# Map-Enhanced Detection and Tracking from a Moving Platform with Local and Global Data Association

Qian Yu, and Gérard Medioni
Institute for Robotics and Intelligent Systems
University of Southern California
{qianyu, medioni}@usc.edu

## Abstract

*We present an approach to detect and track moving objects from a moving platform. Moreover, given a global map, such as a satellite image, our approach can locate and track the targets in geo-coordinates, namely longitude and latitude. The map information is used as a global constraint for compensating the camera motion, which is critical for motion detection on a moving platform. In addition, by projecting the targets' position to a global map, tracking is performed in coordinates with physical meaning and thus the motion model is more meaningful than tracking in image coordinate. In a real scenario, targets can leave the field of view or be occluded. Thus we address tracking as a data association problem at the local and global levels. At the local level, the moving image blobs, provided from the motion detection, are associated into tracklets by a MCMC (Markov Chain Monte Carlo) Data Association algorithm. Both motion and appearance likelihood are considered when local data association is performed. Then, at the global level, tracklets are linked by their appearance and spatio-temporal consistence on the global map. Experiments show that our method can deal with long term occlusion and segmented tracks even when targets leave the field of view.*

## 1 Introduction

One of the goals in video surveillance is to identify and track all the relevant moving objects in the scene, and to generate exactly one track per object. This may involve detecting the moving objects, tracking them while they are visible, and re-acquiring the objects once they emerge from an occlusion to maintain identity. This is a very difficult problem, even more so when the sensor is moving, as in aerial surveillance scenarios. To track from a moving camera, we need to project targets at different times into a common reference frame. Accumulated errors are introduced when fixed coordinates are selected and no further alignment is performed. Usually the first frame [4] or the ground plane in the first frame [6] is selected as the reference frame. Moreover, due to scale change, image coordinates of the targets are not meaningful. Here, we propose to use a global map (a satellite image) as the reference frame. By registering UAV (Unmanned Aerial Vehicles) images with the satellite image, we can generate the absolute geo-location of targets. Also, tracking is performed in geo-coordinates, which have clear physical meaning.

In surveillance applications, occlusion is common. We introduce a two-step procedure for tracking with occlusion. The first step (called local association) links detected regions within a sliding window and generates tracklets. The second step (called global association) links the tracklets to form longer tracks and maintain tracks ID.

Local association is essential for successful tracking since errors in local association are not rectified in the global one. We formulate the local association as multiple targets tracking, in which the purpose is to find the best partition of observation (*i.e.* detected moving regions) graph. In the global association, by assuming the maximum speed and acceleration of targets on the geo-coordinates, we can define the compatibility of tracklets and this reduces ambiguity in tracklet association. In ad-

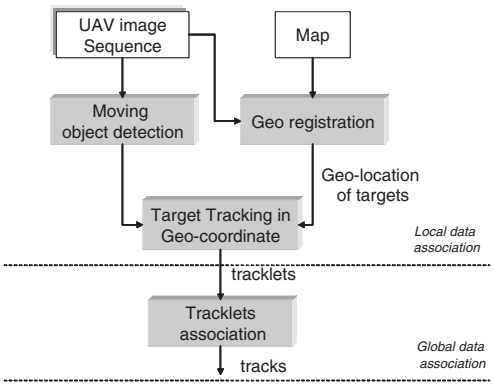



Figure 1: Overview of the tracking framework.

dition, we adopt rotation invariant appearance descriptors [5] to represent both color and shape distribution of targets in each tracklets. The flowchart of our framework is shown in Figure 1.

The paper is organized as follows: we introduce the geo-registration and tracking coordinates in section 2; the method to detect moving regions from a moving camera is presented in section 3. The local data association algorithm and global tracklets association in sections 4 and 5. Section 6 provides experimental results on real UAV data set.

### 1.1 Related work

Digital elevation model (DEM) is usually given in 3D geo-registration approach [3, 2]. Kumar in [3] proposed a coarse to fine approach for geo-registration. The coarse initialization is implemented with local appearance matching using normalized correlation. The fine geo-registration is acquired by estimated the projection matrix of camera given DEM. Here we assume the scene is planar and only 2D geo-registration is considered. The idea of tracklet was proposed in [6], in which a simple single target tracker is used and the ground plane in the first frame is used as the common coordinates in tracking. Recently in [9], the authors introduced a MCMC based sampling method to address the association of punctual observations. The posterior distribution assume a *prior* knowledge on the detection and the targets' behavior and consider only dynamics likelihood.

## 2 Geo-Registration

In our UAV environment, we assume the scene can be approximated as a plane. This assumption is reasonable when the structure on the ground plane is relatively small compared with the camera height to ground plane. We use a map $M$ to represent the ground plane. Making this planar assumption, the transformation between a UAV image and the map can be represented as a homography, $H_{iM}$, namely $H_{iM}I_i = M$. The transformation between two UAV images $I_i$ and $I_j$ can be represented as $H_{ij}$, *i.e.* $H_{ij}I_i = I_j$. For the tracking task, we need to define the targets' state (position, velocity, dimension, etc.) of different times in a common reference frame, where we can introduce the motion model of targets, such as constant velocity motion. The first frame could be selected as the reference coordinates. However accumulated errors are then introduced since the $H_{i0} = H_{i(i-1)} \cdots H_{10}$ need to be computed. Here we use the map $M$ as the common coordinates, and each UAV image $I_i$ is registered with $M$ with the homography $H_{iM}$. The accumulated error is reduced by registering the UAV images with the global map. In addition, it makes more sense to define the motion kinematics since the geo-coordinates have physical meaning.

To compute the homography $H_{iM}$, we propose a two-step procedure to register a UAV image sequence to the global map. In the first step, we compute $H_{i(i+1)}$ register consecutive frames using RANSAC to estimate the best homography to align the feature points in each frame. Given the homography between two consecutive frames, the homography $H_{i,j}$ between any two frames can be represented as follows.

$$H_{ij} = \begin{cases} \prod_{k=i}^{j-1} H_{k,k+1} & i < j \\ I & i = j \\ (H_{j,i})^{-1} & i > j \end{cases} \quad (1)$$

Given an initialization $H_{0M}$ and the homography $H_{i0}$ obtained from Eq.1, we roughly align the UAV image with the global map. In the second step, we refine the registration by computing the homography between roughly aligned UAV image and the map. Since UAV images are captured at different times and in different views with the satellite image, the color, illumination, and the dynamic content (such as vehicles, trees and shadow and so on) could be very different. Thus to find the correspondence

from such two images, we apply mutual information [12] to establish patch-based correspondence.

Given the correspondence between the roughly aligned UAV image and the map, again we apply RANSAC to compute a refined homography. By linking the refined homography and the initialized homography from the first step, we can register the UAV image with the map without incremental accumulated registration error. The detail of this Geo-registration method can be found in [8]. Figure 2 shows 2000 geo-mosaiced frames overlayed on top of the map. We can see even after 2000 frames, the registration is still maintained within a small error bound.

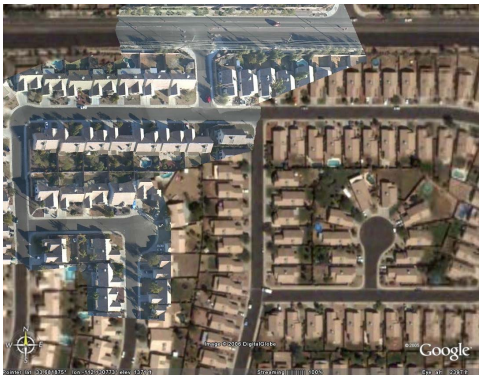



Figure 2: Geo-mosaicing 2000 consecutive frames on top of the reference frame.

# 3 Detecting Moving Regions

In a stationary camera, motion in the image sequence is modeled at the pixel level and a background model is defined for each pixel using statistical-based techniques. The concept of background model can be extended to non-stationary cameras by compensating for the camera motion before estimating of the background model. We adopt the sliding window method in [4]. The sliding window contains $W = 2w + 1$ frame. We typically use $W = 91$. The center frame $I_c$ of the sliding window is selected as the reference frame. The homographies between consecutive frames can be concatenated to register the current frame to the selected frame in 1. Every frames in the sliding window is registered to the reference frame using $H_{i,c}$. For each pixel $p$ on the reference frame, the background model is established by computing the mode of a histogram (size of $W$) $\{I_i(H_{c,i}p)\}, i \in [c-w, c+w]$. Then The moving region in the reference frame can be acquired by thresholding the residual image against background. The sliding window method keeps the accumulation within the half size of sliding window. Erroneous registrations do not influence the quality of the motion detection for the whole sequence but only the frames within the sliding window.

# 4 Local Data Association

Given a set of observations $Y$ over time $T$, the local data association problem is formulated as maximizing a posterior (MAP) of a partition $\omega = \{\tau_0, \tau_1, \tau_2, ..., \tau_K\}$ such that:

$$\omega^* = \arg\max(p(\omega|Y)) \tag{2}$$

where $\tau_0$ is the set of false alarms, $\tau_k$ is the track $k$ among $K$ tracks from the given partition. We use a graph representation of all measurements within the time frame $[0, T]$. Let $y_t = \{y_t^i : i = 1, ..., n_t\}$ denote the observations at time $t$, $Y = \cup_{t\in\{1,...,T\}} y_t$ is the set of all the observations during $[0, T]$. The partition can be explicitly drawn from this measurement graph $(V, E)$, where each measurement $y_t^i$ is represented by a node in $V$, and each edge corresponds to a temporal association reflecting spatial properties such as spatial overlap between detected regions. We define a neighborhood in the graph $(V, E)$ where edges are defined between any two neighboring nodes:

$$N = \{(y_{t_1}^i, y_{t_2}^j) : \left\| y_{t_1}^i - y_{t_2}^j \right\| < t \cdot v_{max})\} \tag{3}$$

where $v_{max}$ is the maximum speed of targets.

The posterior distribution for the partition with unknown number of targets and observations over $T$ frames can be modeled as:

$$P(\omega|Y) \propto \prod_{k=1}^{K} \psi(\tau_k) \prod_{j\neq k} \phi(\tau_k, \tau_j) \tag{4}$$

where $\psi(\tau_k)$is the temporal compatibility within one track, and $\phi(\tau_k, \tau_j)$ is the spatial compatibility between different tracks respectively. The posterior distribution in Eq. 4 can be viewed as having two distinct components: (i) $\psi(\tau_k)$ controls the inner-smoothness for each track encoded by the joint motion and appearance likelihood (ii) $\phi(\tau_k, \tau_j)$ encodes the interaction between different tracks. We will now discuss each one of these in turn.

## 4.1 Motion and Appearance Model

Here targets are represented by image blobs. Once a partition $\omega$ is chosen, the tracks $\{\tau_1, ..., \tau_K\}$ and false alarms $\tau_0$ are determined and for each track the assigned observations are determined.

To make full use of the observations for target tracking, we consider a joint probability framework for incorporating both motion and appearance information. Therefore $\psi(\tau_k)$ in Eq. 4 can be represented as follows.

$$\psi(\tau_k) = \prod_{l=1}^{|\tau_k|-1} P_{\text{mot}}(\tau_k(t_{l+1})|\bar{\tau}_k(t_l)) P_{\text{app}}(\tau_k(t_{l+1})|t_l)) \quad (5)$$

Given the geo-registration result, we can map an image blob from UAV image to the map. We denote $x_t^k$ the state vector of the target $k$ at time $t$ to be $\left[l_x, l_y, w, h, \dot{l}_x, \dot{l}_y\right]$ (centroid's position, width, height and velocity in the 2D map). We consider a linear kinematic model of constant velocity dynamics:

$$x_{t+1}^k = A^k x_t^k + w^k \quad (6)$$

where $A^k$ is the transition matrix, and we assume $w^k$ to be a normal probability distribution, $w^k \sim N(0, Q^k)$. The observation $y_t^k = [l_x, l_y, w, h]$ contains the measurement of a target position and size in 2D map. Since observations often contain false alarms, the observation model is represented as:

$$y_t^k = \begin{cases} H^k x_t^k + v^k & \text{if it belongs to a target} \\ \delta_t & \text{false alarm} \end{cases} \quad (7)$$

where $y_t^k$ represents the measurement which may arise either from a false alarm or from the target. We assume $v^k$ to be normal probability distributions, $v^k \sim N(0, R^k)$. $\delta_t$ is a 2D random variable with uniform distribution on the map.

Let $\hat{\tau}_k(t_i)$ and $\hat{P}_t(\tau_k)$ denote the posterior estimated states (i.e. $x_k^t$ in Eq.6) and posterior covariance matrix of the estimated error at time $t$ of $\tau_k$. $\tau_k(t)$ is the associated observation (i.e. $y_k^t$ in Eq.7) for track $k$ at time $t$. The motion likelihood of track $\tau_k$ of one edge $(\tau_k(t_1), \tau_k(t_2)) \in E, t_1 < t_2$ can be represented as $P_{\text{motion}}(\tau_k(t_2)|\hat{\tau}_k(t_1))$. Given the transition and observation model in a Kalman filter, the motion likelihood then can be written as:

$$P_{\text{mot}}(\cdot) = \frac{1}{(2\pi)^2 \det(\hat{P}_{t_2}(\tau_k))} \exp\left(\frac{-e^T \hat{P}_{t_2}^{-1}(\tau_k) e}{2}\right) \quad (8)$$

where $e = \tau_k(t_2) - HA^{t_2-t_1}\hat{\tau}_k(t_1)$ and $\hat{P}_{t_2}(\tau_k)$ can be computed recursively by a Kalman filter as $\hat{P}_{t_2}(\tau_k) = H(A\hat{P}_{t_2-1}(\tau_k)A^T + Q)H^T + R$.

In order to model the appearance of each detected region, we adopt a histogram-based appearance of the image blobs. All RGB bins are concatenated to form a one-dimension histogram. The appearance likelihood between two connected image blobs $(\tau_k(t_1), \tau_k(t_2)) \in E, t_1 < t_2$ in track $k$, is measured using the symmetric Kullback-Leibler Distance (KL) is defined as follows, where $P(c)$ is the bin value of normalized histogram.

$$P_{\text{app}}(\cdot) = \exp(\frac{1}{2} \sum_{c=r,g,b} (P_i(c) - P_j(c)) \log(\frac{P_i(c)}{P_j(c)})) \quad (9)$$

## 4.2 Interaction model

The motion and appearance likelihood models provide the inner-smoothness constraint for each track independently. However, without an *a priori* knowledge of the number of targets, the inner-smoothness constraint favors shorter paths, and therefore tends to split a trajectory into a large number of sub-tracks. To overcome this overfitting problem, commonly a *prior* knowledge on the detection and the targets' behavior (such as detection and false alarm rate, termination and birth rate etc.) is assumed known [9, 1].

We propose to use an interaction model that penalizes object overlapping based on Markov random fields (MRFs) [7, 11] defined on the neighborhood graph. The joint interaction between all existing nodes over time is factored as the product of local potential functions at each node. In this MRF, the cliques are pairs of nodes that are connected in the graph $(V, E)$. The interaction potential between $\tau_k$ and $\tau_j$ is defined by:

$$\phi(\tau_k, \tau_j) = \prod_{l=1}^{|\tau_k-1|} \prod_{m=1}^{|\tau_j-1|} \exp(-\lambda \rho(\tau_k(l), \tau_j(m)) \quad (10)$$

where $\rho$ is the spatial overlap between two observation nodes. The interacting potential is minimum when the observations have a large spatial overlap and maximum when they do not overlap. The introduction of the inter-track exclusion will prevent from splitting tracks into smaller tracks when a good overlapping of the regions exists.

### 4.3 MCMC Data association Algorithm

We use a data-driven MCMC for estimating the best partition of the space $\Omega$. The sampling is guided by the posterior distribution defined in Eq. 4. Here the sampling is similar in [9]. The difference is that we propose to drive the sampler, in a probabilistic manner, using both motion and appearance likelihoods. Moreover, in order to make the sampler more efficient, we draw samples in both temporal directions: looking forward and backward in time. This bidirectional sampling gives more flexibility and reduces significantly the total number of samples in terms of convergence.

We use the following notations on the graph structure: $N(\cdot)$ is the neighbor set of an observation, i.e. $N(y^i_{t_1}) = \{y^j_{t_2}, (y^i_{t_1}, y^j_{t_2}) \in E\}$; Observation $y^j_{t_2} \in N(y^i_{t_1})$ belongs to the parent set $N^c(y^i_{t_1})$, child set $N^p(y^i_{t_1})$ exclusively, when $t_2 < t_1$ or $t_2 > t_1$.

**Extension/Reduction:** The purpose of the extension/reduction move is to extend or shorten the estimated trajectories given a new set of observations. For the forward extension, we select uniformly at random (*u.a.r*) a track $\tau_k$ from $K$ available tracks, $\tau_1, ..., \tau_K$. Let $\tau_k(end)$ denote the last node in the track $\tau_k$. For each node $y \in N^c(\tau_k(end))$, we have the association probability $p(y) = \frac{p(y|\bar{\tau}_k(end))}{\sum_z p(y|\bar{\tau}_k(end))}$. We associate $y$ and track $\tau_k$ according to this normalized probability, and then append the new observation $y$ to $\tau_k$ with a probability $\gamma$, where $0 < \gamma < 1$. Similarly, for a backward extension, we consider a node $y \in N^p(\tau_k(start))$ and use reverse dynamics for estimating the association probability $p(y)$.

The reduction move consists of randomly shortening a track $\tau_k$ (*u.a.r* from $K$ available tracks, $\tau_1, ..., \tau_K$), by selecting a cutting index $r$ *u.a.r* from $2, ..., |\tau_k| - 1$. In the case of a forward reduction the track $\tau_k$ is shortened to $\{\tau_k(t_1), ..., \tau_k(t_r)\}$, while in a backward reduction we consider the sub-track $\{\tau_k(t_r), ..., \tau_k(t_{|\tau_k|})\}$.

**Birth/Death:** This move controls the creation of new track or termination of an existing trajectory. In a birth move, we select *u.a.r* a node $y \in \tau_0$, associate it to a new track and increase the number of tracks $K' = K + 1$.

The birth move is always followed by a extension move. From the node $y$ we select the extension direction forward or backward *u.a.r* to extend the track $\tau_{K'}$. Similarly, in a death move we choose *u.a.r* a track $\tau_k$ and delete it. The nodes belonging to the deleted track are added to the unassigned set of measurements $\tau_0$.

**Split/Merge:** In a split move, we *u.a.r* select a track $\tau_k$, and a split point $t_s$, which is selected according to the normalized joint probability between two consecutive connected nodes in the track: $(1 - p(\tau_k(t_{i+1})|\bar{\tau}_k(t_i))) \,/\, \sum_{i=1}^{|\tau_k|-1} (1 - p(\tau_k(t_{i+1})|\bar{\tau}_k(t_i)))$. And we split $\tau_k$ into two new tracks $\tau_{s_1} = \{\tau(t_1),...,\tau(t_s)\}$ and $\tau_{s_2} = \{\tau(t_{s+1}),...,\tau(t_{|\tau_k|})\}$.

Often, due to missing detection or erroneous detection, trajectories of objects are fragmented. The merge move provides the ability to link these fragmented sub-tracks according to their joint likelihood of appearance and motion and the interaction based on spatial overlapping. The merge move operates on the candidate set of track pairs, for which the start node of one track is the child node of the end node of the other track and is defined by the set: $C^t_{merge} = \{(\tau_{k_1}, \tau_{k_2}) : \tau_{k_2}(start) \in N^c(\tau_{k_1}(end))\}$ . We select u.a.r pairs of tracks from $C^t_{merge}$ and merge the two tracks into a new track $\tau_k = \{\tau_{k_1}\} \cup \{\tau_{k_2}\}$

**Switch:** In a switch move, we are probing the solution space for better labeling of nodes that belong to multiple tracks. We consider the following candidate set of track pairs.

$$C^t_{switch} = \{(\tau_{k_1}(t_p), \tau_{k_2}(t_q)) : \tau_{k_1}(t_p) \in N^p(\tau_{k_2}(t_{q+1})), \tau_{k_2}(t_q) \in N^p(\tau_{k_1}(t_{p+1})), k_1 \neq k_2\}. \quad (11)$$

We *u.a.r* select a candidate node from $C^t_{switch}$ and define two new tracks as:
$\tau'_{k_1} = \{\tau_{k_1}(t_1), ..., \tau_{k_1}(t_p), \tau_{k_2}(t_{q+1}), ..., \tau_{k_2}(t_{|\tau_{k_2}|})\}$ and
$\tau'_{k_2} = \{\tau_{k_2}(t_1), ..., \tau_{k_2}(t_q), \tau_{k_1}(t_{p+1}), ..., \tau_{k_1}(t_{|\tau_{k_1}|})\}$.

**Online Processing:** The complexity of local data association depends on the size of the observation graph. Moreover, the algorithm is performed in a deferred logic way. The decision is made when all observations in the graph are available. Thus we implemented the proposed association algorithm as an online one within a sliding window which contains the latest 45 frames and only observations within this sliding window are stored in the measurement graph. When the sliding window moves, the partition of the graph at the previous time is used as initialization.

## 5 Global Tracklets Association

Although merge/split operation in local data association can deal with missing detection, local data association

only considers observations within a short time span. Some situations, such as long occlusions, may cause the tracker to lose target identification. Increasing the size of sliding window cannot solve the problem all the time and increases the complexity. Thus we introduce the global data association algorithm to associate tracklets to maintain track identification.

## 5.1 Spatio-temporal Consistency

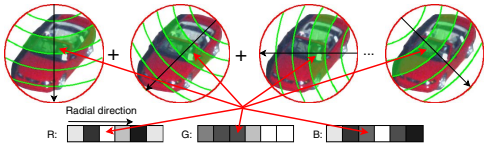
First we define the consistency of in temporal and spatial relationship between tracklets. Given two tracklets $\tau_1$ and $\tau_2$, which start at time $s_1$, $s_2$ and terminate at time $t_1$, $t_2$. If the condition $s_1 \geq t_2$ or $s_2 \geq t_1$ holds, the two tracklets are temporally consistent. For two temporally consistent tracklets $\tau_1$ and $\tau_2$, say $s_2 \geq t_1$, the terminating position and velocity of $\tau_1$ on the global map is $P_{t_1}$ and $V_{t_1}$ The starting position and velocity of $\tau_2$ on the global map is $P_{s_2}$ and $V_{s_2}$. If the $||P_{t_1} - P_{s_2}|| \leq v_{max} \times (s_2 - t_1)$ and $||V_{t_1} - V_{s_2}|| \leq a_{max} \times (s_2 - t_1)$, the two are spatially consistent as well, where $v_{max}$ and $a_{max}$ represent the maximum speed and acceleration of targets on the map.

## 5.2 Tracklet Descriptor

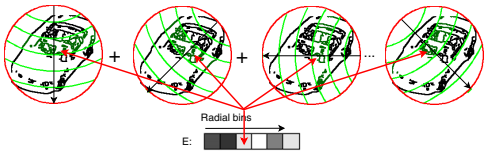
In order to associate the temporally and spatially consistent tracklets, we adopt the appearance model proposed in [5]. This descriptor is invariant to 2D rotation and scale change, and tolerates small shape variations. Instead of applying this descriptor on a single image blob, we use the descriptor on a tracklet, which contains a sequence of image blobs.

For each detected moving blob within a tracklet, the reference circle is defined as the smallest circle containing the blob. The reference circle is delineated as the 6 bin images in 8 directions depicted in Figure 3. For each bin $i$, a Gaussian color model is built on all the pixels located in bin $i$ for all 8 directions and for all image blobs within the tracklet. Thus the color model for each tracklet is then defined as a 6D vector by summing the contribution of each bin image in all 8 directions and for all image blobs. We can similarly encode the shape properties of each blob by using a uniform distribution of the number of edge pixels within each bin, namely a normalized vector $[E_1(\tau), E_2(\tau), \cdots, E_6(\tau)]$.

The appearance likelihood between two compatible




(a) Appearance Color Model




(b) Appearance Shape Model

Figure 3: Appearance descriptor of tracklets

tracklets can be defined:

$$p_{app}(\tau_i, \tau_j) = \exp\left(-\lambda \left(d_{color}(\tau_i, \tau_j) + d_{edge}(\tau_i, \tau_j)\right)\right) \tag{12}$$

where $\tau_i$, $\tau_j$ are tracklets on which the appearance probability model is defined.

The appearance distance between two compatible tracklets is computed using the Kullback-Leibler (KL) divergence. For the color descriptor, since each bin is modeled by a Gaussian model, the KL distance is reduced to:

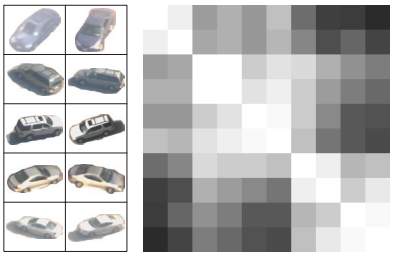
$$d(\tau_i, \tau_j) = \frac{1}{2N} \sum_{N} \left\{ (\mu_i - \mu_j)^2 \left( \frac{1}{\sigma_j^2} + \frac{1}{\sigma_i^2} \right) + \frac{\sigma_i^2}{\sigma_j^2} + \frac{\sigma_j^2}{\sigma_i^2} \right\} \tag{13}$$

where $\mu_i$, $\mu_j$, $\sigma_i$ and $\sigma_j$ are the parameters of the color Gaussian model. For the edge descriptor we use the following similarity measure:

$$d_{edge}(\tau_i, \tau_j) = \frac{1}{2} \sum_{r=1}^{6} (E_r(\tau_i) - E_r(\tau_j)) \log \frac{E_r(\tau_i)}{E_r(\tau_j)} \tag{14}$$

In the global association, two compatible tracklets will be assigned the same ID if the distance between the two tracklets' appearance is smaller than a threshold. Due to the existence of both target motion and camera motion, the target's orientation could be quite different in different tracklets, thus the rotation-invariant property of the descriptor is quite important for our tracklets association. In Figure 4, we show the confusion matrix of the several tracklets. From the confusion matrix the rotation-invariant descriptor works very well when tracklets undergo obvious rotation. In addition, illumination may vary

for tracklets acquired at different time. Since the appearance considers the edge information, the appearance can partially deal with illumination changing.



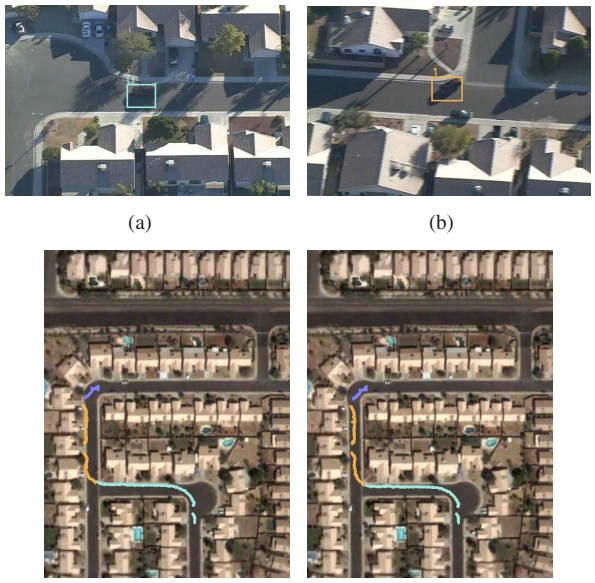
(a) (b)

Figure 4: (a) represents the first blob of different tracklets (b) the confusion matrix of tracklets

## 6 Experimental Results

We show the tracking results on the following two UAV sequences. Using the longitude and latitude information coming with image sequences, the map is acquired from Google Earth. The homography between the first frame and the map $H_{0M}$ is manually selected offline. Figure 5 shows the tracking result on a sequence with one moving object. Considering the computation cost, the geo-registration refinement with the map is performed every 50 frames. Figure 5(c) and Figure 5(d) display the tracking result on the map. The trajectory of tracklets in Figure 5(c) is generated using the initial homography between UAV image and map without refinement. Figure 5(d) is generated using our geo-registration. It is clear that the trajectories of tracklets without geo-registration are out of the road boundary. Since the target is fully occluded by the shadows of trees, the trajectory of the single target breaks into tracklets. In real scenarios, the moving shadow may affect the target's appearance. We apply the deterministic nonmodel-based method [10] working in HSV space to remove the strong moving shadow. However, due to the noisy moving shadow removal, the target identity is not fully maintained.

Figure 6 shows the tracking result on the sequence with multiple moving targets. Again when targets are occluded by shadows, local data association may lose the track identification and thus tracklets are formed. The missing



(a) (b)

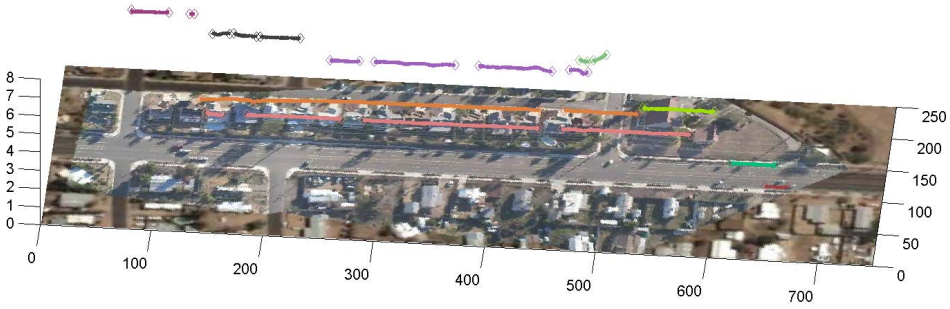
(c) Trajectory of tracklets without geo-registration (d) Trajectory of tracklets with geo-registration

Figure 5: Comparison of with and without geo-registratoin

detection caused by occlusion even lasts for longer than the sliding window of local data association (45 frames). However in global data association, the tracklets are associated with correct ID throughout the video. The different tracks are listed in the $Z$ direction in different colors. Figure 6(b), 6(c), 6(d) and 6(e) show the beginning frame of the tracklets of the red truck. Although the appearance of the white van and the white SUV in 6(b) is quite similar, the temporal and spatial constraint on the global map prevents from associating them together.

## 7 Conclusions and future work

We have proposed a framework to detect and track moving objects from a moving platform. The geo-registration with a global map provides us reference coordinates to geo-locate targets with physical meaning. In geo-coordinates, correlation between tracklets produced in the local data association algorithm is evaluated using spatio-temporal consistency and similarity of appearance. Ex-




(a) The tracking result with geo-mosaicing the UAV images on the satellite image

(b) (c) (d) (e)

Figure 6: The tracklets and tracks obtained using the local and global data association framework. The UAV image sequence is overlayed on top of the satellite image.

periments show the local and global association can maintain the track ID across long term occlusion.

In the future, we expect to use a discriminative combination of local features to reacquire targets against long occlusions. Also, we will investigate the scene understanding using the map to reduce false alarms caused by noisy motion detection.

## Acknowledgements

This work was supported by grants from Lockheed Martin and MURI-ARO W911NF-06-1-0094. We thank Mark Pritt for providing the data.

## References

[1] Y. Bar-Shalom, T. Fortmann, and M. Scheffe. Joint probabilistic data association for multiple targets in clutter. In *Proc. Conf. on Information Sciences and Systems*, 1980.

[2] R. W. Cannata, S. G. Blask, J. A. V. Workum, and M. Shah. Autonomous video registration using sensor model parameter adjustments. In *AIPR: Proceedings of 29th Applied Imagery Pattern Recognition Workshop*, page 215, 2000.

[3] K. Hanna, H. Sawhney, R. Kumar, Y. Guo, and S. Samarasekara. Annotation of video by alignment to reference imagery. In *ICCV '99*, pages 253–264, 1999.

[4] J. Kang, I. Cohen, and G. Medioni. Continuous tracking within and across camera streams. In *CVPR*, volume 1, pages 267–272, Jun 2003.

[5] J. Kang, I. Cohen, and G. Medioni. Object reacquisition using invariant appearance model. In *ICPR*, pages 759–762, 2004.

[6] R. Kaucic, A. G. A. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs. A unified framework for tracking through occlusions and across sensor gaps. In *CVPR '05 Volume 1*, pages 990–997. IEEE Computer Society, 2005.

[7] Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE PAMI*, (11):1805–1918, 2005.

[8] Y. Lin, Q. Yu, and G. Medioni. Map-enhanced uav image sequence registration. In *IEEE Workshop on Applications of Computer Vision (WACV)*, 2007.

[9] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, 2004.

[10] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara. Detecting moving shadows: algorithms and evaluation. 25(7):918–923, July 2003.

[11] K. Smith, D. Gatica-Perez, and J.-M. Odobez. Using particles to track varying numbers of interacting people. In *CVPR*, pages 962–969, 2005.

[12] P. Viola. Alignment by maximization of mutual information. phd thesis mit. 1995.