Airborne Visual Tracking of UAVs with a Pan-Tilt-Zoom Camera

Athanasios Tsoukalas[1], Nikolaos Evangeliou[1], Nikolaos Giakoumidis[2] and Anthony Tzes[1]
[1]*Engineering Division, New York University Abu Dhabi, P.O. 129188, Abu Dhabi, U.A.E.*
[2]*Core Technology Platform, New York University Abu Dhabi, P.O. 129188, Abu Dhabi, U.A.E.*

Keywords: Variable Object Tracking, Visual Homography, Optical Flow.

Abstract: The visual detection and tracking of UAVs using a Pan-Tilt-Zoom (PTZ) camera attached to another aerial platform is the scope of this article. The long-term tracker is performing image background subtraction using visual homography and is used to initialize a short term tracker that works in parallel to enhance the tracking for various motions. The moving UAV is detected using optical flow concepts and its bounding box encapsulates its detected features. A Kalman predictor provides a robust smooth tracking of the bounding box in the temporary absence of a detected UAV. The camera pans and tilts so as the tracked UAV is centered within its Field-of-View and zooms in order to expand the UAV's view. Experimental results are offered using an evader-tracker UAV-group to validate the presented tracking algorithm.

1 INTRODUCTION

In this work the subject of the Long Term Tracking (LTT) of a target UAV from another one equipped with a Pan - Tilt - Zoom (PTZ) camera is considered. The estimation of the strong background image feature points in comparison to those of a moving object is performed through a comparison of the image optical flows and the predicted position of the feature points according to the Homography transformation derived between feature points found in previous frame and the estimate in the current frame. A Kalman prediction filter (Paul and Musoff, 2015) can smooth the trajectory in the presence of noisy measurements, while a correlation based tracking algorithm is also used in combination with our LTT scheme to account for the cases where the UAV blends with the background.

Previous research on Variable Object Tracking (VOT) has focused on strong image features matching and Homography. In (Eshel and Moses, 2008) multiple static cameras are used to infer multiple moving targets, using the homography of three planes parallel to the floor between each pair of cameras. In (Arróspide et al., 2010) the ground plane is detected using a single moving camera, by finding the homography between the ground planes in successive images, using a feature matching approach. These methods assume that the floor or a top down view is always present in the camera view. In (Cui et al., 2019) a moving object segmentation is performed from stationary or moving cameras based on homography constraints across multiple frames, that describes the background motion, without assuming a planar scene using a pixel precision. In (Dey et al., 2012) the detection of independently moving foreground objects in non-planar scenes captured by a moving camera is performed, using a multi frame monocular epipolar constraint of camera motion. The method applies optical flow across the whole image between consecutive frames, that scales exponentially in performance loss as the image resolution is higher.

In (Viswanath et al., 2015) foreground object segmentation is performed with a background modelling technique that models each pixel with a single Spatio-Temporal Gaussian. In (Sheikh et al., 2009) objects are tracked using a freely moving camera with no assumptions that the background approximated by a plane, by building a sparse background model by estimating a compact trajectory basis from trajectories of salient features across the video. Following the background subtraction by removing trajectories that lie within the space spanned by the basis, builds the foreground and background appearance models. Then an optimal pixel-wise foreground/background labeling is obtained by efficiently maximizing a posterior function. In (Fu et al., 2016) the algorithm tracks by finding feature correspondences in a way that an improved binary descriptor is developed for global feature matching, while an iterative Lucas–Kanade

Tsoukalas, A., Evangeliou, N., Giakoumidis, N. and Tzes, A.
Airborne Visual Tracking of UAVs with a Pan-Tilt-Zoom Camera.
DOI: 10.5220/0010112900900097
In *Proceedings of the International Conference on Robotics, Computer Vision and Intelligent Systems (ROBOVIS 2020)*, pages 90-97
ISBN: 978-989-758-479-4

optical flow algorithm is employed for local feature tracking and a local geometric filter (LGF) module maintaining pairwise geometric consistency is used to identify outliers.

In all of the above methods (excluding (Viswanath et al., 2015)), the feature points are identified and matched between frames, resulting in one extra step dependent on both the feature matching between frames algorithm and the feature selection algorithm. In our method we do not perform any feature matching, using directly the positions estimated by the optical flow as next-frame features and calculate the homography matrix based on the features found in the previous frame and the estimated by the optical flow in the current frame. This allows for a one-to-one correspondence of features positions between frames, removing any ambiguity as to which features are matched and dramatically enhances performance as there is no need for a feature matching step. In (Viswanath et al., 2015), similar to our method the feature points in the previous frame are tracked using an optical flow method and a Homography is calculated for the background pixels, under the assumption of an initial background state image and subsequently updating a distribution of background probability of each pixels in the new frame. This is used to model the new background and perform background subtraction using the entire new background image as mask, requiring a significant computational burden.This method does not behave well in the absence of many background points to estimate the Homography, and the foreground object estimation needs more computations, since it operates on the whole image for both the background transformation/subtraction.

In Section 2 the visual tracking problem of the UAV from another airborne UAV statement is provided, followed by the case of identifying a flying object using the Homography based method in Section 2.1. In Section 3 the method of estimating the image based optical flow is discussed using the current and previous image frames. In Section 4 the method of estimating the image flow for background image features based on the Homography transformation derived between image feature points found in previous and current frames is discussed as well as the method of comparing the two flows in order to evaluate which parts of the image are background, that is when the optical flow velocity vectors match closely those of the expected by the Homography based flow estimation of the image features corresponding to the background. The method of calculation of the bounding box around the tracked target UAV points that have been identified as foreground is also discussed. In Section 5 the implementation of the Kalman predictor for the smooth follow of the tracked UAV during tracking window estimation noise is discussed. In Section 6 a correlation based tracker is considered and described, for further enhancing the ability to track the object when the flow based method is not providing a result due to object blending with the background. In Section 7 the control of the PTZ camera motion to track the target UAV is discussed along with the overall algorithm. In Section 8 the results of an experiment of tracking a UAV from a flying UAV pursuer is presented and analyzed.

2 AIRBORNE VISUAL-TRACKING OF UAV

The algorithms are running locally on a mini i7-based Intel NUC PC on board of an enhanced Vulkan UAV shown in Figure 1.

Figure 1: Tracker UAV with mounted PTZ-camera.

Real time LTT of an object using a video-stream is a complex task due to the limited information we have about the object. The various correlation methods give a good tracking performance, given the motion is not very fast and can keep the correlation strong between frames, but fail when the motion is fast and loose the tracking window. These must be initialized with a starting window that engulfs the object to track. Thus a LTT-scheme based on object motion is required in order to track a general object across all motions regions, occlusion and change of appearance. Such a scheme is proposed in this article, enhanced with a correlation based tracker for the cases where the object blends with the background since its motion which can be slow or not moving relative to it.

2.1 Object Tracking Method using Homography

The background subtraction method used in order to identify the moving object motion is relying only on visual feedback and homography calculations (Szeliski, 2010) between two frames. This method is robust against general disturbances and performs well when IMU measurements are not available. Initially a strong image features set is identified on the previous camera frame and an optical flow technique is used to estimate the position of the features in the current frame, as shown in Figure 2. The method involves the discovery of special image areas with specific characteristics. The feature set is a collection of pixel points in the camera image, while various algorithms are available for the discovery of image features (Tareen and Saleem, 2018).

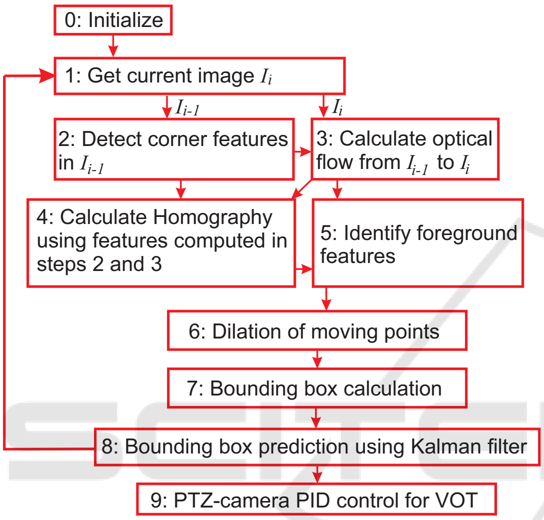

Figure 2: Visual Homography based tracking.

OpenCV's "goodFeaturestoTrack" (Shi and Tomasi, 1994) algorithm was used for finding the strong corners image features in our experiments Under the assumption that the background holds the majority of the pixels with a consistency in displacement, a homography is calculated that transforms the features positions from the previous to the current frame, which correspond to the background pixels. The previous frame features are then transformed using the homography to get their position in the current frame. We assume that the background points transformed with the homography that corresponds mainly to the background pixels group will coincide with the estimated ones by the optical flow, while the moving objects features estimated by the optical flow will diverge from the homography transformed pixels, as shown in Figure 3.

The pixels that correspond to static background objects will follow the predicted motion by the Homography transformation and coincide to the positions predicted by an optical flow based estimation, while the rest will be classified as belonging to

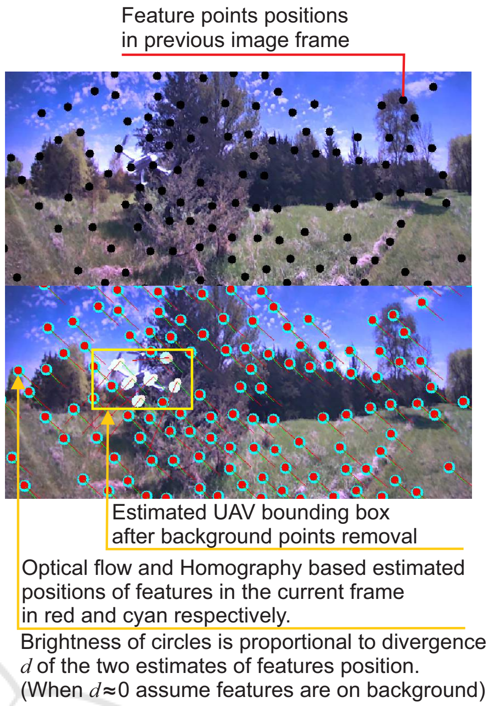

Figure 3: Homography/Optical flow tracking method.

moving objects of interest (Figure 4). The calculations of the optical flow follow the Lucas - Kanade method (Lucas and Kanade, 1981) with an extension using a pyramidal scheme with variable image resolutions (Bouguet, 2000). The basic optical flow premise is to discover the position of an image feature from previous frame, in the current frame.

One downside of the technique is that when the tracked object remains static and blends with the background, it cannot be identified, thus a correlation short term tracker relying on the MOSSE tracking algorithm (Bolme et al., 2010) is used. This identifies the UAV's position until new estimated positions from the flow based method are available. The benefit of the LTT-schemer is that requires zero a priori information, unlike the methods that require prior knowledge of the target bounding window for initialization.

2.2 Homography Calculation

The homography or perspective transform can be formulated as a 3×3 operator H on homogeneous coordinates

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = H\tilde{x} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}. \quad (1)$$

The resulting homogeneous coordinates x_i', y_i' are normalized as

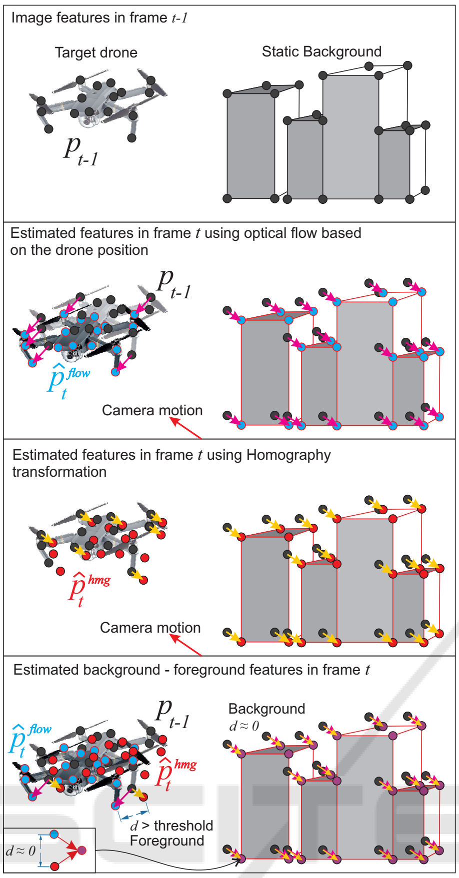

Figure 4: Foreground identification method.

$$x_i' = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}}, \; y_i' = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}}, \quad (2)$$

while the operator H is calculated such that minimizes the back projection error

$$\sum_i \left(x_i' - \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}} \right)^2 + \left(y_i' - \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}} \right)^2. \quad (3)$$

In order to get a good estimation of the transformation matrix in the presence of outliers, which may include foreground points that move not inline with the background point majority, the RANSAC method is used to estimate the homography matrix using this subset followed by a simple least-squares algorithm to compute the quality of the computed homography (defined by the number of inliers). The best subset is then used to produce the initial estimate of the homography matrix. The computed homography matrix is refined further with the non linear Levenberg-Marquardt optimization (Levenberg, 1944; Marquardt, 1963) method to further reduce the re-projection error.

3 OPTICAL FLOW ON SPARSE IMAGE FEATURES

When the tracker UAV is moving, static camera based background subtraction algorithms cannot provide an accurate UAV-tracking, and another background subtraction method needs to be implemented for this case. The method involves the discovery of special image areas with specific characteristics in the camera image, which we note as the image feature set F_{s_t} at time t. The feature set is a collection of pixel points in the camera image $p_t = [px_t, py_t]$. Various algorithms are available for the discovery of image features (Tareen and Saleem, 2018).

OpenCV's "goodFeaturestoTrack" (Shi and Tomasi, 1994) adapted to run on the GPU using the "setUseOpenCL" of the OpenCL module and UMat image holder is employed to find the strong corners image features. This algorithm calculates the corner quality measure at every source image pixel using the Minimum Eigenvalue or the Harris method. For each of the discovered image features of the previous image frame, the optical flow F_{v_t} is derived as a collection of velocity vectors that correspond to each of the image features. For the optical flow estimation of the discovered image features, the OpenCV function "calcOpticalFlowPyrLK" is used. The method operates on a sparse feature set using the iterative Lucas-Kanade method with pyramids and the inputs are the current and previous frames as well as the image features identified from the previous frame. The output is the estimated feature position in the current frame. An example of the feature matching between the current and previous frames is presented in Figure 5.

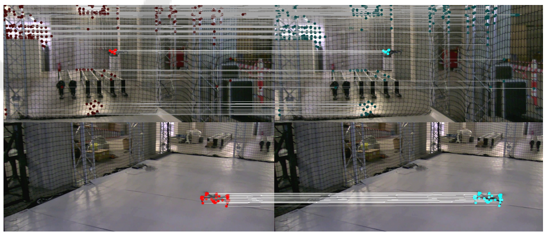

Figure 5: Image feature matching between successive frames.

The algorithm also provides a status flag for each point that describes whether a flow was found for the specific point. The calculations of the optical flow follow the Lucas - Kanade method (Lucas and Kanade, 1981) with the an extension using a pyramidal scheme with variable image resolutions (Bouguet, 2000). The basic optical flow premise is to discover the positioning of an image feature in the previous frame, in the current captured frame.

OpenCV's calcOpticalFlowPyrLK function is used accepting as inputs the previous camera image, and the feature points selected in the previous image to estimate the optical flow on the current camera image. The output is the estimated points along with status number for each estimation, which has a value of one if an estimation of the point in the new image is found, otherwise is zero.

4 HOMOGRAPHY-BASED TRACKING

The tracking method differentiates the foreground to the background image features discovered in the previous image frame. The estimated background points given by the transformation of those points using a Homography matrix are compared to the foreground ones and their estimated positions by an optical flow method, in the current frame.

Given an image g_{t-1} in time $t-1$, we derive the estimation of the positions of the selected image feature points set F_{s_t} in the image g_t in time t, based on the Homography calculated between the strong feature points of the previous and current camera frames, as described in the previous sections. Then an optical flow method as described in Section 3 is used for the same estimation of the feature point positions. The two resulting positions are compared to distinguish the background points from the moving ones, considering the estimation points with distance less than a threshold would belong to the background. The optical flow provides the estimation $\hat{p}_t^{flow}$ of the position of a feature point in the previous frame g_{t-1}, in the current frame g_t for the p_t point in the previous frame and an estimation of the point $\hat{p}_t^{Hmg}$ is also provided from the Homography based transformation on the feature points set of the previous frame g_{t-1}. The distance $b_t = ||\hat{p}_t^{flow} - \hat{p}_t^{Hmg}||$ is evaluated and if $b_t(p_t) \geq b_{bkgd}$, $\forall p_t \in F_{s_t}$ (image feature point set), then the point p_t is considered as moving, otherwise as a background one.

The foreground points correspond to those describing the UAV; a dilation operator followed by the computation of a predicted bounding around the blob of points that surrounds the UAV, as shown in Figure 6 bottom left window, using OpenCV's function "Find Contours" in combination with the "approxPolyDP" function that approximates found contours with polylines that are inserted into the "boundingRect" function to create the bounding box around the polyline. The aforementioned search is performed at each frame and the performance of the system using a 960×540 pixels is at 65fps on the airborne i7 CPU Intel NUC PC. The NUC's GPU is used with OpenCL library for OpenCV to enhance the performance of the optical flow and image manipulation calculations. When the background has static objects across a wide range of distances to the tracker UAV, some points may be identified as foreground, thus a second background layer is assumed and the Homography is recalculated using this new subset, in order to identify the remaining background points. It should be noted that this layer can be computed extremely fast due to the lack of need to compute the optical flow.

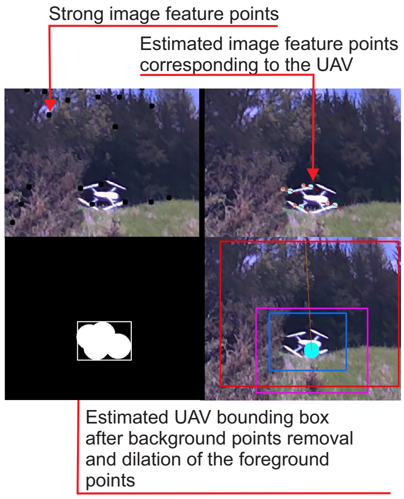

Figure 6: Estimated image features on tracked UAV.

5 KALMAN FILTER TRACKING OF BOUNDING BOX

A Kalman filter prediction scheme tracks the motion smoothly in the absence of a tracking window (or short time sensor malfunction) in some frames and provides the an estimate of the UAV's trajectory. The Kalman predictor uses a state vector $X = [x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}, w, h]$ comprising of the position (Velocity) of the center of the tracking window $[x, y]$ $([\dot{x}, \dot{y}])$, and the width and height of the window w, h. The estimated vector $m_e = [x_m, y_m, \dot{x}_m, \dot{y}_m, w_m, h_m]$ contains the currently identified window center x_m, y_m, its velocity $\dot{x}_m, \dot{y}_m$ and box's width and height w_m, h_m. The $\hat{x}_k$-estimate of X is found in a recursive manner as

$$\begin{aligned}\hat{x}_k &= F_k \hat{x}_{k-1} \\ \hat{P}_k &= F_k \hat{P}_{k-1} F_k^T + Q_k\end{aligned}$$

Under the assumption of a constant acceleration, the state transition matrix is

$$F_k = \begin{bmatrix} \mathbb{I}_2 & dt\mathbb{I}_2 & \frac{dt^2}{2}\mathbb{I}_2 & \mathbb{O}_2 \\ \mathbb{O}_2 & \mathbb{I}_2 & dt\mathbb{I}_2 & \mathbb{O}_2 \\ \mathbb{O}_2 & \mathbb{O}_2 & \mathbb{I}_2 & \mathbb{O}_2 \\ \mathbb{O}_2 & \mathbb{O}_2 & \mathbb{O}_2 & \mathbb{I}_2 \end{bmatrix}$$

where dt is the time delta between two consecutive frames, and $\mathbb{I}_2$ $(\mathbb{O}_2)$ is the 2×2 identity (zero) matrix . The expected process noise covariance matrix of the model is assumed to be $Q = \text{diag}(\sigma_x, \sigma_y, \sigma_{\dot{x}}, \sigma_{\dot{y}}, 0, 0, \sigma_w, \sigma_h)$. The conversion of the state space to measurement units is performed through the measurement adaptation matrices $m_e = H_k \hat{x}_k$, and

$S_e = H_k P_k H_k^T$, where

$$H_k = \begin{bmatrix} \mathbb{I}_2 & \mathbb{O}_2 & \mathbb{O}_2 & \mathbb{O}_2 \\ \mathbb{O}_2 & \mathbb{I}_2 & \mathbb{O}_2 & \mathbb{O}_2 \\ \mathbb{O}_2 & \mathbb{O}_2 & \mathbb{O}_2 & \mathbb{I}_2 \end{bmatrix}.$$

For our tracking window measurements, we define $R_k = \mathrm{diag}(\sigma_{x_m}, \sigma_{y_m}, \sigma_{\dot{x}_m}, \sigma_{\dot{y}_m}, \sigma_{w_m}, \sigma_{h_m})$ as the covariance of the estimated window parameters measurement noise induced uncertainty and z_k the mean of the measurements. The final measurement update $\hat{x}_{k_t}$ and the covariance $\hat{P}_{k_t}$ are defined as

$$\hat{x}_{k_t} = \hat{x}_{k_{t-1}} + K(z_k - H_k \hat{x}_k) \tag{4}$$

$$\hat{P}_{k_t} = P_{k_{t-1}} - K H_k P_{k_{t-1}} \tag{5}$$

where the Kalman gain K is given by

$$K = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \tag{6}$$

The measurement matrix is updated with the estimated bounding box position-width-height and velocity calculated using the position estimations in previous and current frames and the time between them.

6 CORRELATION BASED MOSSE TRACKING ALGORITHM

In the case where a UAV is hovering, the optical flow method may not provide results as essentially the UAV can be considered as background. To avoid this behavior we use the MOSSE tracking algorithm (Bolme et al., 2010) around the tracking window to check if the correlation to the previous frame stays strong, and decide whether to keep the window to the same position or move it to some other position, according to the latest measurements. The MOSSE algorithm requires an initial bounding box around the target, which we provide through our LTT when the confidence is high enough across multiple frames and the MOSSE tracking window to our estimated tracking window have distance beyond a threshold. A weight is given based on how strong the correlation and how far the new measurement from the previous to make the final decision. When MOSSE algorithm alone is used to track the target UAV, the results are generally adequate, but the algorithm needs an initial tracked object bounding box and also can fail to provide information for the loss of tracking, getting locally trapped in an image position different than the required one. Thus an LTT-scheme is needed for MOSSE re-initialization in loss of tracking. The MOSSE algorithm uses the correlation of a filter calculated by the previous bounding box engulfed image and the current image, to track closeness of the current frame to the initial target provided by the LTT. The calculations are performed using FFT, due to the fact that correlation can be expressed as the element wise multiplication of the individual FFT on the correlation filter and current image. Thus the correlation between the input image $F = F(f)$ and the filter $H = F(h)$ is expressed as

$$G = F \odot H^* \; . \tag{7}$$

The filter is initially trained by the UAV image, provided by the optical flow based method that determines the initial tracked object bounding box. The target is then tracked by correlating the current filter over a search window in the next frame and the maximum correlation output position indicates the new position of the target. The filter is then retrained based on the newly identified position. The filter is defined such that it minimizes the square error sum between the actual and desired output of the convolution.

$$\min_{H^*} \sum_i |F_i \odot H^* - G_i|^2 \; . \tag{8}$$

resulting in

$$H^* = \sum_i \frac{G_i \odot F_i^*}{F_i \odot F_i^*} \; . \tag{9}$$

The numerator can be interpreted as the correlation between the input and the desired output and the denominator is the input energy spectrum. The training set is constructed using random affine transformations to generate eight small perturbations (f_i) of the tracking window in the initial frame. Training outputs (g_i) are also generated with their peaks corresponding to the target center. During tracking, a UAV-target can often change appearance by changing its rotation, scale, pose, by moving through different lighting conditions. Therefore, filters need to quickly adapt in order to follow objects. Running average is used for this purpose. The iterative MOSSE filter after the initialization is computed as

$$H_i^* = \frac{A_i}{B_i} = \frac{n G_i \odot F_i^* + (1-n) A_{i-1}}{n F_i \odot F_i^* + (1-n) B_{i-1}} \tag{10}$$

where n is the learning rate. MOSSE's performance ws found to be suitable for real time tracking. In Figure 6, the MOSSE tracking window is shown in blue box around the tracked UAV. The Kalman tracking window is represented by the red box and the magenta one corresponds to the predicted tracking object bounding box by the optical flow and Homography method.

7 PTZ CAMERA CONTROL

The PTZ camera moves according to commands issued by its programming API. A PID controller has been devised to track the discovered target UAV by turning the camera towards the UAV with the goal to center it on the screen at every frame. When the target has been near centered, a zoom action is performed in order to extend the target's bounding box thus eliminating a portion of the background that may affect the tracking result. The overall algorithm for tracking a target UAV is presented in the sequel.

1: Initialize algorithm and retrieve first frame I_0.
2: Retrieve current image I_i and buffer I_{i-1}.
3: Detect the coordinates (x, y) of image feature set g_{i-1}, in previous frame I_{i-1}.
4: Calculate the optical flow from I_{i-1} to I_i such that g_{i-1} corresponds to the g_i features.
5: Estimate background pixels position $\hat{g}_i$ for every g_{i-1} feature using Homography transformation.
6: Subtract $\hat{g}_i$ from the optical flow g_i to get estimation distance $b_t = ||\hat{p}^{flow}_{t+1} - \hat{p}^{Hmg}_{t+1}||$.
7: The feature points p_t in g_i where $b_t(p_t) \geq b_{bkgd} \forall p_t \in g_t$ are considered moving, otherwise these are marked as background.
8: Dilate the foreground points and extract the polygonal boundaries around point concentrations.
9: Generate bounding boxes around identified features clusters and select the largest one that corresponds to the tracked object.
10: Initialize and run in parallel the MOSSE correlation based tracker when the measurements confidence is high across multiple frames.
11: Use a Kalman filter to smoothly track the MOSSE estimated tracking window in each frame.
12: Use the PID controller to direct the PTZ to the tracked object.
13: Repeat step 2.

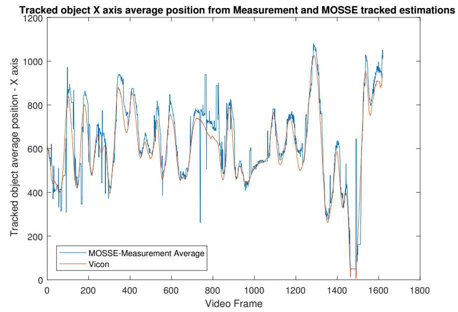

Figure 7: Estimation of X axis of the tracked UAV using MOSSE and Measurements estimations.

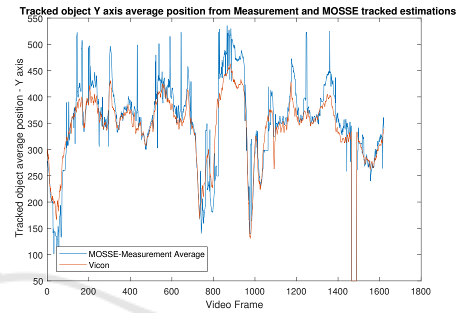

Figure 8: Estimation of Y axis of the tracked UAV using MOSSE and Measurements estimations.

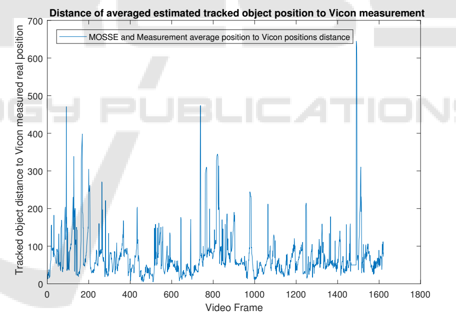

Figure 9: Distance betwen Tracked UAV estimated position and MoCaS real position.

8 EXPERIMENTAL STUDIES

The results of an airborne tracking of a UAV is presented in this section. Both UAVs are monitored by a Motion Capture System (MoCaS) using a set of 24 Vicon cameras. Figures 7 to 9 show the tracking results in pixels. It is shown that the tracking using the Homography based method closely follows the tracked UAV and is further enhanced when the MOSSE correlation based tracking is also used to account for the cases where there is no position estimate. The MOSSE algorithm is re-initialized based on the measurements by our method, given a confidence of three consecutive frames with an estimated object position that does not diverge beyond a threshold. In the case where no measurement tracking window or MOSSE tracking window is estimated, the previous known estimation is used respectively. A presentation of the algorithm working on smooth motion videos and multiple targets can be previewed in the videos in (Tsoukalas, 2020).

9 CONCLUSIONS

A method to determine the trajectory of a moving UAV has been presented in this work, using a PTZ camera. The foreground is discriminated from the background based on the comparison of the predicted background image features motion using the Homography derived by image feature sets between two successive frames and the motion estimated by the optical flow between these frames. The optical flow long term tracking and a correlation short term tracking are combined to provide a robust tracking scheme. A Kalman predictor is also used in order to smoothly follow the trajectory and increase the tolerance of the system to the tracked UAV's bounding box prediction temporal errors.

REFERENCES

Arróspide, J., Salgado, L., Nieto, M., and Mohedano, R. (2010). Homography-based ground plane detection using a single on-board camera. *Intelligent Transport Systems, IET*, 4:149 – 160.

Bolme, D., Beveridge, J., Draper, B., and Lui, Y. (2010). Visual object tracking using adaptive correlation filters. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2544–2550.

Bouguet, J. Y. (2000). Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*.

Cui, Z., Jiang, K., and Wang, T. (2019). Unsupervised moving object segmentation from stationary or moving camera based on multi-frame homography constraints. *Sensors*, 19:4344.

Dey, S., Reilly, V., Saleemi, I., and Shah, M. (2012). Detection of independently moving objects in nonplanar scenes via multi-frame monocular epipolar constraint. In *Computer Vision – ECCV 2012*, pages 860 – 873.

Eshel, R. and Moses, Y. (2008). Homography based multiple camera detection and tracking of people in a dense crowd. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*.

Fu, C., Duan, R., Kircali, D., and Kayacan, E. (2016). Onboard robust visual tracking for UAVs using a reliable global-local object model. *Sensors*, 16:1406.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Math.*, 2(2):164–168.

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, volume 2, pages 674–679.

Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441.

Paul, Z. and Musoff, H. (2015). *Fundamentals of Kalman Filtering: A Practical Approach*. American Institute of Aeronautics and Astronautics.

Sheikh, Y., Javed, O., and Kanade, T. (2009). Background subtraction for freely moving cameras. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1219–1225.

Shi, J. and Tomasi, C. (1994). Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600.

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer-Verlag, Berlin, Heidelberg, 1st edition.

Tareen, S. A. K. and Saleem, Z. (2018). A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–10.

Tsoukalas, A. (2020). Tracking videos using Homography based algorithm. https://drive.google.com/open?id=1MgL1RoW7r36ftm_aloNzV_GJstS4L7Wr.

Viswanath, A., Behera, R., Senthamilarasu, V., and Kutty, K. (2015). Background modelling from a moving camera. *Procedia Computer Science*, 58:289–296.