerial videos.

```
Data: Unstable aerial video frames: $\{f_1, f_2, \ldots, f_n\}$.
Result: Stable aerial video frames: $\{f'_1, f'_2, \ldots, f'_n\}$.
$i \leftarrow 1$;
while not at the end of the video do
    $f_i \leftarrow$ ReadNextFrame(); $i \leftarrow i + 1$;
    if a new segment begins then
        $f_{ref} \leftarrow f_i$;
        $KP_{ref}$ = DetectSurfKeypoint($f_{ref}$);
        $Desc_{ref}$ = Descriptor($KP_{ref}$);
    end
    $KP_i$ = DetectSurfKeypoint($f_i$);
    $Desc_i$ = Descriptor($KP_i$);
    $M$ = FindMatch($Desc_{ref}$, $Desc_i$);
    $H$ = FindHomography($M$, $RANSAC$);
    $f'_i$ = ProjectiveTransform($f_i$, $H$);
end
```

B. Motion Region Detection

In this work, we use a simple but robust background subtraction based algorithm to detect motion regions in the stabilized video frames described in [35]. At first, we construct an initial background model (B_1) by taking the arithmetic mean of the pixel values of first N frames of a video segment, assuming that there are no moving objects, as follows:

$$B_1(i, j) = \frac{1}{N} \sum_{k=1}^{N} I_k(i, j)$$

where, $B_1(i, j)$ is the intensity of the pixel (i, j) of the initial background model and $I_k(i, j)$ is the intensity of the pixel (i, j) of the k^{th} frame. Then, we obtain the difference between the current frame at time t

and the background model at time $t-1$, as follows:

$$D_t(i,j) = |I_t(i,j) - B_{t-1}(i,j)|.$$

The pixels in the difference image are classified, as follows:

$$D_t(i,j) \in \begin{cases} \textit{Foreground} & \text{if } D_t(i,j) \geq T_d \\ \textit{Background} & \text{if } D_t(i,j) < T_d \end{cases}$$

Then, we update the background model in order to accommodate the dynamics of the scene, as follows:

$$B_t(i,j) = \alpha_t I_t(i,j) + (1-\alpha_t) B_{t-1}(i,j)$$

where, α_t is the learning rate of the background model. We further apply a median filter in the foreground image to remove spurious and smaller noisy regions. Then, we sequentially apply morphological image dilation, erosion, and dilation to obtain connected motion regions.

C. Appearance Based Target Classification

In addition to motion region detection, we apply a classifier around the motion regions to detect the targets of interest. However, detecting humans or vehicles in high altitude aerial video is difficult because the number of pixels representing them in the video frames is too small and the extracted features are not discriminating enough. Therefore, training a single classifier is not effective and has less generalization power. A very good solution to this problem is to train a number of classifiers and arrange them in a cascade [36]. The resultant classifier is called a cascade because it is comprised of several classifiers and applied subsequently to a candidate until, at some stage, the candidate is rejected or all the stages are passed, as shown in Figure 3. The classifier at every stage of the cascade consists of several basic boosted classifiers. Classifiers built this way are efficient and have relatively high generalization power. In this work, we use a Haar feature based boosted cascaded classifier [37]. We train separate detectors for humans and vehicles.

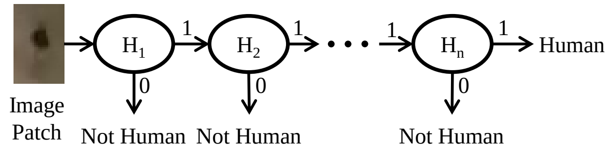


Fig. 3: Haar feature based boosted cascaded classifier for human and vehicle detection. Each stage is comprised of a number of boosted simple classifiers.

IV. Target Tracking Methodology

In this Section, we provide a detailed description of the target tracking methodology of GMAC aerial video tracker, which is comprised of tracking with Kalman filter (KF) and tracklets association.

A. Tracking with Kalman Filter

System Model and Kalman Filter: Due to the instability of the aerial videos, the system transition model and the observation model governing the dynamical system as shown in Equations 1 and 2 are not linear.

$$\mathbf{x'}_t = f(\mathbf{x'}_{t-1}) + \mathbf{w'}_{t-1} \tag{1}$$

$$\mathbf{z'}_t = h(\mathbf{x'}_t) + \mathbf{v'}_t \tag{2}$$

where, $\mathbf{x}'_t \in \mathcal{R}^n$ is the system state and $\mathbf{z}'_t \in \mathcal{R}^m$ is the observation at time t in the unstable image plane, f represents the non linear state transition model, and h represents the observation model. The random variables $\mathbf{w}'_t$ and $\mathbf{v}'_t$ represent the process and the measurement noise respectively. They are zero-mean Gaussian noise with covariance matrices Q' and R'. However, we stabilize the video frames using a projective transformation that uses estimated homography matrices. This transformation projects the target state from unstable image plane to the stabilized image plane, as follows:

$$\mathbf{x}_t = P(\mathbf{x}'_t) \tag{3}$$

where, $\mathbf{x}_t$ is the state of a target in the stabilized image plane and P is the projective transformation that use homography matrix. In the stabilized image plane, the trajectory of a target becomes linear, as well as the system transition and observation models. Additionally, observation noise is Gaussian around the ground truth target position, since we concurrently use motion detection and object classification during target detection. In this case, basic Kalman filter provides an optimal tracking solution in terms of error co-variance of state variables. Therefore, in the stabilized image plane, the dynamic system as shown in Equations 1 and 2 become linear and is governed by a linear system and measurement model as shown in Equations 4 and 5.

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + \mathbf{w}_{t-1} \tag{4}$$

$$\mathbf{z}_t = H\mathbf{x}_t + \mathbf{v}_t \tag{5}$$

where, $\mathbf{x}_t \in \mathcal{R}^n$ is the system state and $\mathbf{z}_t \in \mathcal{R}^m$ is the observation state at time t in the stabilized image plane, A represents the state transition model and H represents the observation model. The random variables $\mathbf{w}_t$ and $\mathbf{v}_t$ represent the process and the measurement noise respectively. They are zero-mean Gaussian noise with covariance matrices Q and R.

The Kalman filter is a predictor corrector type estimator, which can be used to solve the above model. In the prediction step, a priori state $\hat{\mathbf{x}}_t^-$ is predicted using a state transition model, whereas in the correction step, the a posteriori state $\hat{\mathbf{x}}_t$ is estimated from the linear combination of the a priori state $\hat{\mathbf{x}}_t^-$ and the new measurement $\mathbf{z}_t$. In our case, we model these state equations as follows.

State Transition Model: We use a constant velocity state transition model as shown in Equation 6. The state vector has four variables, first two are the position of the target, and second two are the velocity vector. The process noise (w_x, w_y, w_u, w_v) for each state variable are independent and drawn from a zero mean normal distribution. The variance of the position noise is (σ_x, σ_y), which we set higher than the variance of the velocity noise, (σ_u, σ_v).

$$\begin{pmatrix} x_t \\ y_t \\ u_t \\ v_t \end{pmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ u_{t-1} \\ v_{t-1} \end{pmatrix} + \begin{pmatrix} w_x \\ w_y \\ w_u \\ w_v \end{pmatrix} \tag{6}$$

Observation Model: The observation model is shown in Equation 7. The observation vector is the position of a target in a frame, while the observation noise (v_x, v_y) is independent and drawn from a zero mean Gaussian distribution.

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ u_{t-1} \\ v_{t-1} \end{pmatrix} + \begin{pmatrix} v_x \\ v_y \end{pmatrix} \tag{7}$$

Tracker Initialization and Tracking: In this work, we combine motion detection and object detection in an unified framework. This approach has a significant effect on target detection and tracking performance as illustrated in Figure 4. Figure 4(a) and 4(b) show the target detection performance due to the application of motion detection alone. Application of only motion detection may result in the fragmentation of a target into multiple parts (Figure 4(a)) or the unification of two targets into an single target (Figure 4(b)). On the other hand, only the application of object detection may result in a lot of false alarms due to the less discriminating target features. These problems can be resolved by the combined application of motion detection and object detection as shown in the second column of Figure 4.

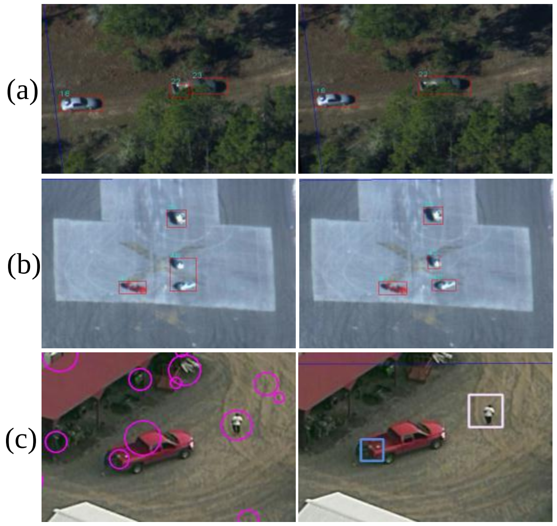


Fig. 4: Only the application of motion detection to track moving objects may result in fragmentation of a single object (left of (a)) and unification of multiple objects (left of (b)). On the other hand, only the application of object detection may results in an increasing number of false alarms (left of (c)). However, the joint application of motion detection and object detection resolve these problems as shown in the right figures of (a), (b), and (c).

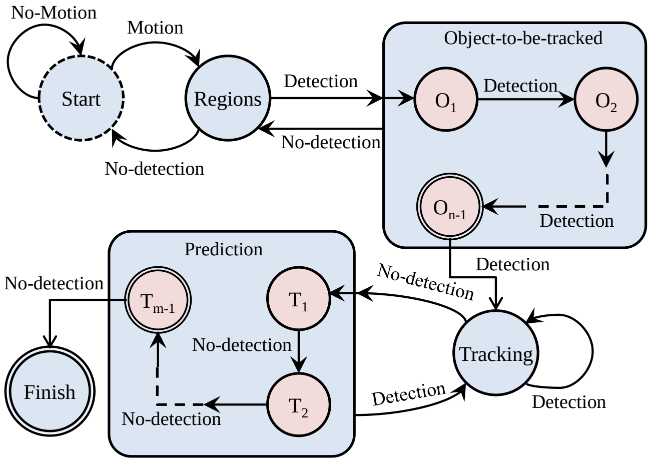


Fig. 5: Finite state machine for tracking (FSMT) targets in aerial videos.

The finite state machine (FSM) of the overall tracking process is illustrated in Figure 5. The definition of the finite state machine is given as follows.

$$\begin{aligned}
\Sigma &= \{\textit{No-motion, Motion, Detection, No-detection}\}\\
S &= \{\textit{Start, Regions, Object-to-be-tracked,}\\
&\quad \textit{Tracking, Prediction, Finish}\}\\
s_0 &= \{\textit{Start}\}\\
\delta &= S \times \Sigma \to S\\
F &= \{\textit{Finish}\}
\end{aligned}$$

where, Σ is the input alphabet, S is the set of states, s_0 is the initial state, δ is the state transition function, and F is the final state. The state transition function (δ) is outlined in Table I.

Current State	**Input**	**Next State**
Start	*No-motion*	*Start*
Start	*Motion*	*Regions*
Regions	*No-detection*	*Start*
Regions	*Detection*	*Object-to-be-tracked*
Object-to-be-tracked	*No-detection*	*Regions*
Object-to-be-tracked	*Detection*	*Tracking*
Tracking	*Detection*	*Tracking*
Tracking	*No-detection*	*Prediction*
Prediction	*Detection*	*Tracking*
Prediction	*No-detection*	*Finish*

TABLE I: State transition function

The FSM for target tracking illustrated in Figure 5 works as follows. At first, the system is in the *Start* state. Using the background subtraction algorithm described in Section III-B, we detect the motion regions in the current frame. If there is significant motion in the current frame, we put the system in *Regions* state, otherwise the system remains in the *Start* state. These motion regions could be the result of the motion of the targets of interest or could be the reason of stabilization error. It could be the broken part of a target or two or more targets may fall into the same motion region. So, we apply the pre-trained classifiers to detect the exact location of the target in the current frame. We update the motion regions after each successful detection. We do not start tracking immediately after the first detection in the current frame because it may be the result of noise or a false alarm. After the first detection, we put the system state into the *Object-to-be-tracked* state. From this state, if we get n consecutive detections, we put the system into the *Tracking* state and start tracking the target. If we do not get any valid detection for the corresponding target when the system is in the *Object-to-be-tracked* state, we put the system back into the *Regions* state.

During the *Tracking* state, if we have a valid detection for the corresponding target, system remains in the *Tracking* state and we continue tracking using the Kalman filter. However, due to noise and stabilization error, system can miss the detection of the corresponding target in some frames. In this case, instead of terminating the tracking immediately, we put the system in *Prediction* state and continue tracking only using the prediction step of the Kalman filter. During *Prediction* state, if the system finds any valid detection of the corresponding target, it moves into the *Tracking* state again. During the *Prediction* state, if the system fails to find any valid detection within m subsequent frames, it terminates the tracking after saving the track information of the target.

Data Association: Data association is one of the most important tasks in multi-target tracking, where the detections in the current frame are associated with the active targets. In this work, we employ a bipartite graph matching scheme. It works well considering the situation in aerial videos, where target motion become linear after video stabilization. Let us consider that we have a set of active targets $\{t_i\}$ up to frame f_k. We also have a set of detections $\{d_j\}$ in frame f_{k+1}. Number of targets and number of detections are not required to be same. We compute two similarity metrics between the active targets,

$\{t_i\}$ and the detections, $\{d_j\}$. We compute weighted arithmetic mean of these two similarity metrics and use Hungarian algorithm to find the optimal assignment between the targets and the detections.

The first similarity metric is the overlap between a target (t_i) and a detection (d_j). This metric is normalized between zero to one, where one means fully overlapped and zero means totally distant. The second metric is the appearance similarity. In order to compute the similarity between a target (t_i) and a detection (d_j), we get the color histograms of these target and detection and compute the Bhattacharya distance between these two color histograms. We subtract this appearance distance from one to obtain the appearance similarity metric.

$$\begin{aligned}&Sim(t_i, d_i) = \alpha Overlap(t_i, d_i) + (1-\alpha) App\_Sim(t_i, d_i)\\&App\_Sim(t_i, d_i) = 1 - Bhat\_Dist(t_i, d_i)\end{aligned}$$

B. Tracklet Association

It is possible for the tracker to prematurely end a track for two reasons. Recalling from Section III-A, since, we divide the video sequence into a number of equal length segments and we initialize a tracker at the beginning of a video segment, we terminate it at the end. Therefore, the length of a tracklet can never be greater than the length of a video segment. Finally, if a tracker failed to find any valid detection of the corresponding target during the *Prediction* state in m consecutive frames as illustrated in Figure 5, it will terminate the tracking of that target. Therefore, we perform tracklet association in two steps: (1) Intra segment tracklet association, and (2) Inter segment tracklet association.

Intra segment tracklet association: In this step, we associate the tracklets, which were terminated due to missed detection, noise, occlusion, and clutter inside a video segment. In these cases, the Kalman filter produces a good estimation of short trajectories known as tracklets. Estimated tracklets are associated based on their similarities. We use appearance and distance similarity models. The appearance similarity between two tracklets is determined based on their color histograms. In order to compute the color histogram of a tracklet, we sample the bounding boxes from the trajectory of the corresponding tracklet. We compute color histogram of a sampled bounding box for each of the three color channels and concatenate them. Then, we compute the arithmetic mean of the color histograms of the sampled bounding boxes to obtain the color histogram of a tracklet. We compute the Bhattacharya distance between two color histograms of two tracklets in order to measure the appearance distance. On the other hand, distance similarity is measured based on the distance between two tracklets. At first, we extrapolate the first tracklet in the forward direction up to the beginning of the second tracklet. Then, we calculate the Euclidean distance between this extrapolated point and the first point of the second tracklet. Finally, we add these two similarity measure and use Hungarian algorithm to compute an optimal assignment among all the tracklets.

Inter segment tracklets association: In this step, we associate the tracklets, which were produced due to video segmentation. Let f_i and f_j be the first and the last frame of a video segment respectively and f_k be the first frame of the next video segment. Let h_j and h_k be the homography transformation between between the frames (f_i, f_j) and (f_i, f_k) respectively. Let $\{t_j\}$ be the set of last bounding boxes of a set of tracks terminated at frame f_j and $\{t_k\}$ be the first bounding boxes of a set of tracks initiated from frame f_k. Without loss of generality, let us assume that the number of bounding boxes in $\{t_j\}$ and $\{t_k\}$ are same. We associate the $\{t_j\}$ and $\{t_k\}$ as follows. At first, we compute the homography transformation, h_{jk} between the frames f_j and f_k by the following equation:

$$h_{jk} = h_j / h_k.$$

Then, we project the bounding boxes $\{t_j\}$ in frame f_j to the plane of frame f_k by the following equation:

$$\{t'_j\} = P\ \left(\{t_j\},\ h_{ij}\right)$$

where, $\{t'_j\}$ is the set of projected bounding boxes and P is the projective transformation using homography matrix. If a bounding box in $\{t'_j\}$ overlaps with a bounding box in $\{t_k\}$, we associate respective tracklets.

C. Overall Tracking Algorithm

The overall tracking process is detailed in Algorithm 2. Codes of the overall system will be available at http://www.ee.ucr.edu/~amitrc/AerialVideoAnalysis.php.

Algorithm 2: GMAC aerial video tracking

```
Data: Unstable aerial video:$\{f_1, f_2, \ldots, f_n\}$.
Result: Tracks
$\{f'_1, \ldots, f'_n\}$ = stabilize ($\{f_1, \ldots, f_n\}$) (Alg. 1);
$i \leftarrow 1$;
while not at the end of the video do
    $f'_i \leftarrow$ readNextStableFrame(); $i \leftarrow i + 1$;
    if a new video segment begins then
        $B \leftarrow$ Compute initial background model;
    $regions \leftarrow$ DetectMotion($f'_i$, $B$);
    $B \leftarrow$ Update the background model;
    foreach $regions$ do
        Run FSMT. Figure 5.
    end
end
foreach Video Segment do
    foreach Two tracklets $t_i$ and $t_j$ do
        Compute similarity matrix; (Sec. IV-B).
    end
    Run Hungarian algorithm on the similarity matrix;
end
foreach Two Consecutive Video Segment do
    Run Inter Segement Tracklet Association; (Sec. IV-B).
end
foreach Track do
    Repoject the track to the original image ground plane using inverse homography transformation.
end
```

V. Experiments

We use two state-of-the-art challenging aerial video datasets VIRAT and VIVID, for the evaluation of our proposed approach. We briefly describe these datasets as follows.

VIRAT: VIRAT [7] aerial video dataset is more challenging than any other publicly available dataset due to rapidly changing viewpoints, illumination, and visibility. This dataset includes buildings and parking lots where people and vehicles are engaged in different kinds of activities. The aerial videos are captured by a camera mounted on a gimbal in a manned aircraft. The video resolution is 640×480 at a $30Hz$ frame rate. Typically, people are about 20 pixels tall.

VIVID: DARPA's VIVID [8] dataset was specifically developed for low-resolution moving target detection, tracking, and activity analysis, which were collected at Eglin during DARPA's VIVID program. In these videos, a number of military and civilian vehicles maneuver on a runway, dirt road, and concrete road. Sometimes vehicles are occluded by either each other or trees. Videos suffer from defocussing, dropped sensor reading, non-smoothness, duplicated frames, discontinuities, and changing illumination. Video resolution is about 640×480 pixels and objects are about 20 to 50 pixels in height and width.

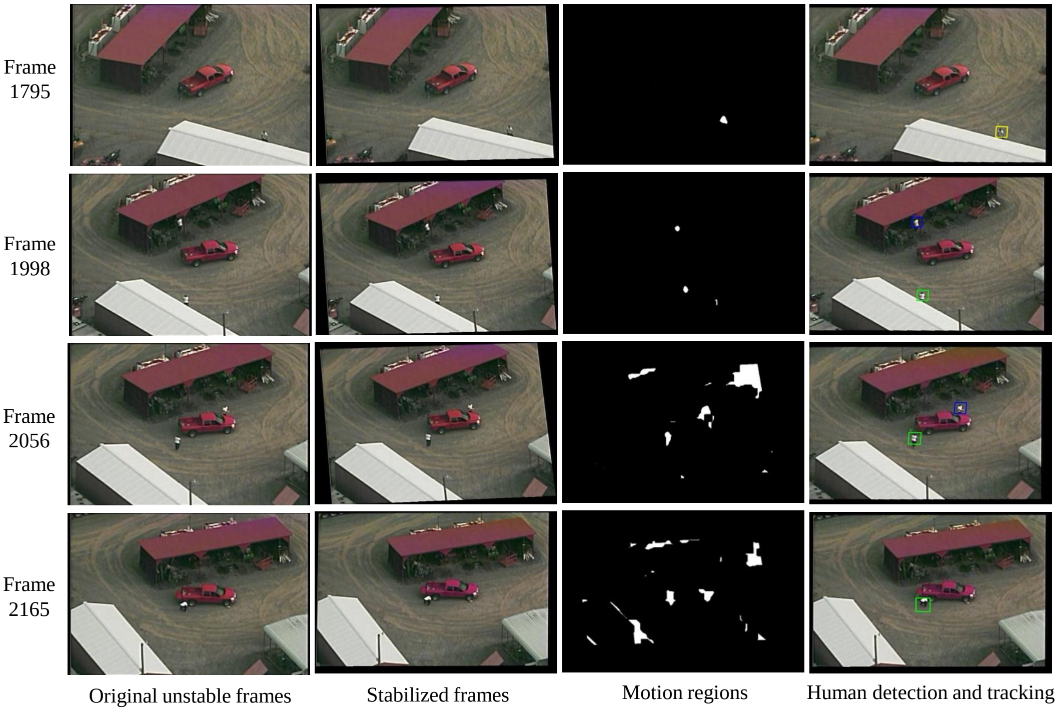
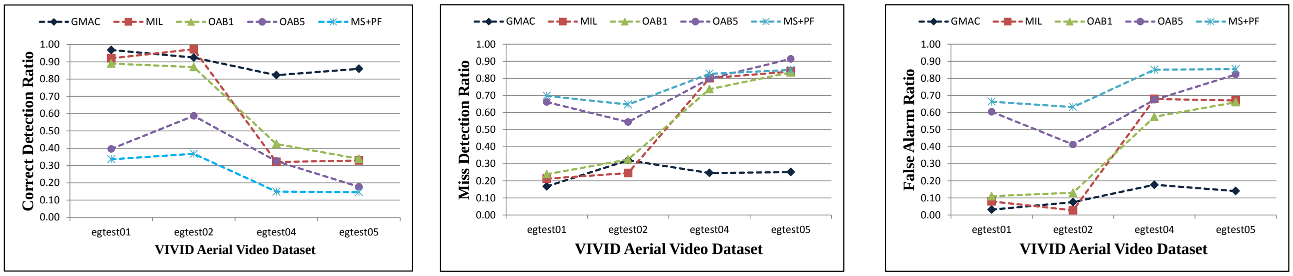


Fig. 6: Outputs generated from an experimental run showing different steps of our proposed GMAC aerial video tracker.

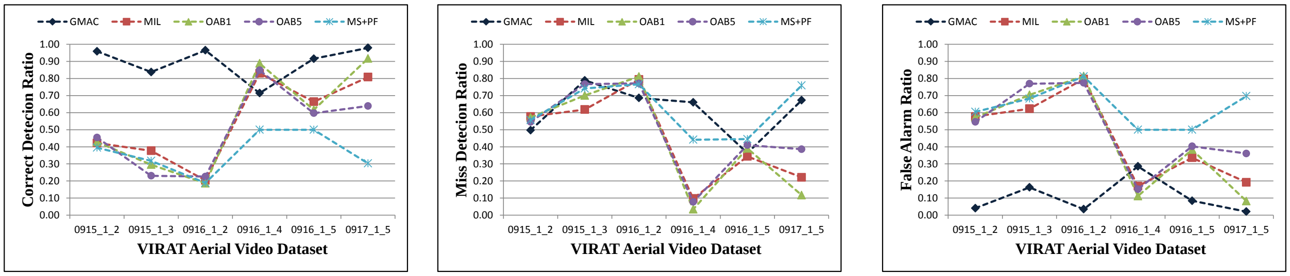
Processing Stages: The results from three processing stages of our proposed approach are illustrated in Figure 6. Four rows of Figure 6 contain results for four different frames (1795, 1998, 2056, and 2165) of a scene from VIRAT. Figures in the first columns contain original video frames. We stabilize these video frames by applying homography transformation. These stabilized frames are shown in the second column. We apply our motion region detection algorithm on these stabilized frames. Motion regions are shown in the third column and finally, the detection and tracking results are shown in the fourth column.



(a) (b) (c)

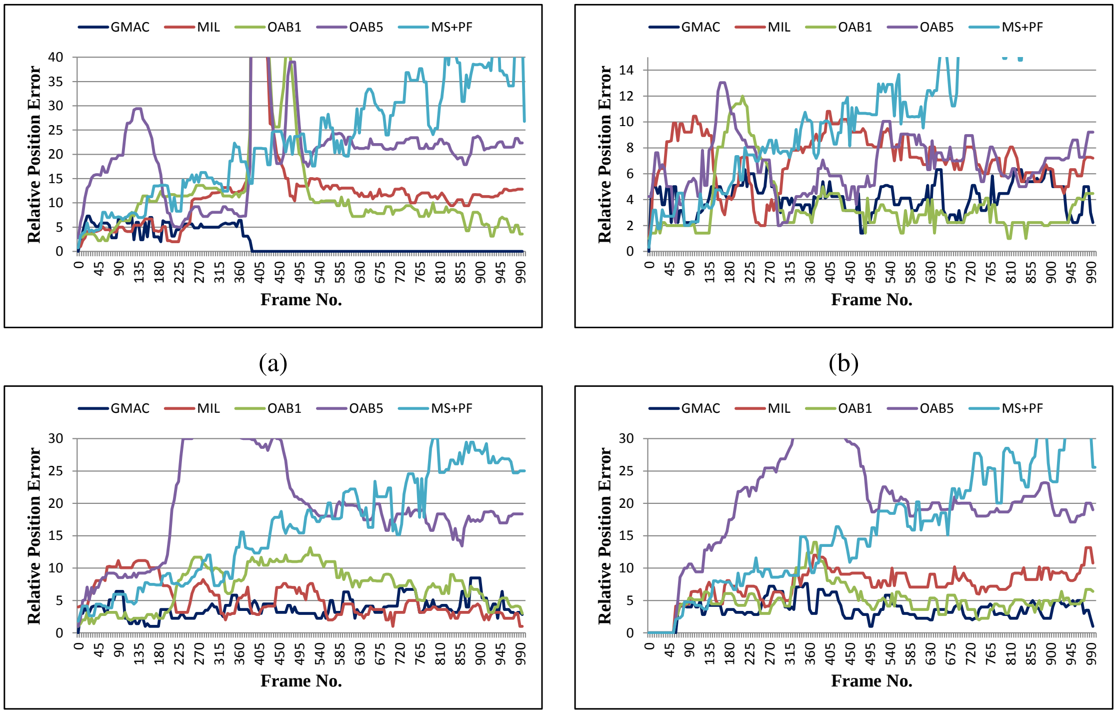
Fig. 7: Vehicle detection statistics for different scenes of VIVID aerial video dataset. (a) Correct detection ratio, (b) Miss detection ratio and (c) False alarm ratio.

Comparative Analysis: We have also compared our method with four state-of-the-art tracking algorithms such as the multiple instance learning (MIL) tracker [25], the online AdaBoost (OAB1) tracker [26], the modified online AdaBoost (OAB5) tracker [26], and the mean shift particle filter (MSPF) tracker. We have used the implementation of MIL, OAB1, and OAB5 from [25]. We implemented the MSPF tracker by ourselves. All the parameters of these algorithms are set to achieve highest performance on the dataset we have used to show comparisons. However, MIL, OAB1 and OAB5 are essentially single target trackers. So, for fair comparison with our method, we run these algorithms separately for each of the targets present in



(a) (b) (c)

Fig. 8: Human detection statistics for different scenes of VIRAT aerial video dataset. (a) Correct detection ratio, (b) Miss detection ratio and (c) False alarm ratio.



(a) (b)

(c) (d)

Fig. 9: Relative position error with respect to the ground truth of four targets in *egtest01* scene from VIVID aerial video dataset.

the videos. In this sense, performance of these algorithms are not affected by the performance degradation due to data association during tracking, which gives them advantages over MSPF and our method.

Target Detection Statistics: Comparisons of target detection statistics among our method, MIL, OAB1, OAB5, and MSPF on VIVID and VIRAT aerial video dataset are shown in Figure 7 and 8 respectively. These statistics are correct detection ratios, missed detection ratios, and false alarm ratios. A detection is considered to be a correct detection if there is at least 50% overlap between the system detection and the ground truth detection. These ratios are computed by the following formulas:

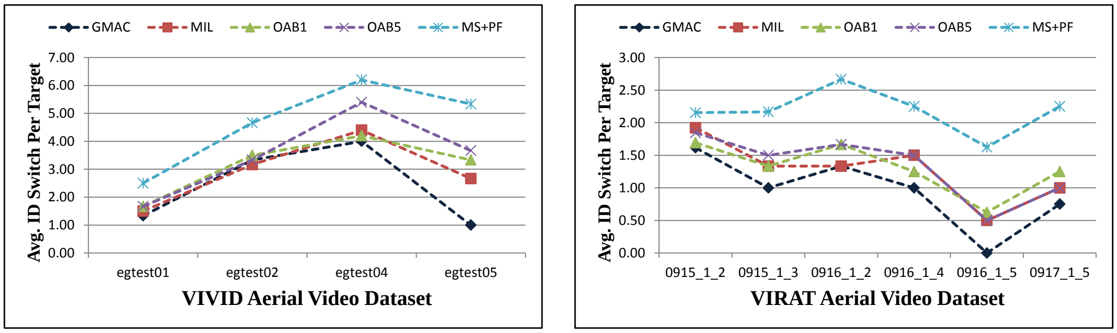
$$Correct\ Detecion\ Ratio = \frac{No.\ of\ Correct\ Detecions}{No.\ of\ System\ Detecions}$$

$$Miss\ Detecion\ Ratio = \frac{No.\ of\ Miss\ Detecions}{No.\ of\ Ground\ Truth\ Detecions}$$

$$False\ Alarm\ Ratio = \frac{No.\ of\ False\ Alarm}{No.\ of\ Ground\ Truth\ Detecions}$$

Plots illustrated in Figures 7(a), (b), and (c) show the comparison of target detection statistics on different scenes of VIVID among different tracking algorithms. It is evident in the plots that our method outperforms other algorithms in every scene except *egtest02*, where only MIL performs better. Performance of the algorithms are better in scenes *egtest01* and *egtest02* compared to the scenes *egtest04* and *egtest05* because the targets of *egtest04* are very small and have low contrast with few discriminating properties, whereas targets of the scene *egtest05* are larger but they are frequently occluded by the trees. Plots illustrated in Figures 8(a), (b), and (c) show the comparison of target detection statistics on different scenes of VIRAT among different tracking algorithms. As shown in Figure 8(a) the correct detection ratio of our algorithm is much higher than the other algorithms except in one scene (0916_1_4). Though the missed detection ratio shown in Figure 8(b) is not much better relative to other algorithms, the false alarm ratio shown in Figure 8(c) of our method is much lower than the other algorithms except in one scene (0916_1_4).

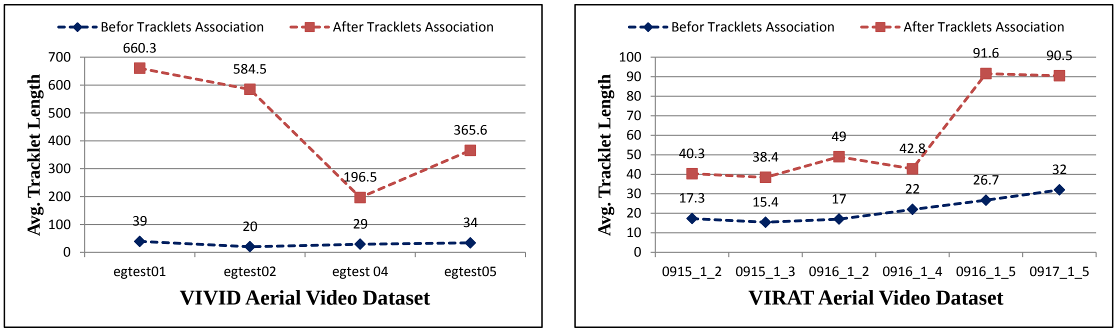
Relative Position Error: Figure 9 shows relative position errors of four targets in a scene of VIVID. It shows that our method is more accurate and consistent than other algorithms.



(a) (b)

Fig. 10: (a) Average per target ID switch for the VIVID aerial dataset, and (b) Average of per target ID switch for the VIRAT aerial video dataset.

Average Id Switch: A comparison of average ID switch for the different algorithms is shown in Figure 10 for different scenes of VIVID and VIRAT. For each scene, we compute the number of ID switches for each target. We sum up these numbers and divide it by the number of targets to determine the average ID switch. For MIL, OAB1, and OAB5, we consider an event as the ID switch if the tracker jumps off to another target or to the background. In these cases, we reinitialize the tracker by giving the position of the target manually in the next frame. On the other hand, for our method and MSPF, we consider an event to be an ID switch if two target switch their IDs. Figure 10(a) shows the comparison on the first four scenes of the VIVID aerial video. In all the scenes, our method outperforms other algorithms. A similar comparison is shown in Figure 10(b) for the VIRAT aerial video dataset, for which our method also outperforms the other algorithms. In VIVID, scenes like *egtest02* and *egtest04* have higher ID switches per target than the other two scenes because in these two scenes, targets are small, close to each other and the camera is also moving faster than other two scenes.

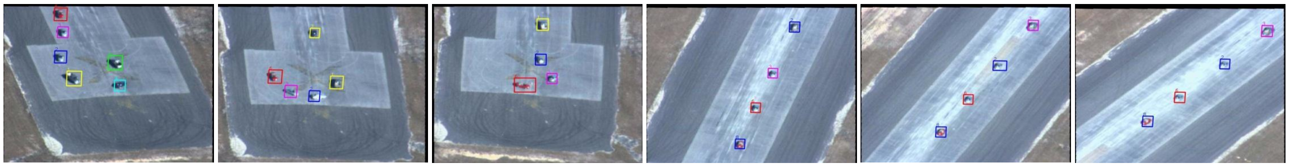


(a) (b)

Fig. 11: Average tracklet length of different scenes of the VIVID aerial video dataset. (a) Average tracklet length before tracklet association, (b) Average tracklet length after tracklet association.

Average Tracklet Length: Figure 11 shows the average length of the tracklets generated by our method in two cases: before and after tracklet association. Since, we divide the video sequence into a number of equal length segments and we initialize a tracker at the beginning of a segment and terminate it at the end, the maximum length of a tracklet before tracklet association can not be greater than the length of a video segment. In our experiment, the length of a video segment is set to fifty frames. Average tracklet length before data association and after data association for VIVID and VIRAT are shown in Figure 11(a) and (b), which is much higher in VIVID than in VIRAT. In VIRAT, tracker tend to lose targets quickly because the targets are frequently occluded by buildings and have less discriminating properties.

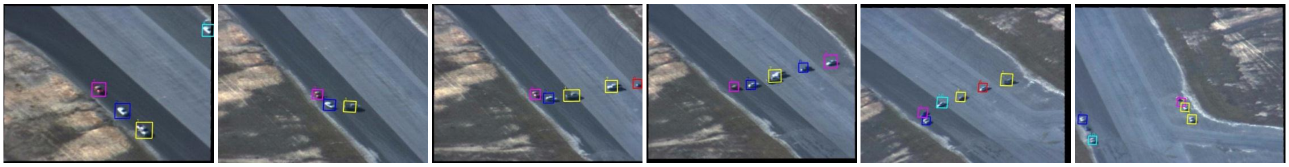
Tracking Results: We have shown some tracking results for different scenes of VIVID and VIRAT in Figures 12 - 18. For more results, we would like to refer the readers to our project web page at http://www.ee.ucr.edu/~amitrc/AerialVideoAnalysis.php.



Frame: 70 Frame: 242 Frame: 354 Frame: 577 Frame: 725 Frame: 947

Fig. 12: Multi-target tracking results of a scene (frame no. 70, 242, 354, 577, 725, and 947) of the VIVID aerial video dataset. Different targets are given different colors and IDs.

Figure 12 shows the tracking results of scene *egtest01* of the VIVID aerial video dataset. This scene is relatively simple compared to other scenes. In this scene, vehicles are looping around the runway and then driving straight. At some point one vehicle speed up and passes another. It is visible in the figure that tracking results are very good as our algorithm is able to track all the targets correctly.



Frame: 23 Frame: 329 Frame: 410 Frame: 475 Frame: 735 Frame: 947

Fig. 13: Multi-target tracking results of a scene (frame no. 23, 329, 410, 475, 735, and 947) of the VIVID aerial video dataset. Different targets are given different colors and IDs.

Figure 13 shows the tracking results of scene *egtest02* of the VIVID aerial video dataset. This scene contains two sets of vehicles passing each other. There is also continuous change of scale as the camera is constantly moving away from the scene. In frame 410, vehicles start to pass each other. In frame 735, vehicle passing is almost completed. It is visible that after the vehicle passing event, the tracker was able to correctly maintain respective targets.

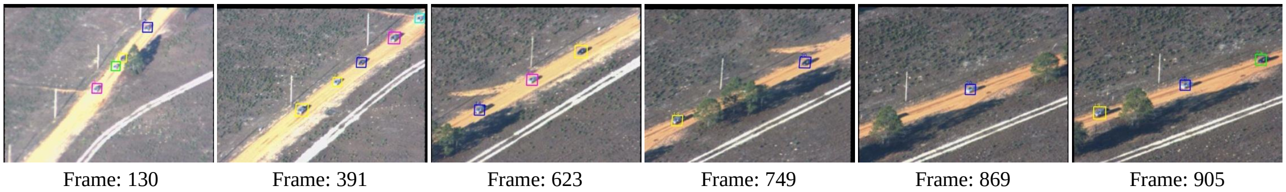

Frame: 130 Frame: 391 Frame: 623 Frame: 749 Frame: 869 Frame: 905

Fig. 14: Multi-target tracking results of a scene (frame no. 130, 391, 623, 749, 869, and 905) of the VIVID aerial video dataset. Different targets are given different colors and IDs.

Figure 14 shows the tracking result of the scene *egtest04* of the VIVID aerial video dataset. Here, a line of cars are traveling down a red dirt road. There are also occlusion by trees, defocusing, and frame dropping. As a result, track switch is more frequent than the other scenes. There are also relatively higher miss detection ratio and false alarm ratio in this scene.

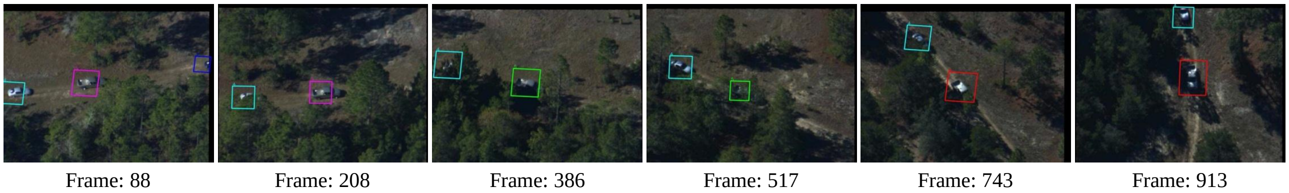

Frame: 88 Frame: 208 Frame: 386 Frame: 517 Frame: 743 Frame: 913

Fig. 15: Multi-target tracking results of a scene (frame no. 88, 208, 386, 517, 743, and 913) of VIVID aerial video dataset. Different targets are given different colors and IDs.

Figure 15 shows the tracking result of scene *egtest05* of the VIVID aerial video dataset. Here the number of vehicles is three; they are relatively larger than other scene. Vehicles are moving along a dirt road in a wooden area. Challenges present in this scene are vehicles frequently occluded by trees, changes of illumination, and changes of camera viewpoint. Despite of these challenges, our method is able to successfully track the targets.

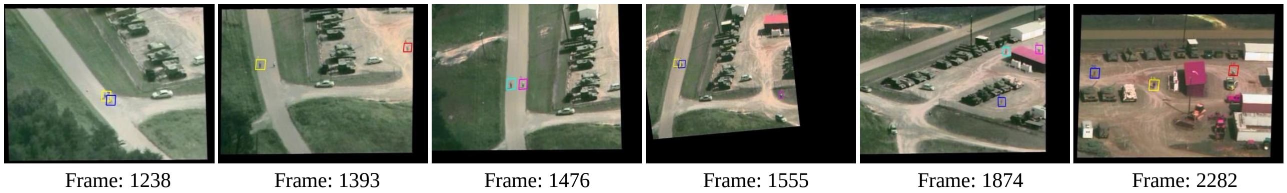

Frame: 1238 Frame: 1393 Frame: 1476 Frame: 1555 Frame: 1874 Frame: 2282

Fig. 16: Multi-target tracking results of a scene (frame no. 1238, 1393, 1476, 1555, 1874, and 2282) of the VIVID aerial video dataset. Different targets are given different colors and IDs.

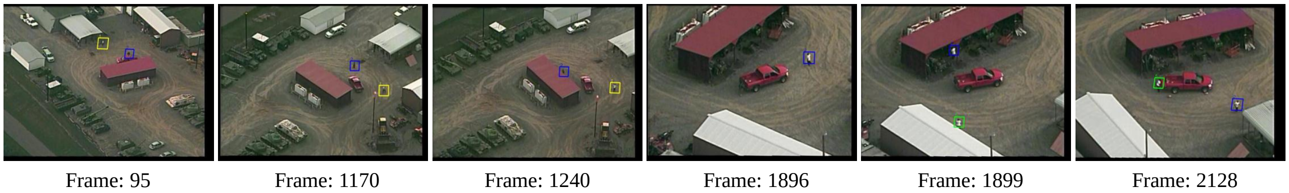

Frame: 95 Frame: 1170 Frame: 1240 Frame: 1896 Frame: 1899 Frame: 2128

Fig. 17: Multi-target tracking results of a scene (frame no. 95, 1170, 1240, 1896, 1899, and 2128) of the VIRAT aerial video dataset. Different targets are given different colors and IDs.

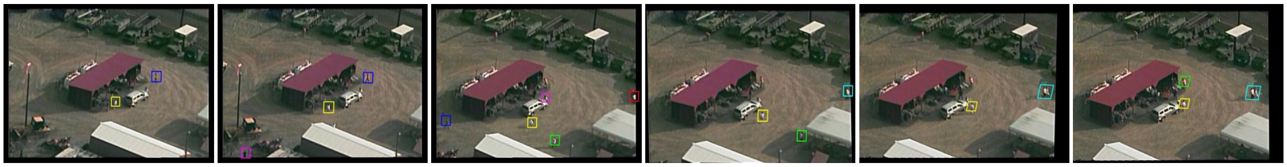


Fig. 18: Multi-target tracking results of a scene (frame no. 4, 59, 258, 385, 445, and 485) of the VIRAT aerial video dataset. Different targets are given different colors and IDs.

Figures 16, 17, and 18 show the tracking results of three scenes of the VIRAT aerial video dataset. There are a lot challenges present in these scenes, such as varying number of targets over the time, occlusion by other targets, occlusion by building, tiny targets, varying target velocities, different target poses, etc. Targets are engaging in different kinds of activities such as walking, running, standing, working, etc. In most of the cases, our method is able to track all the targets successfully for a longer period of time.

VI. Discussions and Conclusion

In this paper, we presented a multi-target tracking method that simultaneously uses the advantages of motion region detection and target classification to robustly detect and track moving targets, such as humans and vehicles, in unconstrained aerial videos. We divided the whole video into a number of equal length segments and stabilized each of the unstable video segment separately using homography transformation. We detected motion regions in the stabilized videos. Around these motion regions we searched for the target of interest. We devised a finite state machine for target tracking that utilized the Kalman filter and target detection in an unified framework. We associated the tracklets in two stages. First we associated the tracklets inside s video segment and then, we associated the tracklets between the video segments. Finally, we reprojected the tracks from the stabilized video frame to the original plane. We compared our method with state-of-the-art tracking algorithms and showed that our method outperforms other algorithms in most of the scenes.

References

[1] Y. Zhanfeng, L. N. Pramod, and T. Pankaj, “Improved target tracking in aerial video using particle filtering,” *SPIE*, vol. 7307, 2009. 1, 4

[2] C. Xianbin, L. Renjun, Y. Pingkun, and L. Xuelong, “Visual attention accelerated vehicle detection in low-altitude airborne video of urban environment,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 366–379, 2012. 1, 4

[3] K. Jinman, C. Isaac, M. Gerard, and Y. Chang, “Detection and tracking of moving objects from a moving platform in presence of strong parallax,” *ICCV*, 2005. 1, 4

[4] R. Vladimir, S. Berkan, and M. Shah, “Geometric constraints for human detection in aerial imagery,” *ECCV*, 2010. 1, 4

[5] P. Jan, Z. Xuemei, and M. Gerard, “Tracking many vehicles in wide area aerial surveillance,” *WASA*, 2012. 1, 4

[6] S. Ali, V. Reilly, and M. Shah, “Motion and appearance contexts for tracking and re-acquiring targets in aerial video,” *CVPR*, 2007. 1

[7] Virat aerial video dataset. [Online]. Available: http://www.viratdata.org/products/archive/VIRAT_Video_Dataset_Download_Instruction_Release2.pdf 2, 11

[8] Vivid aerial video dataset. [Online]. Available: http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html 2, 11

[9] H. Bay, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features.” *ECCV*, 2006. 2, 3

[10] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Comm. of the ACM*, vol. 24, pp. 381–395, 1981. 2

[11] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Journal of Computing Surveys*, vol. 38, no. 4, 2006. 3

[12] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, “Recent advances and trends in visual tracking: A review.” *Neurocomputing*, vol. 74, pp. 3823–3831, 2011. 3

[13] V. Ferrari, T. Tuytelaars, and L. V. Gool, “Object detection by contour segment networks,” *ECCV*, 2006. 3, 4

[14] B. Wu and R. Nevatia, “Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors,” *ICCV*, 2005. 3

[15] D. Lowe, “Object recognition from local scale-invariant features.” *ICCV*, 1999. 3

[16] N. Dalal and B. Trigges, “Histograms of oriented gradients for human detection,” *CVPR*, 2005. 3

[17] K. Shafique and M. Shah, “A non-iterative greedy algorithm for multi-frame point correspondence.” *ICCV*, 2003. 3

[18] Y. Bar-Shalon and T. Fortmann, “Tracking and data association.” *Academic Press.*, 1988. 3

[19] S. Maskell and N. Gordon, “A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking.” *Target Tracking: Algorithms and Applications IEEE Workshop*, 2001. 3

[20] P. Pan and D. Schonfeld, “Video tracking based on sequential particle filtering on graphs.” *IEEE TIP.*, vol. 20, pp. 1641–1651, 2011. 3

[21] C. Rasmussen and G. D. Hager, “Probabilistic data association methods for tracking complex visual objects.” *IEEE TPAMI*, vol. 23, pp. 560–576, 2001. 3

[22] C. Hue, J. L. Cadre, , and P. Prez, “Sequential monte carlo methods for multiple targettracking and data fusion.” *IEEE Trans. Sign. Process*, vol. 50, pp. 309–325, 2002. 3

[23] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo, “Visual tracking via incremental log-Euclidean Riemannian subspace learning.” *CVPR*, 2008. 4

[24] Y. Wu, J. Cheng, J. Wang, and H. Lu, “Real-time visual tracking via incremental covariance tensor learning.” *ICCV*, 2009. 4

[25] B. Babenko, M.-H. Yang, and S. Belongie, “Visual tracking with online multiple instance learning.” *CVPR*, 2009. 4, 12

[26] H. Grabne, M. Grabner, and H. Bischof, “Real-time tracking via on-line boosting.” *BMVC*, pp. 47–56, 2006. 4, 12

[27] C. Aeschliman, J. Park, and A. Kak, “A probabilistic framework for joint segmentation and tracking.” *CVPR*, 2010. 4

[28] S. Stalder, H. Grabner, and L. Gool, “Cascaded confidence filtering for improved tracking-by-detection.” *ECCV*, 2010. 4

[29] M. Yang, Y. Wu, and S. Lao, “Intelligent collaborative tracking by mining auxiliary objects.” *CVPR*, 2006. 4

[30] W. Hu, X. Zhou, W. Li, W. Luo, X. Zhang, and S. Maybank, “Active contour-based visual tracking by integrating colors, shapes, and motions.” *IEEE TIP.*, vol. 22, pp. 1778–1792, 2013. 4

[31] J. Fan, X. Shen, and Y. Wu, “What are we tracking: A unified approach of tracking and recognition.” *IEEE TIP.*, vol. 22, pp. 549–560, 2013. 4

[32] C. Huang, Y. Li, and R. Nevatia, “Multiple target tracking by learning-based hierarchical association of detection responses.” *IEEE TPAMI*, vol. 35, pp. 898–910, 2013. 4

[33] A. Butt and R. Collins, “Multi-target tracking by lagrangian relaxation to min-cost network flow.” *CVPR*, 2013. 4

[34] O. Omar, L. , Xin, and S. Mubarak, “Simultaneous video stabilization and moving object detection in turbulence,” *IEEE TPAMI*, vol. 35, pp. 450–462, 2013. 4

[35] M. Piccardi, “Background subtraction techniques: A review.” *IEEE Int. Conf. Syst., Man Cybern.*, vol. 4, pp. 3099–3104, 2004. 5

[36] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features.” *CVPR*, pp. 511–518, 2001. 6

[37] R. Lienhart and J. Maydt, “An extended set of Haar-like features for rapid object detection.” *ICIP*, 2002. 6