Enhanced Target Tracking in UAV Imagery with P-N Learning and Structural Constraints

Mennatullah Siam
Nile University, Center of Informatics Science
B2 Smart Village, Cairo, Egypt
menna.siam@nileu.edu.eg

Mohamed ElHelw
Nile University, Center of Informatics Science
B2 Smart Village, Cairo, Egypt
melhelw@nileuniversity.edu.eg

Abstract

This paper presents improved automatic moving target detection and tracking framework that is suitable for UAV imagery. The framework is comprised of motion compensation phase to detect moving targets from a moving camera, target state estimation with Kalman filter, and overlap-rate-based data association. Finally, P-N learning is used to maintain target appearance by utilizing novel structural constraints to select positive and negative samples, where data association decisions are used as positive (P) constraints. After learning target appearance, a cascaded classifier is employed to detect the target in case of association failure. The proposed framework enables to recapture targets after being out of camera field of view and helps discriminating between targets with similar appearance while alleviating drift problems. Experimental results obtained with publicly available DARPA aerial datasets demonstrate that the proposed tracker with automatic detection feedback achieves better recall and average overlap than existing manually-initialized trackers.

1. Introduction

Unmanned Aerial Vehicles (UAVs) are becoming increasingly vital for a variety of applications in urban and non-urban settings such as aerial surveillance /reconnaissance, search and rescue, traffic monitoring, to name a few. In order to achieve the goal of autonomous UAV operation, algorithms for automatic target detection and tracking are deployed on aerial platforms. Nevertheless, processing of UAV imagery is challenging due to several factors such as rapid camera motion and low resolution images captured from high altitude vehicles making it difficult to discriminate similar targets. Additional challenges stem from the fact that targets may undergo significant appearance changes and they can also go in and out of camera field of view.

Plethora of work has been conducted specifically on target detection and tracking in UAV imagery. In [8][9][10][11][12] motion compensation was used to separate target motion from scene motion using image registration. In [14] a particle filter was used for target tracking to estimate the state of a tracked target. Instead of manual initialization of the targets to be tracked, in[12] an automatic detection of moving targets was presented. It was followed by applying a Kalman filter to track each target. In [13] an adaptive tracker for manually labeled targets was used to select the most discriminative features from color and shape-texture cues.

Further work on target tracking not concerned with aerial imagery can also be found in the literature. For instance, the basic mean shift technique [27] is mainly based on a color model and fails with significant appearance changes due illumination or viewpoint changes. In [26], two approaches are introduced for selecting discriminative features online: the variance ratio between foreground and background classes and the peak difference. The former is used to rank the features and fails with cluttered environments. The latter favors features that minimize the effect of distraction due to clutter. In [28], a static model is used to represent the object where its patch is represented as multiple image fragments. It is robust in case of partial occlusions, but doesn't learn the target's appearance.

Some of the challenges in the processing of aerial imagery include rapid camera motion, target going out of the camera field of view, and targets with similar appearance that are difficult to discriminate. To solve the aforementioned problems, techniques for learning target appearance throughout the aerial video are used. Semi-supervised learning techniques are one family of algorithms that utilize both labeled and unlabeled data to accommodate changes in target appearance. In self learning [1][5][6][7][29], an initial classifier is trained using labeled data, then evaluated on unlabeled data and the most confident examples are augmented to the training set of the classifier. In [29], an online semi-supervised learning algorithm was presented and it was observed that the performance got better when another criterion was used independent on the classifier confidence like the PN constraints introduced later. In [30] multiple instance learning(MIL) was introduced where the idea was to present the examples as bags instead of instances. Still, the update strategy caused tracker drift in some cases. In

[2][4], P-N Learning is introduced which is an approach for utilizing the structure in the unlabeled data was presented. This work is the main inspiration for the proposed framework. It used positive and negative constraints to restrict the labeling of unlabeled examples. In this case, the positive constraint is the one used to identify samples labeled wrongly as negative while the negative constraint identified samples labeled wrongly as positive. The constraints used were based on the median flow tracker [3], where the positive constraint required all patches close to the target's trajectory to be positive. But, as will be shown in the results section, this caused drift issues. It was suggested in [4] to use motion detection techniques to enhance the performance of the tracker.

This paper proposes automatic moving target detection and tracking framework as an extension on our previous work in[15][16]. The proposed extension uses novel P-N structural constraints extracted from efficient overlap-rate-based data association [12] are used to maintain target appearance where data association decisions are used as positive (P) constraints. This enables the tracker to recapture the target after being out of the camera field-of-view for several consecutive frames. The tracker is able to discriminate between nearby targets with similar appearances. The performance of the framework supersedes the original P-N algorithm [2] and alleviates the drift issues by using automatic feedback from moving target detection. Furthermore, compared to existing target tracking approaches that are adaptive to the target's appearance, such as in [13], the proposed framework provides better performance with experiments applied to the DARPA datasets where the former failed in partial occlusions and in discriminating similar targets. The paper is organized as follows: Section 2 provides an overview of the framework architecture, Section 3 describes the target detection phase, Section 4 discusses target tracking and data association, and Section 5 describes the P-N learning approach and the newly developed constraints. Section 6 presents experimental results obtained while Section 7 concludes the paper.

2. Framework Architecture

The steps involved in the proposed moving target detection and tracking framework are as follows: (1)moving target detection (2)overlap-rate based data association, (3)Kalman tracking, (4)target localization using cascaded classifier comprised of patch variance, ensemble classifier and Parzen window, and (5)learning target appearance using P-N learning. The framework combines both detection and tracking of moving targets for improved accuracy. Figure 1 illustrates the architecture of the framework.

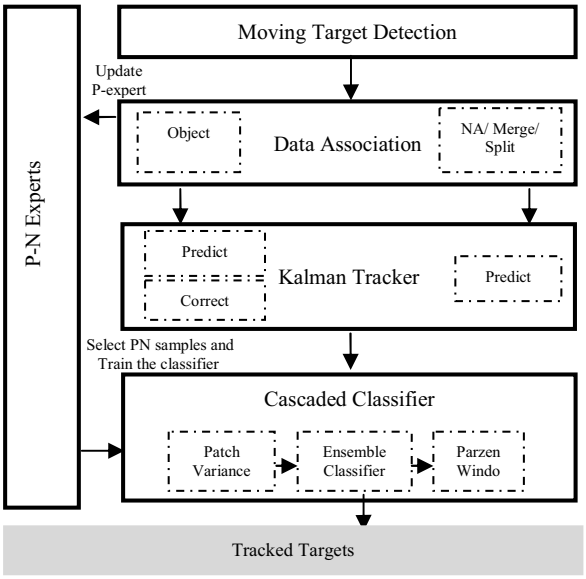

Figure 1: Block Diagram of target detection and tracking famework

3. Moving Target Detection

The first phase in the framework detects moving targets by clustering the outlier features that don't conform to scene motion. The output clusters represent potential targets that have to be checked first for false positives. The details of the steps of the proposed detection algorithm are discussed below.

3.1. Homography Computation

First step is to extract FAST features[17], which is performed on two stages. First, the segment criterion test is performed, which searches for candidate corners. A candidate corner has neighboring pixels, which are brighter or darker than itself above a certain threshold. That's followed by computing decision trees to classify corners in a computationally efficient manner. Then features correspondences are calculated using Lucas Kanade optical flow[18]. The motion between consecutive frames is determined to be projective transformation, thus represented by a 2D homography matrix. The Least Median Square Estimator (LMedS) algorithm[19] is used to estimate the coefficients of the homography while rejecting outliers.

LMedS is used rather than RANSAC which is another parameter estimation method, because it requires a fixed threshold to determine the inliers. In LMedS the only condition is to have at least half of the data as inliers. That is provided in aerial imagery, in which most of the scene is background. Instead the LMedS algorithm uses a standard deviation estimate to reject outliers, which is computed in equation 1, where M is the minimum median of squared residuals and n is the number of corners.

$$\sigma = 1.4826\left[1 + \frac{5}{n-4}\right]\sqrt{M} \quad (1)$$

3.2. Outliers Clustering

Instead of applying image registration as in[8][10], outliers detected from the previous step are clustered and used directly as the potential targets. Thus, the proposed work avoids registration errors. The clustering approach used is Density-Based Spatial Clustering (DBSCAN) algorithm [20]. It doesn't need the number of clusters as input, works real-time on large data and is able to estimate clusters of arbitrary shapes. The algorithm's input is the search area to look for cluster members, and the minimum number of points constituting a cluster. The clusters produced from the previous stage are potential targets and are further filtered. As in [12] those potential targets are checked for track persistency over certain number of frames. To ensure the rejection of false positives due to false matches or motion estimation errors. The output clusters are shown in Figure 2.

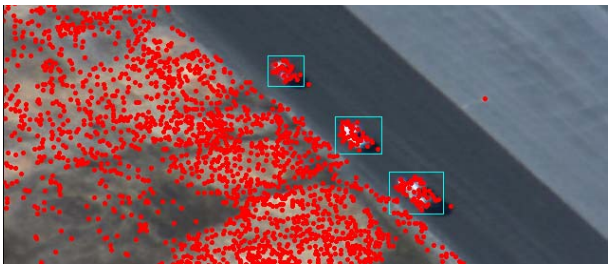

Figure 2: Clustered outliers detected from LMedS, redpoints indicate all corners detected with very low threshold.

4. Target Tracking

4.1. Overlap-Rate-Based Data Association

The main task of the data association stage is to correlate the detected targets to the previous tracks. There exist different approaches, like the simple one-to-one data association which assumes that one track can correlate to only one detection, which fails in split and merge cases to be explained. In merge case, two cars are near each other and detected as only one cluster, the association technique should identify that this is a merge. In split case, when a single tracked target is detected as two clusters, the association technique should also identify this as split. The data association technique used is the overlap-rate based data association mentioned in [12]. Based on the intersection between detections and previous tracks, an association flag is determined. The flag is either 'Object', 'NA', 'Split', or 'Merge'. 'NA' flag means that no association is established, this can occur if a new target entered the scene. In case of no association to previous tracks, a cascaded classifier is utilized to search for the target. This work is further explained in the P-N Learning section. In Algorithm 1, a pseudo-code for the algorithm is presented. The two important ratios to determine the flags R_D and R_T are defined as follows:

1) $\boldsymbol{R_D}$: The ratio between the area of intersection and the area of the detection.
2) $\boldsymbol{R_T}$: The ratio between the area of intersection and the area of the track.

4.2. Kalman Filter

Kalman filter [21] is used to model the state of the target, its position and velocity. It is comprised of two stages prediction and correction. In case of 'Object data association, the measurement is used in the Kalman correction stage. Otherwise, the predicted location is used within a cascaded classifier to search for the target. This classifier is explained in details in the next section. The Kalman is initialized with the detected targets' centroids. In[12] the matrices used in the Kalman filter are mentioned. The process and measurement noise W and V are assumed to be Gaussian noise with constant variances equals 0.5 and the initial estimation error covariance is set to zero.

Algorithm 1:
Foreach track T:
  Foreach detection D:
    Calculate R_D, R_T
    IF $R_D < TH$ AND $R_T < TH$:
      Assoc = "NA"
    ELSEIF $R_D > TH$ AND $R_T < TH$:
      Assoc = "Split"
    ELSEIF $R_D < TH$ AND $R_T > TH$:
      Assoc = "Merge"
    ELSEIF $R_D > TH$ AND $R_T > TH$:
      Assoc = "Object"
  End

  IF Assoc == "Object"
    Update P-N Learning Database
    Update Kalman
  ELSEIF Assoc == "Merge" || "Split" || "NA"
    Predict Kalman
    Search for target using P-N Learning
End

5. Learning Target Appearance

The proposed framework maintains target appearance throughout the video sequence by using semi-supervised P-N learning [2] algorithm. This module learns a classifier from labeled data, which are the patches of detected moving targets, and then uses the unlabeled data to bootstrap its performance. This process creates a learned database of positive and negative samples to be used for robust target tracking.

5.1. P-N Learning

P-N learning is similar to self learning, which is one of the oldest approaches in semi-supervised learning. The main difference between them is that self learning uses the most confident examples from unlabeled data to augment the training data iteratively. P-N learning, on the other hand, uses defined constraints to select examples from the unlabeled data to augment the training set. Given examples x and labels y, the set of labeled data is called L={(x, y)} and the set of unlabeled data is called X. The task of the P-N learning is to learn a classifier f from labeled data L and bootstrap it with unlabeled data X, to estimate classifier parameters θ. The bootstrapping is done iteratively; in iteration k the procedure is as follows:

1) Compute labels ${y_u}^k = f(x_u|\theta^{k-1})$ for all$x_u \epsilon$ X.
2) P-N experts identify the incorrect labels, where P-expert identifies false negatives and labels them as positive $n^+(k)$ while N-expert identifies false positives and labels them as negative $n^-(k)$.
3) Augment the training set with $n^+(k)\ and\ n^-(k)$, then retrain the classifier to estimate θ^k .

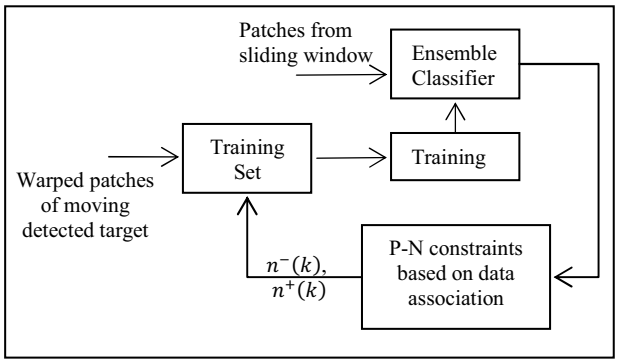

Figure 3: Block Diagram explaining P-N Learning procedure

Figure 3 shows the procedure of the P-N Learning as a semi-supervised technique. The task of target tracking using P-N learning, utilizes warped patches of the moving detected target, which are rotated and scaled versions of the detected target with increments set as parameter to the algorithm. And those warped patches are used first time as the labeled set L. Then with each image frame, a sliding window approach [24] is used, where the input image is scanned by a window of various scales. Then each window (patch) is considered as an unlabeled example x_u, and passed to a cascaded classifier to assign label y as object or non-object. The cascaded classifier used has three stages to enable the early rejection of non-object patches. The stages are as follows: patch variance, ensemble classifier, Parzen window. In the patch variance stage, patches with gray-level variance less than half the variance of the original patch are rejected.

In the ensemble classifier, a number of base classifiers are used [22][23] where each one performs pixel comparisons. Initially the locations of pixel comparisons are randomly generated as a priori to the classifier. Then a class label is assigned to the patch according to the average of posteriors of the base classifiers. Subsequently, it is iteratively augmented with unlabeled data according to the predefined structural constraints explained in the next section. In the last stage, Parzen window is used to classify all the patches within fixed range from the predicted Kalman centroid as object. If no detection is finally labeled as object, a simple Lucas-Kanade flow is computed for a small grid of points around the centroid to determine the target location.

5.2. Novel P-N Constraints

In order to understand the nature of P-N structural constraints, structured data are defined first. Structured data are data with dependant labels. For example, in the case of target tracking by detection, a trajectory for the target is to be found and the patches that are close to this trajectory are labeled object, while far patches are labeled non-object. This structure within the data is utilized using the P-N constraints. The positive (P) and negative (N) constraints are used to verify the labels assigned to the examples with the classifier, and after being corrected are added to the training set. In [2] a proof of convergence of the classifier's error was shown depending on equation (2) where the state vector $x(k) = [\alpha(k)\ \beta(k)]^T$ contains the false positives $\alpha(k)$ and false negatives $\beta(k)$. It was proved that if the Eigen values of M were less than 1, it will converge. The transition matrix M in equation (3) contains four valuesP^+, R^+, P^-, R^- explained as follows:

1)$\boldsymbol{P}^+$: P-precision, number of correct positive examples divided by total number of samples output by P-constraint.
2)$\boldsymbol{R}^+$: P-recall, number of correct positive examples divided by number of false negatives.
3)$\boldsymbol{P}^-$: N-precision, number of correct negative examples divided by number of all examples output by N-constraint.
4)$\boldsymbol{R}^-$: N-recall, number of correct negative examples divided by total number of false positives.

$$x(k+1) = M.x(k) \tag{2}$$

$$M = \begin{bmatrix} 1 - R^- & \frac{(1-P^+)}{P^+}R^+ \\ \frac{(1-P^-)}{P^-}R^- & 1 - R^+ \end{bmatrix} \tag{3}$$

Novel P-N constraints are used in the proposed framework. The positive (P) constraint utilizes the temporal structure; *i.e.* it assumes that the target moves along a trajectory. In this case, the data association

decision, which relates the moving target detections to the targets' tracks, is used as the (P) constraint. When an association between moving target detection and an old track is established as 'Object', it requires that patches with high overlap with that detection are positive. The negative (N) constraint utilizes the spatial structure that assumes that the object can appear in only one location. Patches that are far from the target's trajectory and were labeled wrongly as positive are corrected. In Figure 4 the positive and negative samples are shown.

6. Results and Discussion

This section presents quantitative and qualitative analysis of the proposed framework in comparison with the work in literature. The datasets used are the DARPA VIVID datasets [25] which are aerial datasets collected with different challenging scenarios. Eglin1 and Eglin 3 both contain illumination changes of the target. Eglin 1 contains targets going out of the camera field of view and in again in consecutive times. Eglin 2 contains rapid camera panning. Both Eglin 2 and Eglin 3 demonstrate the challenging scenario of two convoys running pass each other with similar targets partially occluding each other. Finally, the Redteam dataset has continuous viewpoint and target's appearance change. The first section shows the qualitative analysis of the proposed work compared to the TLD [2] framework to highlight the effect of the newly introduced P-N constraints.

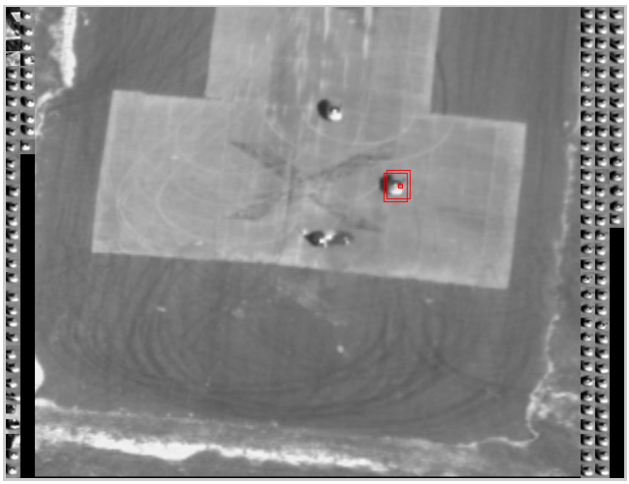

Figure 4: The two red rectangles resemble potential detections. The red dot indicates the predicted Kalman centroid, the positive and negative samples are insets on right and left of the image, respectively.

The second section shows the quantitative analysis where the proposed framework is compared to the work in [13] which reports on the performance of 3 algorithms: the basic meanshift tracker [27], the variance ratio [26], and the peak difference [26]. Our work is also compared against Fragtrack [28], Online Semi-supervised Boosting (SemiBoost) [29], Multiple Instance Learning (MIL)[30], and the Tracking-Learning-Detection framework (TLD) [2][4]. As well as comparison with Moving Target Detection (MTD)[15][16] which is similar to this framework but without utilizing P-N structural constraints. All trackers were initialized with manual labels for the targets, except for[16] and the proposed work where both were initialized by automatic target detection. Finally, last section shows the ability of the framework to recapture the target.

6.1. Qualitative Analysis

In Figure 6, qualitative analysis comparing both the proposed work and TLD [2] is presented. The frames selected from all datasets are to manifest the challenging scenarios. In Eglin 1 dataset top row in Figure 6, the scenario of an illumination change occurs and the target is rotating. TLD shows low overlap because of its dependence on the median flow tracker in the PN constraints. In Eglin 2 dataset second row of Figure 6, both trackers succeed in discriminating similar targets from each other. But all other trackers as will be shown in the quantitative analysis except the adaptive one fails in this scenario. In the third row in Figure 6 Eglin 3, both trackers succeed in tracking the target in case of the partial occlusion scenario, while all other trackers failed in it. But TLD showed low overlap with illumination changes on the target in the last frame. Finally, in Redteam dataset last row in Figure 6, it's obvious that TLD drifted with the shadow of the target. This is due to again the PN constraints used depending on the median flow tracker. It's also shown in the last frame that TLD had already drifted and wasn't able to recover. While the proposed work succeeded again in partial occlusion scenario.

6.2. Quantitative Results

In this section quantitative comparisons between the proposed framework and existing techniques are presented in Table 2 and Table 3. Two metrics are used for the evaluation, the recall and the average overlap. The recall is the number of frames from the dataset tracked before drift, *i.e.* when no overlap occurs between the target and ground truth. The average overlap is the percentage of the intersection between the target and ground truth with respect to the ground truth area [13]. Note that the overlap is only measured on the initial frames till the first time the tracker loses the target. As demonstrated in the recall results in Table 2, the proposed framework alleviates the drift problem that occurs with the TLD [2] framework. This is due to the fact that the proposed framework uses the feedback from the moving target detection unlike TLD that uses the median flow tracker. The median flow tracker main weakness is that it drifts with the background points

as the target undergoes scale changes. Furthermore, it overcomes the deficiency of the adaptive tracker [13] in the Eglin3 dataset that happened when the target was partially occluded with another car. While the proposed work succeeded because of using the predicted Kalman location in the cascaded classifier.

Table 3 shows the results of the average overlap metric where it can be seen that the proposed framework supersedes the other trackers. In case of both Eglin 1and Redteam it superseded with 25 % the best approach from among the techniques that achieved recall of 1, without including MTD which is our previous work. In Eglin 3, the variance ratio [26] and the adaptive tracker [13] achieved higher overlap in case of the Eglin 3 dataset, however *both have very low recall*. Both tracked the target to the first 12% and 20% of the dataset, respectively, and then lost the target. So their overlap is measured on this small percentage. While the proposed work achieved best overlap among approaches with 100% recall [2]. In case of Eglin 2, the only tracker to have reported better is the adaptive tracker [17] with 100% recall. However, this is due to the fact that the target's first initialization was manual, whereas the target initialization used in the proposed work is from the automatic detection phase. That is why when the target is not perfectly segmented, the tracker is affected and the results give lower overlap. So the deficiency occurred from the detection phase not from the proposed tracker.

Our previous work MTD without using PN constraints failed in Eglin 2 and Eglin 3 due to its simple tracker. Finally, to test that the novel P-N constraints presented cause the classifier's error to converge to zero, the four values P^+, R^+, P^-, R^- are calculated for the Eglin 1 and Redteam datasets. The results shown in Table 1 demonstrate that the Eigen values are less than one which proves that the PN constraints will cause convergence. Although the Eigen values in Eglin 3 are high, it's worth noting that the ensemble classifier is followed by Parzen window stage to reject false positives.

	P^+, R^+	P^-, R^-	λ_1, λ_2
Eglin 1	0.99, 0.89	0.78, 0.68	0.33, 0.1
Eglin 3	0.95, 0.185	0.9, 0.02	0.98, 0.8

Table 1:PN Constraints Evaluation

6.3. Recapture Target out/in Field of View:

One of the main contributions of learning the target's appearance is the ability to recapture the target after going out of the field of view. In the MTD framework[16] when the target went out of field-of-view and in again, it was identified as a new target. But since the proposed approach uses the novel P-N constraints discussed, the classifier is able to maintain a set of positive samples for the target. Then the approach applies a sliding window over the whole image with different scales. Those windows (patches) are passed to the classifier to identify and redetect the target. In Figure 5, an example scenario is shown for one of the targets, marked with the dark blue bounding box, going twice out of the camera field of view and in again. In frame #368, the target is about to go out of camera field of view. Frame #450 shows that the target is still out of view for about 100 frames. The target is recaptured when it entered the field of view again as shown in frame #573. The target disappears in frame #658 occurs, and recaptured again in frame #730.

7. Conclusions

In this paper, we introduced novel P-N constraints using the data association decision as structural constraints. Based on the assumption that unlabeled data are structured, meaning that their labels are dependant. The positive and negative constraints are used to control the labeling of the data and bootstrap the performance of the classifier. Those constraints were utilized within target detection and tracking framework for UAV imagery, and tested on the publicly available DARPA datasets. The experiments were conducted among different state of the art trackers. The quantitative results showed improvement in both recall and average overlap in the proposed work. Also the framework introduced had additive advantages over the trackers compared to. The ability to recapture the target after being out of the field of view and to discriminate similar targets was shown.

Figure 5: Eglin 1 dataset, recapture target scenario, the target is colored in dark blue. Frames from left to right. Frame # 368, 450, 573, 658, 730.

	Means hift [27]	VarianceRatio[26]	PeakDiff[26]	FragTrack [28]	SemiBoost [29]	MIL [30]	Adaptive Tracker [13]	MTD [16]	TLD [2]	Proposed Work
Eglin1	0.19	0.29	**1**	**1**	0.55	**1**	**1**	**1**	**1**	**1**
Eglin2	0.4	0.29	0.33	**1**	0.18	0.42	**1**	0.4	**1**	**1**
Eglin3	0.22	0.12	0.12	0.89	0.13	0.2	0.22	0.1	**1**	**1**
Redteam	0.85	**1**	**1**	0.79	0.1	**1**	**1**	**1**	0.57	**1**

Table 2: Recall evaluation for the trackers on Eg1, Eg2, Eg3, and redteam. The proposed superseded TLD in Redteam.

	Mean Shift [27]	VarianceRatio [26]	PeakDiff [26]	FragTrack [28]	SemiBoost [29]	MIL [30]	Adaptive Tracker [13]	MTD [16]	TLD [2]	Proposed Work
Eglin1	65.5	76.87	61.76	58.1	62.9	47.5	61.62	83.8	55.7	**86.7**
Eglin2	91.09	85.19	90.54	76	90.2	82.5	**93.32**	74.9	54.9	83.1
Eglin3	86.96	93.74	92.27	81.7	80.9	79.2	88.66	64.8	63.6	**73.5**
Redteam	68.37	73.24	72.37	43.99	70.5	63.7	72.61	84.9	38.5	**98.3**

Table 3: Average Overlap evaluation for the trackers on Eg1, Eg2, Eg3, and redteam datasets. The proposed work superseded other trackers in Eg1 and Redteam with a marginal difference. In Eg 3 although it appears that Variance Ratio got the highest overlap but it had very low recall 12% only of the dataset .

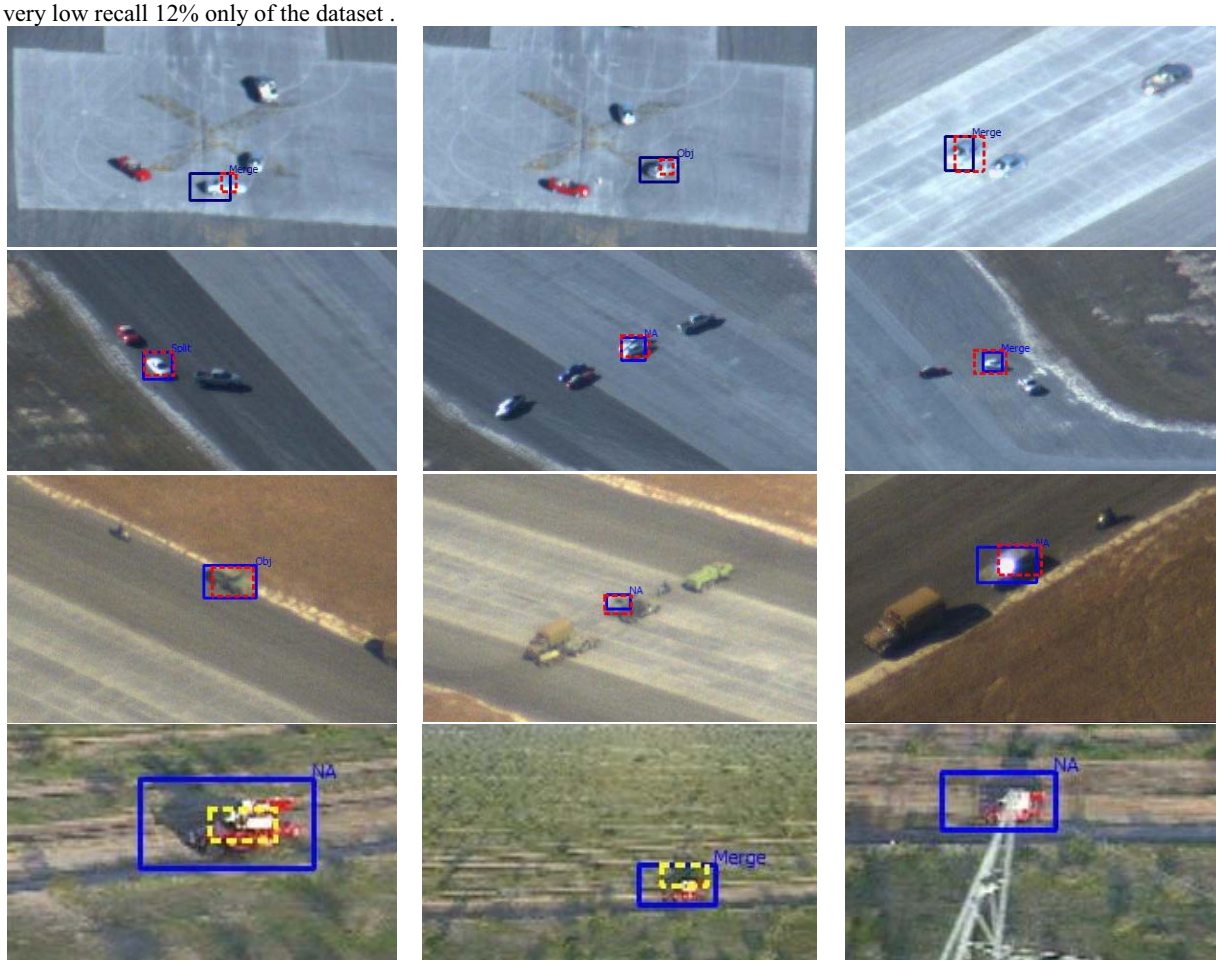

Figure 6: Qualitative Analysis, blue solid line is the proposed work, and the dashed line is the ouput of TLD. From top to bottom, the datasets presented are Eglin1, Eglin2, Eglin3, and Redteam. From left to right, Eglin1 Frame #301, 351, 1355. Eglin2 Frame # 301, 641, 848. Eglin3 Frame # 50, 561, 2457. Redteam Frame # 613, 780, 523.

References

[1] O. Chapelle, B. Scholkopf, and A. Zien, (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, MA.

[2] Z. Kalal, J. Matas, and K. Mikolajczyk, (2010). *P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[3] Z. Kalal, K. Mikolajczyk, and J. Matas, (2010). *Forward-Backward Error: Automatic Detection of Tracking Failures*. International Conference on Pattern Recognition. 23-26.

[4] Z. Kalal, K. Mikolajczyk, and J. Matas, (2012). *Tracking-Learning-Detection*. Pattern Analysis and Machine Intelligence (PAMI).

[5] S. Avidan, (2007). *Ensemble tracking*. Pattern Analysis and Machine Intelligence (PAMI). 29(2), 261-271.

[6] H. Grabner and H. Bischof, (2006). *On-line boosting and vision*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[7] R. Collins, Y. Liu, and M. Leordeanu, (2005). *Online selection of discriminative tracking features*. Pattern Analysis and Machine Intelligence (PAMI). 27(10),1631–1643.

[8] S. Ali and M. Shah (2006). *COCOA: tracking in aerial imagery.* SPIE, volume 6209.

[9] C. Benedek, T. Sziranyi, Z. Kato, J. Zerubia (2007). *A Multi-Layer MRF Model for Object-Motion Detection in Unregistered Airborne Image-Pairs*. IEEE International Conference on Image Processing, (ICIP). VI-141-VI-144.

[10] Yao Fenghui, A. Sekmen, M. Malkani, (2008). *A novel method for real-time multiple moving targets detection from moving IR camera.* 19th International Conference on Pattern Recognition, (ICPR). 1-4.

[11] A. Ollero, J. Ferruz, F. Caballero, S. Hurtado, L. Merino (2004). *Motion compensation and object detection for autonomous helicopter visual navigation in the COMETS system*. IEEE International Conference on Robotics and Automation. Proceedings, (ICRA). 19- 24.

[12] Hongwei Mao, Chenhui Yang, Jennie Si and Glen P. Abousleman, (2010). *Automated multiple target detection and tracking in UAV Videos*. Proc. SPIE 7668, 76680J.

[13] Junqiu Wang, Yasushi Yagi, (2008) . *Integrating Color and Shape-Texture Features for Adaptive Real-Time Object Tracking*. IEEE Transactions on Image Processing. 17(2), 235-240.

[14] Celine Teuliere, Laurent Eck, Eric Marchand, (2011). *Chasing a moving target from a flying UAV*. Intelligent Robots and Systems (IROS). 4929-4934.

[15] Menna Siam, Mohamed ElHelw, (2012). *Robust Real-time Multiple Moving Targets Detection and Tracking Framework for UAV Imagery*. IROS'12 Workshop on Robots and Sensors integration in future rescue Information system (ROSIN'12).

[16] Menna Siam, Ramy ElSayed, Mohamed ElHelw, (2012). *On-board multiple target detection and tracking on camera-equipped aerial vehicles*. IEEE International Conference on Robotics and Biomimetics (ROBIO).

[17] Rosten and T. Drummond, (2006). *Machine learning for high-speed corner detection*. In European Conference on Computer Vision (ECCV).

[18] B. Lucas and T. Kanade (1981). *An iterative image registration technique with an application to stereo vision.* IJCAI, 81, 674–679.

[19] Z. Zhang, (1995). *Parameters estimation techniques. A tutorial with application to conic fitting.* RR-2676, Inria.

[20] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, (1996). *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. in Proc. KDD. 226-231.

[21] G. Welch and G. Bishop, (1995). *An introduction to the Kalman filter.* Technical report, Univ. North Carolina, Chapel Hill. TR95-041.

[22] V. Lepetit and P. Fua, (2006). *Keypoint recognition using randomized trees*. IEEE transactions on pattern analysis and machine intelligence. 28, 1465–79.

[23] M. Calonder, V. Lepetit, and P. Fua, (2010). *BRIEF : Binary Robust Independent Elementary Features*. European Conference on Computer Vision (ECCV).

[24] P. Viola and M. Jones, (2001). *Rapid object detection using a boosted cascade of simple features.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[25] Robert T. Collins, Xuhui Zhou, and Seng Keat Teh, (2005). *An Open Source Tracking Testbed and Evaluation Web Site*. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005).

[26] R. T. Collins and Y. Liu,(2005). *Online selection of discriminative tracking features*. IEEE Trans. Pattern Anal. Mach. Intell., 27(10),1631–1643.

[27] D. Comaniciu, V. Ramesh, and P. Meer, (2003). *Kernel-based object tracking*. IEEE Trans. Pattern Anal. Mach. Intell. 25(5), 564–577.

[28] A. Adam, E. Rivlin, and I. Shimshoni, (2006). *Robust fragments-based tracking using the integral histogram*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 1, 798–805.

[29] H. Grabner, C. Leistner, and H. Bischof, (2008). *Semi-supervised on-line boosting for robust tracking*. European Conference on Computer Vision (ECCV).

[30] B. Babenko, M.-H. Yang, and S. Belongie, (2009). *Visual Tracking with Online Multiple Instance Learning*. Conference on Computer Vision and Pattern Recognition (CVPR).