Effective and Efficient Detection of Moving Targets From a UAV's Camera

Sara Minaeian, Jian Liu, and Young-Jun Son

***Abstract*— Accurate and fast detection of the moving targets from a moving camera are an important yet challenging problem, especially when the computational resources are limited. In this paper, we propose an effective, efficient, and robust method to accurately detect and segment multiple independently moving foreground targets from a video sequence taken by a monocular moving camera [e.g., onboard an unmanned aerial vehicle (UAV)]. Our proposed method advances the existing methods in a number of ways, where: 1) camera motion is estimated through tracking background keypoints using pyramidal Lucas–Kanade at every detection interval, for efficiency; 2) foreground segmentation is applied by integrating a local motion history function with spatio-temporal differencing over a sliding window for detecting multiple moving targets, while the perspective homography is used at image registration for effectiveness; and 3) the detection interval is adjusted dynamically based on a rule-of-thumb technique and considering camera setup parameters for robustness. The proposed method has been tested on a variety of scenarios using a UAV camera, as well as publically available data sets. Based on the reported results and through comparison with the existing methods, the accuracy of the proposed method in detecting multiple moving targets as well as its capability for real-time implementation has been successfully demonstrated. Our method is also robustly applicable to ground-level cameras for the ITS applications, as confirmed by the experimental results. More specifically, the proposed method shows promising performance compared with the literature in terms of quantitative metrics, while the run-time measures are significantly improved for real-time implementation.**

***Index Terms*— Effectiveness, image motion analysis, object detection, robustness, unmanned aerial vehicles.**

I. Introduction

THE first functional step in most autonomous systems (e.g. visual surveillance and intelligent transportation) is to detect the events/targets through sensors (mostly cameras), so that proper decisions can be made for the actuators, accordingly [1]. Furthermore, cameras are increasingly adopted onboard unmanned aerial vehicles (UAVs), autonomous vehicles, and other types of intelligent agents, which require making accurate and real-time decisions. Hence, it is vital to design and develop effective, yet efficient computer vision algorithms for robust operations in dynamic scenes. In particular, moving object detection via UAV for the application of crowd control could be a very challenging problem due to the onboard moving camera with various orientations. This problem is considered ill-posed with respect to the unknown surveillance environment (due to the freely moving UAVs) and possible variations in appearance and motion of the targets. Although several scholarly works in the literature have addressed motion detection from videos captured by a hand-held camera, the existing methods are either too complex [2], [3], or cannot accurately segment the independently moving foreground from moving background in a robust manner [4]–[6]; hence, they lack adequate accuracy and speed to be applied with a monocular camera in an online application. Moreover, for moving object detection via UAV, the focus is on segmenting the independently moving foreground regions. Due to the unknown prior model of the background and dynamics of the foreground targets, supervised object detection techniques are not directly applicable to the problem of unknown moving targets segmentation. This includes popular deep learning and convolutional neural networks (CNN)-based methods, such as fast region-based CNN (R-CNN) [7], superpixel-level CNN (s-CNN) [8], fully conventional networks (FCN) [9], or region proposal network (RPN) [10], which are mostly being used for recognition of known targets or region segmentation (e.g. road detection). It is noted that in general, deep learning methods feature high computational costs and mostly focus on single image detection and recognition, which is not feasible for UAVs with limited onboard computational resources.

Considering such challenges, the main goal of this paper is to propose an effective and efficient foreground segmentation method for detection of independently moving targets to be used via a UAV's moving camera for fast and reliable decision-making (e.g. for autonomous surveillance of human crowds as in [11]). The proposed method is intended to advance the existing literature in three ways, where: 1) Background motion estimation (in absence of a prior model) is done using pyramidal version of optical flow method for tracking extracted background keypoints at every Δt frames (detection interval); 2) Foreground segmentation is done through integrating spatio-temporal differencing and local motion history techniques over a sliding window of frames (with gap Δt), registered via a perspective transformation (i.e. Homography) for reducing

Manuscript received February 28, 2017; revised August 11, 2017, October 1, 2017, and November 18, 2017; accepted December 2, 2017. Date of publication January 5, 2018; date of current version February 1, 2018. This work was supported by the Air Force Office of Scientific Research [a part of Dynamic Data-Driven Application Systems (DDDAS) projects] under Grant FA9550-17-1-0075. The Associate Editor for this paper was Q. Wang. *(Corresponding author: Young-Jun Son.)*

The authors are with the Systems and Industrial Engineering Department, The University of Arizona, Tucson, AZ 85721 USA (e-mail: minaeian@email.arizona.edu; jianliu@sie.arizona.edu; son@sie.arizona.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2782790

the image registration error and also differentiating multiple moving targets; and 3) A heuristic is proposed to adjust the detection interval, based on UAV's altitude and speed for robust real-time performance.

As a result, the independently moving foreground blobs can be segmented accurately in near real-time, while the method is robust to changes in the view angle and movement velocity. To test and demonstrate the proposed method, experimental studies have been conducted, involving a variety of scenarios using UAV videos for crowd control application, as well as available datasets in the literature for autonomous surveillance and other domains such as intelligent transportation systems (ITS). The reason is that, although this work is mainly proposed for visual surveillance by UAVs, it can be easily adopted to ITS, which highly rely on accurate vehicle detection and improved situation awareness techniques to provide required data of traffic counting, speed monitoring, presence detection, and vehicle classification. It is also noted that the focus of visual surveillance application in this work is on scenarios with low to medium density crowd scenes, where the approximate ratio of foreground to background regions in a frame is less than one. Alternative methods, such as Anchor-based group detection [12] have been presented in the literature to be used for highly crowded scenarios.

The rest of this work is organized as followed: Section II provides a review of the related literature on moving object detection from moving camera and discusses research gap and limitations of the existing methods. The innovation and contribution of this work is elaborated in this section as well. Section III describes the proposed method for detection of independently moving targets from the moving camera, in detail. The experimental results on a series of captured shots from a UAV, as well as available datasets are presented in Section IV, for demonstrating the accuracy, efficiency, and robustness of the method, while comparisons with other methods are considered. Finally, Section V concludes the paper and discusses the future research directions.

II. Background and Related Work

Three categories of methods are traditionally considered for solving a motion detection problem: 1) background subtraction, which considers differences between the current frame and a reference background model for foreground segmentation, and is mostly applicable to a static or pan-tilt-zoom camera or known environment (i.e. background); 2) spatio-temporal filtering, which characterizes the motion pattern of the moving target over a 3D volume (x, y, t), but is sensitive to noise and variations in the movement pattern; and 3) optical flow, which considers the relative movements between an observer and the scene, and hence is robust to simultaneous motions of both camera and target. However, it suffers from high computational complexity and thus, is not appropriate for real-time onboard processing. In this work, we face the challenge of segmenting multiple moving targets in an online manner using onboard sensors and limited computational resources. In order to separate the foreground targets with visually plausible boundaries, several complex separation methods are proposed, assuming that the camera is mostly stationary, or the background is known or can be modeled [13], [14]. However, only few research works have addressed the problem of multiple moving targets segmentation in a sequence of dynamic background images, where applying existing methods toward an onboard camera imposes many constraints [2]–[6], [11], [15]–[18].

As a background subtraction approach, structure from motion (SFM) method [2] is used to estimate the camera parameters, the sparse 3D points, and the depth map, via a hand-held camera. Although applying this method makes the resulted foreground mask and moving targets boundaries accurate, they are restricted to the scenes with large depth differences between foreground and background (hence, not robust enough). Moreover, the algorithm is too complex and time-consuming to be used for real-time applications, due to its iterative refinement and camera self-calibration.

Spatio-temporal approach is also used for moving objects detection from moving camera, through motion decomposition. However, such approaches generally require accurate estimation of the foreground motion and hence, are not proper for detecting multiple targets. As an instance, pixel displacements and the sparse error matrices over image sequences were computed in [4], where the latter matrix accounted for the articulated motion of a moving object. However, such a method is mainly applicable to scenarios with planar background and only a single moving object. Moreover, it miss-classifies slowly moving objects as background, while extracts background parts with apparent ensemble motion as foreground, and hence, applying a simple fixed ratio of threshold for foreground segmentation is neither robust, nor effective enough. To detect multiple moving targets, some literature works have also used transformation under spatio-temporal approach for segmenting the moving foreground [3], [5], [15]. However, the presented methods still lack required characteristics to be applied via a freely moving camera in real-time. More similar to the approach in this work, a state-of-the-art method for continuous tracking of moving targets over multiple cameras was proposed in [3]. Their motion detection via moving camera was based on an adaptive background model in which, the camera motion was estimated by an affine transformation. However, such transformation is not appropriate for a freely moving camera onboard a UAV, due to its lack of generality in estimating the scene geometry. Another limitation of such a method is the computational complexity that results from calculating the statistics of each single pixel over the sliding window. In more recent works [5], [15], though, Homography is used to estimate the camera transformation and also a conditional random field (CRF) model is applied to obtain the moving foreground mask. Specifically, [5] combined an ellipsoid shape for a camera projection model. However, the detected moving target mask was not compact enough, and their approach is limited to a camera in a forward moving vehicle, rather than freely moving camera onboard a UAV with different orientations.

As the last category, recent works focused on employing particle trajectories based on optical flow, for moving objects detection [6], [11], [17]–[19]. Although optical flow is rather robust to concurrent motions of foreground and

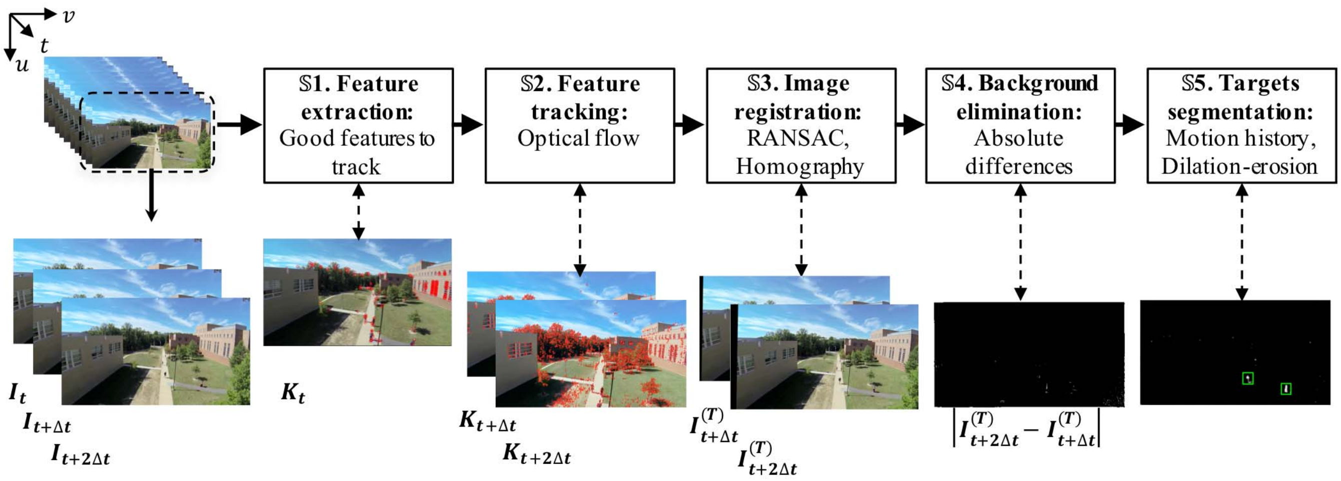

v
t
u
$\mathbb{S}1$. Feature extraction: Good features to track
$\mathbb{S}2$. Feature tracking: Optical flow
$\mathbb{S}3$. Image registration: RANSAC, Homography
$\mathbb{S}4$. Background elimination: Absolute differences
$\mathbb{S}5$. Targets segmentation: Motion history, Dilation-erosion
I_t
$I_{t+\Delta t}$
$I_{t+2\Delta t}$
K_t
$K_{t+\Delta t}$
$K_{t+2\Delta t}$
$I^{(T)}_{t+\Delta t}$
$I^{(T)}_{t+2\Delta t}$
$\left|I^{(T)}_{t+2\Delta t} - I^{(T)}_{t+\Delta t}\right|$

Fig. 1. The framework for effective, efficient, and robust detection of independently moving targets via a monocular moving camera.

background, the existing classification methods generally lack either the speed or the accuracy required for real-time moving target detection onboard a UAV. As an instance, optical flow is used in [6] to extract dense particle trajectories for each mesh-grid pixel only in the first frame, while applying a multi-frame epipolar constraint. Although this constraint provided a consistent classification between moving and static objects, the boundaries of moving targets were not accurate due to mislabeling of neighboring background pixels. Moreover, the assumption of consistent reference plane across all views is rather invalid, due to the camera movements. Minaeian *et al.* [11] used optical flow along with affine transformation between two successive frames for moving target detection. However, as mentioned before, such technique is not robust enough for moving UAV and the accurate results are limited to the hovering movements. In a more recent work [17], the main idea of background motion subtraction (BMS) method is to decompose ensemble motion into those of background and foreground. The algorithm first segments the coarse foreground regions and then applies an adaptive threshold for finer segmentation. Despite the adaptive thresholding, BMS still mixes objects moving at a low speed with the background, at complex scenarios. Moreover, applying mean-shift algorithm for optimizing the foreground segmentation is neither efficient in real-time, nor consistently accurate at boundaries. There are other recent approaches toward motion segmentation (e.g. layered directed acyclic graph [20] or maximum weight cliques [21]), which are not necessarily designed to detect multiple independently moving objects, and their performance can be deteriorated in the case of sudden movements by the target.

Considering the limitations of related literature works, the contribution of our proposed method for detecting independently moving targets from a moving aerial vehicle is three folds:

1) We address algorithm *efficiency* by using pyramidal Lucas-Kanade (LK) tracking of background keypoints every Δt frames, to estimate general background motion, without pixel-to-pixel estimation of camera model.
2) We address algorithm *effectiveness* by integrating local motion history with spatio-temporal approach over a sliding window with gap Δt for segmenting multiple independently moving targets, while reducing image registration error. Also, successive frames are warped using perspective Homography, as a more general model compared to the popular affine transformation.
3) We address algorithm *robustness* by proposing a rule-of-thumb for adjusting the detection interval according to the scenario, to tackle different view-angles and movement velocities.

III. Proposed Methodology

In this section, the details of the proposed method for detecting multiple moving targets via a moving camera onboard UAV are discussed. As the general procedure of motion segmentation in presence of background movements, the first main task includes compensating the camera motion, and then, to subtract the moving background, so that the independently moving blobs in the foreground can be finally segmented. Fig. 1 shows an overview of the major steps of the proposed method. We will cover the first three steps ($\mathbb{S}1$ to $\mathbb{S}3$) in Section III-A below, while the last two steps ($\mathbb{S}4$ and $\mathbb{S}5$) are described in Section III-B.

A. Camera Motion Compensation

To estimate the general motion of the camera, one possible approach (as described by [3]) would be to estimate the displacement of every single pixel across successive frames and then, to compute the affine transformation between the two images. However, such methods would not be efficient due to their high computational complexity. In this work, though, first a number of keypoints are extracted from the reference frame and then, are tracked across multiple frames at Δt intervals, to increase both efficiency (due to a lower number of motion equations) and effectiveness (due to the use of robust features) of the motion estimation method.

During step $\mathbb{S}1$, the method extracts the current frame's keypoints at time t, which have robust features to be tracked. While any robust feature extraction method can be used for this purpose, in this work, we adopt the good features to track (GFTT) method [22], which is invariant to rotation and translation, and thus, provides a reliable motion estimation. In GFTT, corners are characterized by two large eigenvalues for autocorrelation matrix of the second derivative of image I,

$$\begin{bmatrix} {I'}_u^2 & {I'}_u {I'}_v \\ {I'}_u {I'}_v & {I'}_v^2 \end{bmatrix} \tag{1}$$

where ${I'}_u$ and ${I'}_v$ are the vertical and horizontal spatial gradients of the image intensities, respectively, so that the matrix would be both above the image noise level and well-conditioned [22]. The keypoints are chosen such that their smaller eigenvalue is higher than a threshold ($\min(\lambda_1, \lambda_2) > \lambda$). This threshold is set according to the image resolution and illumination, in order to compensate for part of the noise. To this end, the lower bound on λ would be determined by a region of the image with rather uniform brightness, while its upper bound is set based on a highly-textured region (see [22] for more detailed criteria).

After extracting the keypoints from the current frame, we need to track (i.e. match) them across frames in $\mathbb{S}2$. One of the distinguishing properties of the proposed method is the use of a temporal sliding window of 3 frames with gap $\Delta t(t,\, t+\Delta t$, and $t+2\Delta t)$ for background elimination, where the earliest captured frame (t) is always used as the non-constant reference for transformation along this process. By using such a sliding window, we aim at compensating the background motion at different scales, which causes the foreground segmentation error, while limiting the image registration error to the few frames that are currently being processed. Hence, a poor image registration cannot affect the detection quality of the whole sequence. To this end, the extracted keypoints of frame t are tracked over two successive frames of $t+\Delta t$ and $t+2\Delta t$. In this work, the parameter Δt (in terms of number of frames in the interval) is adjusted dynamically, so that the proposed algorithm can detect moving targets robustly in real-time. This value is set based on a series of parameters such as: frame per second (fps) rate of the video stream ($R_{(c)}$), UAV's altitude ($A_{(v)}$), algorithm computational complexity ($O_{(c)}$), and UAV's speed ($S_{(v)}$). As a rule-of-thumb, the relationships between these parameters and the efficient detection interval are shown in (2) and the appropriate value for Δt is discussed in more details in Section IV.

$$\Delta t \propto \frac{A_{(v)}}{S_{(v)}} O_{(c)} R_{(c)}. \tag{2}$$

It is noted that using an adjustable detection interval (more than $\Delta t = 1$) in the proposed framework helps to eliminate the inconsistent motion in lower camera speeds, due to slower relative motion of the targets with respect to camera, and hence, makes the algorithm more robust. The rationale to use 3 frames for the sliding window is that, while using less number of frames cannot adequately reduce the image registration error, using more than 3 frames requires higher computational resources and hence, may not be efficient for real-time applications. On the other hand, since the proposed method does not consider geometric constraints for foreground segmentation (as the approaches in [2], [6]), a sliding window of 3 frames is sufficient for effective moving target detection.

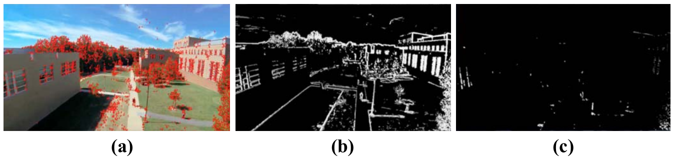


Fig. 2. The comparison of different transformation algorithms: (a) Resulted optical flow arrows; (b) Background subtraction via affine transformation, as in [11]; (c) Background subtraction via perspective transformation, in this work.

In this work, we make use of the sparse optical flow concept in order to solve the keypoints-matching problem by the pyramidal Lucas–Kanade (PLK) algorithm [23]. This algorithm considers the neighborhood of each detected keypoint of frame t and solves an over-constrained system of equations for estimating the displacement of such keypoints across subsequent frames $t + \Delta t$ and $t + 2\Delta t$. The final solution of this system is the keypoints displacement vector,

$$\begin{bmatrix} \mathrm{v}_u \\ \mathrm{v}_v \end{bmatrix} = \begin{bmatrix} \sum_i {I'}_{u_i}^2 & \sum_i {I'}_{u_i} {I'}_{v_i} \\ \sum_i {I'}_{u_i} {I'}_{v_i} & \sum_i {I'}_{v_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i {I'}_{u_i} {I'}_{t_i} \\ \sum_i {I'}_{v_i} {I'}_{t_i} \end{bmatrix} \tag{3}$$

where v_u and v_v are vertical and horizontal displacements of the keypoint, respectively. ${I'}_{u_i}$ and ${I'}_{v_i}$ are the spatial gradients along the vertical and horizontal axes for pixel i in the keypoint's neighborhood, and ${I'}_{t_i}$ is its time-based derivative between the two frames. The details of parameters setting for the PLK algorithm are discussed in the authors' previous work [11]. The rationale to use the pyramidal version of the algorithm is to capture larger motions by local window of keypoint's neighborhood at larger scales of the Gaussian pyramid of the image, while satisfying the spatial coherence assumption of the LK. Hence, PLK ensures that the three main assumptions of: brightness constancy, temporal persistence, and spatial coherence will not be violated, which are required for solving the tracking problem based on least square minimization. The PLK first tracks the keypoints over larger spatial scales of the pyramid and then refines the initial motion velocity assumptions through its lower levels to the raw image pixels. Hence, it can minimize the violations of assumptions, while tracking faster and larger motions for robustness.

Now that the reference frame's keypoints are tracked over the sliding window, we may use these matched pairs of points for estimating and later compensating the camera motion between successive frames. This process is implemented through image registration in $\mathbb{S}3$. To this end, we first need to transform (i.e. register) each frame onto the reference frame based on the camera motion estimation. The transformation can be performed through a variety of methods (e.g. affine and perspective), among which, we consider perspective transformation between each pair of frames, using Homography estimation. While an affine transformation can map a rectangle to any parallelogram, the perspective transformation is more general and transforms this rectangle to any trapezoid. Hence, it can register two different images as alternative projections of the same scene onto two different projective planes, in a robust manner. Fig. 2 compares the background subtraction results of applying affine transformation (as developed in [11], [18]) versus perspective transformation (as in this work), on an urban scene via a freely moving camera onboard UAV.

Fig. 2(a) depicts the magnified optical flow arrows on a grayscale frame from a monocular moving camera (onboard UAV), while Fig 2(b) represents the silhouette image as the result of taking absolute differences of two successive frames. In this image, the subsequent frame is registered on the original frame using the affine transformation. As shown in Fig. 2(b), there is a huge amount of the background segmented by error as foreground (the white edges in silhouetted image). This implies that the motion of the background (as the result of camera movement) cannot be adequately compensated, assuming an affine transformation between different images over time, even after filtering out the miss-tracking of the optical-flow module. Fig. 2(c) shows the same situation under a perspective transformation algorithm (as proposed in this work), which resulted in a more precise background subtraction. As the images are obtained by the perspective projection, a 3 × 3 Homography matrix represents the relationship between keypoints through a projective mapping from one plane (e.g. a frame) to another, and it is defined as

$$H = \left[h_{ij}\right], \quad where\ i = 1, 2, 3\ and\ j = 1, 2, 3. \tag{4}$$

In this work, we first estimate the Homography matrix between frames and then apply it for perspective warp of those frames onto a reference one in the sliding window framework. Since the extracted features include both foreground and background keypoints as well as noise, the initial step is to filter out the moving foreground keypoints and noise as outliers, before estimating the Homography matrix. The rationale to do so is that, in the low to medium density crowd scenes (as considered in this work), only the larger set of keypoints belonging to background can represent the camera motion model as inliers, while the fewer number of keypoints belonging to independently moving foreground targets are considered outliers, which do not fit the model. In this paper, we apply RANSAC filtering scheme towards a robust estimation, since it finds a solution with the largest inlier support (hence, an improved camera compensation). After filtering out the keypoints belonging to the foreground as outliers, the Homography can be estimated between frame t and $t + \Delta t$ as well as frames t and $t + 2\Delta t$. The relationships between these frames are shown as,

$$K_t = H_{t,t+\Delta t} K_{t+\Delta t}, K_t = H_{t,t+2\Delta t} K_{t+2\Delta t}. \tag{5}$$

In (5), K_t and $K_{t+c\Delta t}|c = 1, 2$ represent the homogeneous coordinates of the k refined keypoints (i.e. inliers) in frames t and $t + c\Delta t$, respectively, while $H_{t,t+c\Delta t}|c = 1, 2$ defines the unknown Homography matrix between frame $t + c\Delta t$ and frame t based on these points. Here, K_t (as well as $K_{t+c\Delta t}$) is in the form of

$$\begin{pmatrix} U_1(t) & U_2(t) & & U_k(t) \\ V_1(t) & V_2(t) & \ddots & V_k(t) \\ W_1(t) & W_2(t) & & W_k(t) \end{pmatrix};$$

$$\begin{cases} U_i(t) = u_i(t) \cdot W_i(t)\ \forall i = 1, \ldots, k \\ V_i(t) = v_i(t) \cdot W_i(t)\ \forall i = 1, \ldots, k \end{cases} \tag{6}$$

where $u_i(t)$ and $v_i(t)$ stand for the vertical and horizontal position of the i^{th} detected keypoint in frame t, and $W_i(t)$ is any arbitrary scalar (due to homogeneous coordinates), which can be set to 1, without loss of generality. In general, the unknown Homography parameters of (5), in the vector format of $\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^{\text{T}}$, can be estimated by solving $\mathbf{Ah} = 0$, using homogeneous linear least squares, where $\mathbf{A}(t + c\Delta t)$ is defined as, (7), as shown at the bottom of this page.

After estimating the unknown elements of $\mathbf{h}$, at the image registration step, we warp each of frames $t + \Delta t$ and $t + 2\Delta t$ onto frame t based on perspective transformation of all their pixels, using the estimated Homography matrices in (5),

$$I^{(T)}_{t+\Delta t} = H_{t,t+\Delta t} I_{t+\Delta t}, \quad I^{(T)}_{t+2\Delta t} = H_{t,t+2\Delta t} I_{t+2\Delta t} \tag{8}$$

where $I^{(T)}_{t+c\Delta t}|c = 1, 2$ represents the new pixel values of the transformed frame $t + c\Delta t$, after being warped into frame t.

B. Moving Targets Segmentation

Now that we have access to the two warped images at time $t + c\Delta t|c = 1, 2$ (i.e. $I^{(T)}_{t+\Delta t}$ and $I^{(T)}_{t+2\Delta t}$), we may eliminate the background by taking the absolute differences of these two perspective transformed images in $\mathbb{S}4$. The resulting image would be computed as

$$\Delta I_t = \left| I^{(T)}_{t+2\Delta t} - I^{(T)}_{t+\Delta t} \right|. \tag{9}$$

It is noted that because of using two transformed frames in our sliding window approach, the background motion compensation would be more effective, where the registration error is minimized due to using an independent reference image for transforming the successive frames. Next, a threshold is applied on ΔI_t to get rid of the shadowing regions and creating the silhouette mask of the potential moving foreground (see Fig. 2(c)). Depending on the camera movement direction and speed from frame to frame, we may experience uniformly

$$\begin{bmatrix} -u_1(t+c\Delta t) & 0 & & -u_k(t+c\Delta t) & 0 \\ -v_1(t+c\Delta t) & 0 & & -v_k(t+c\Delta t) & 0 \\ -1 & 0 & & -1 & 0 \\ 0 & -u_1(t+c\Delta t) & & 0 & -u_k(t+c\Delta t) \\ 0 & -v_1(t+c\Delta t) & \ddots & 0 & -v_k(t+c\Delta t) \\ 0 & -1 & & 0 & -1 \\ u_1(t).u_1(t+c\Delta t) & v_1(t).u_1(t+c\Delta t) & & u_k(t).u_k(t+c\Delta t) & v_k(t).u_k(t+c\Delta t) \\ u_1(t).v_1(t+c\Delta t) & v_1(t).v_1(t+c\Delta t) & & u_k(t).v_k(t+c\Delta t) & v_k(t).v_k(t+c\Delta t) \\ u_1(t) & v_1(t) & & u_k(t) & v_k(t) \end{bmatrix}^{\text{T}}. \tag{7}$$

segmented thin bars on the image boundaries because of registration and warping. An instance is shown in Fig. 1, where black bars are generated in step $\mathbb{S}3$. As these boundary bars are results of camera movements, in this work, we disregard them as background and set the value of corresponding pixels to zero in $\mathbb{S}4$, to preserve persistency and robustness.

At the final step $\mathbb{S}5$, we differentiate and segment multiple independently moving targets based on the local motion of their detected blobs. However, there is always a need to smoothen the image by applying a filter (e.g. Gaussian, median, and box kernel) as post processing task to account for the imaging noises caused by the camera. As a low-pass filter, Gaussian mask reduces the image high-frequency components, and hence, is commonly used for edge refinement in target detection applications to improve algorithm performance. In this work, we apply an n_g by n_g Gaussian kernel for a linear convolution on the silhouette mask. The size of the kernel needs to be large enough to cover most of the segmented blobs, but not so large that multiple blobs are overlapped at a time; hence, independently moving targets regions can be separated later in the process. Another source of remained noise, that we need to handle before segmenting the moving foreground, comes from the camera motion estimation error. Applying a smoothing filter would not eliminate this type of noise. Therefore, we apply the connected-components analysis for completing and improving the background elimination phase, while clustering the closely moving foreground regions that are potentially parts of a unified target (e.g. a group of people).

Throughout this process, the morphological operations (e.g. erosion and dilation) are being used on the mask image, to shrink areas of small noise to zero, followed by rebuilding the area of surviving components (i.e. segmented foreground regions) that was lost in the previous operation. In other words, erosion and dilation help in removing separate noises and filling the holes in the segmented blobs belonging to the same unified moving targets in the foreground, respectively. The reason is that by taking the absolute differences between successive frames, mostly edge regions of the moving targets could be segmented due to the slight movement, when comparing adjacent frames. Therefore, by applying morphological operations, we have "large enough" blobs of the surviving foreground regions and can proceed to detect those segments as multiple unified moving foreground targets.

Now, as the last procedure, the independently moving targets regions can be separated using a local motion history function, which will enable tracking the segmented blobs over time. This motion history function uses floating point images to represents the motion template. A set of these images form a representation of the overall motion, by taking the gradient of the silhouette image over time. At every new frame, such function would be first updated based on the newly segmented foreground. Next, the motion gradient and orientation parameters of every single region are calculated based on the spatio-temporal approach in our previous work [11] to estimate the general movement direction of each moving target. Finally, a motion segmentation routine (as discussed in [18]) is applied to separate independently moving targets based on their local motions. This completes our method on effectively and robustly detecting independently moving targets from a moving camera in an efficient manner. We may also assign separate target boundaries for each independent detected blob for display purposes. It is noted that, since the kernel size of the post processing task may affect performance of the local motion history function in segmenting multiple targets, experimental analysis on selecting the appropriate parameters for a general application is provided in Section IV.

IV. Results and Discussion

In this work, a visual surveillance case study is considered to validate and demonstrate the proposed method. An RGB camera with a gimbal is mounted on a 3DR® X8$^+$ drone (i.e. UAV), and experiments are conducted with both vertical and oblique views of the environment. We also used an ODROID® U3 Linux computer with 1.7 GHz Quad-Core processor and 2 GB RAM onboard the UAV for near real-time processing of the data to analyze algorithm efficiency. Using this testbed, we evaluated the proposed method on different scenarios of crowd movements and at different camera setups (i.e. speed, altitude, and view-angle), to verify its effectiveness and robustness in segmenting the moving targets (see Fig. 3). Fig. 3(a) shows three different time snapshots of a scenario in which the movement of a crowd of 4 is captured by the UAV's onboard camera, flying at a low altitude and a rather high speed; hence, the detection interval is set as low as 1 to provide better performance. As shown in these image series, the segmented blob can be used to represent a crowd of people as one unified target (the middle image) or individual targets (the top image), depending on their proximity and motion. Fig. 3(b) shows a scenario of a crowd splitting to two groups, captured at a lower speed; hence, Δt is adjusted to 2 for this scenario. As depicted in the bottom image of this figure, a bicyclist with a faster speed entered the camera detection range (top-left corner), and hence, is segmented with a longer motion trail (due to the local motion history function). It is noted that by setting the appropriate detection interval based on the proposed heuristic, targets with different movement velocities can be detected in a robust manner. Finally, Fig. 3(c) shows the detection results for a scenario of 5 people scattered in different directions, captured at higher altitude and with $\Delta t = 3$. Fig. 4 also compares the results of the proposed method with our previously developed algorithm [11] on a scenario of crowd control at two different altitudes, to verify the achieved improvement as a result of applying perspective, rather than affine transformation.

We applied our proposed method on a series of available datasets related to surveillance and ITS applications to verify its robustness and compare its performance with existing methods in the literature, whose results are publicly available. The selection of datasets has been based on introducing a variety of challenges to the problem by considering different camera setups and movements as well as various sizes, shapes, and speeds for targets. The datasets considered in this work include: Dataset1 (DARPA VIVID-EgTest05, featuring a group of cars tracked along a road via aerial camera) [24], Dataset2 and Dataset3 (featuring traffic scenes in an urban

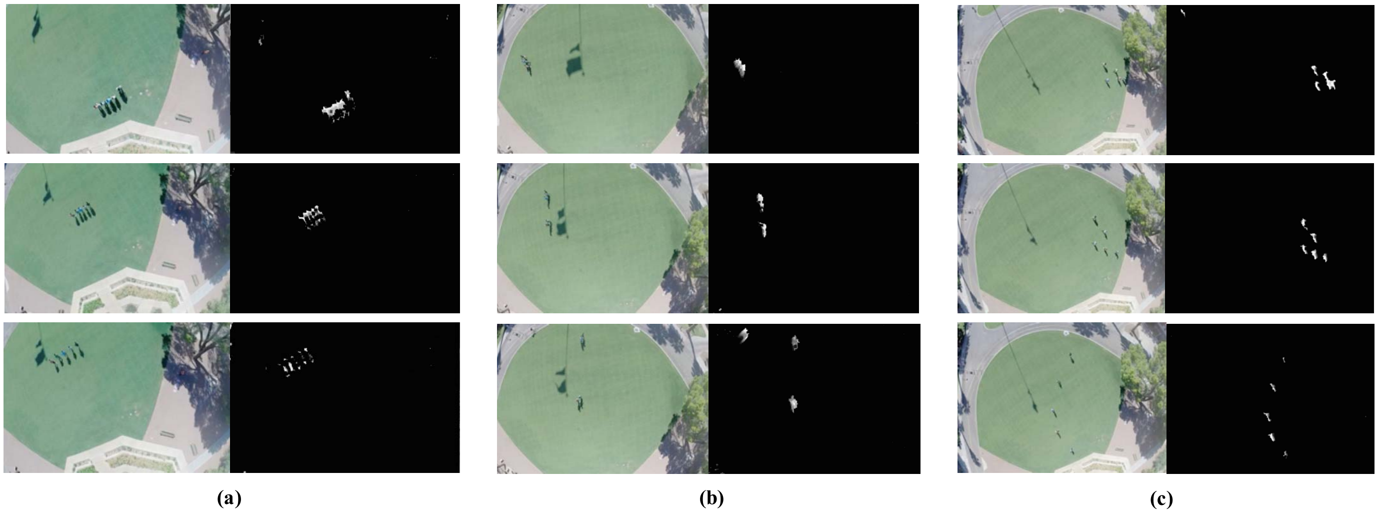


Fig. 3. The results of applying the proposed method on videos of different scenarios captured by UAV: (a) The scenario for a crowd of 4 people moving together; (b) The scenario for a crowd of 4 people splitting into two groups and a bicyclist passing by (at faster speed); (c) The scenario for a group of 5 people scattering.

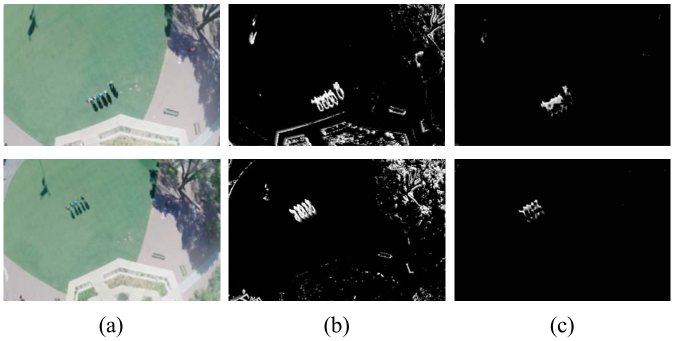


Fig. 4. Comparison with the algorithm previously developed by authors at two different altitudes: (a) Original video frames; (b) Results of algorithm in [11]; (c) Results of the proposed method.

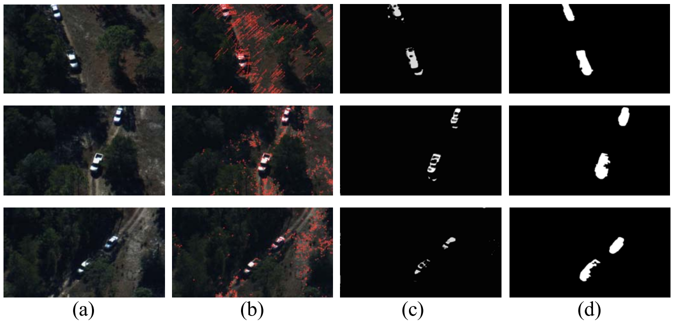


Fig. 5. The results of applying the proposed method on Dataset1, at three frames: t=185; t=329; t=521: (a) Original frames; (b) Optical flow vectors; (c) Detected foreground; (d) Ground-truth blobs [17].

area [19] and from the Hopkins 155 dataset [25], captured via moving hand-held cameras), Dataset4 (featuring a person movements captured via axial rotation of the camera) [17], Dataset5 (UCF Aerial Action dataset, featuring 3 cars on a road captured via extreme shift of aerial camera) [26]. Fig. 5 shows the experimental results of applying the proposed method on Dataset1. As shown in this figure, our method has successfully segmented the independently moving vehicles, despite the encountered challenges, such as rapid rotary motions of the aerial camera, illumination changes, and highly textured background. Dataset2 and Dataset3 are also used to provide different view-angles for the ITS-related applications, so that the robustness of the proposed method can be evaluated. Fig. 6 illustrates the comparisons of results with the ground-truth data of Dataset2, as well as other existing methods in the literature (i.e. multi-layer Homography (MLH) [16], BMS [17], particle video (PV) [19], moving camera background subtraction (MCBS) [27], and segmentation with effective cue (SEC) [28]). As shown in this figure, the qualitative results provided by the proposed method outperforms those reported by the existing methods, in terms of foreground segmentation accuracy. The main reason is that our algorithm estimates the general background motion, rather than reconstructing the camera motion by interpolation, as in BMS. MLH also missed to segment the whole moving regions and has focused on the motion trajectories. Moreover, both MLH and MCBS show some false positives in the left boundaries of the images due to camera movements. Finally, PV and SEC methods provide fuzzy segmentation of the moving targets, while our method presents an accurate segmentation.

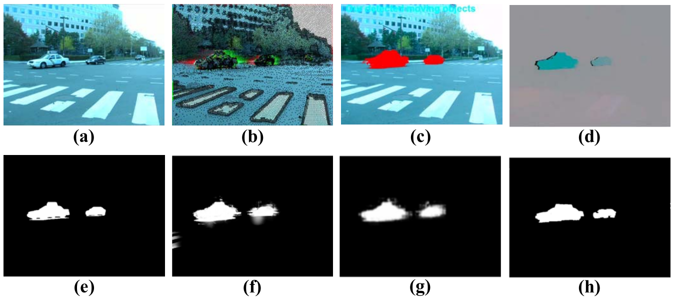


Fig. 6. Comparison with exiting methods on Dataset2: (a) Original frame; (b) Results of MLH; (c) Results of BMS; (d) Results of PV; (e) Ground-truth data; (f) Results of MCBS; (g) Results of SEC; (h) Results of our proposed method.

We have also reported results based on several quantitative performance evaluation metrics, related to target detection and segmentation. The measures considered in this work include

TABLE I

QUANTITATIVE PERFORMANCE ANALYSIS ON THE PROPOSED METHOD

Dataset	Proposed Method		BMS Method [17]		MD [4]
	F-measure	Gmean	F-measure	Gmean	F-measure
Dataset1	0.69	**0.86**	**0.85**	**0.86**	0.74
Dataset2	0.84	**0.93**	**0.90**	0.91	0.19
Dataset3	0.80	**0.89**	**0.84**	0.85	0.55
Dataset4	**0.75**	**0.91**	**0.75**	0.90	0.67
Dataset5	0.70	**0.86**	**0.79**	0.80	0.70
Average	**0.76**	**0.89**	**0.82**	0.86	0.57

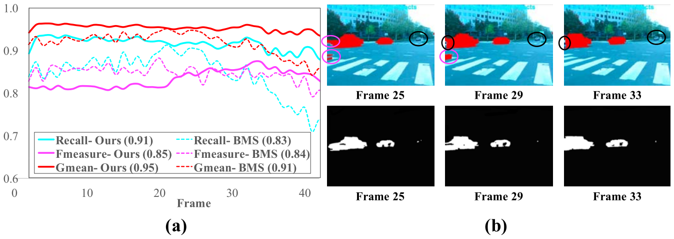


Fig. 7. Performance comparison with BMS on parts of Dataset2: (a) Mean and standard deviation of the metrics; (b) Results of applying the two methods on a series of frames to justify better performance of the proposed method on Recall.

Accuracy, Specificity, Precision, and Recall, which are defined as follows:

$$\text{Accuracy} = \frac{Tp + Tn}{N}, \quad \text{Specificity} = \frac{Tn}{n},$$
$$\text{Precision} = \frac{Tp}{Tp + Fp}, \quad \text{Recall} = \frac{Tp}{p} \tag{10}$$

where p is the number of positive (foreground) pixels and n is the number of negative (background) pixels reported by the detection algorithm, Tp is the number of true positives and Tn is the number of true negatives in the test frame compared to the ground truth, Fp is the number of false positives and Fn is the number of false negatives in the test frame, and finally, N is the total number of pixels in the test frame, depending on the image resolution. Furthermore, two composite performance metrics, F-measure and Gmean [29], are also considered as compromises between previous measures and can be used as the criteria to compare the performance with other existing methods. These metrics are defined as

$$\text{F} - \text{measure} = \frac{2 \times Precision \times Recall}{Precision + Recall},$$
$$\text{Gmean} = \sqrt{Specificity \times Recall}. \tag{11}$$

Table I summarizes quantitative performance analyses on the proposed method compared to some other methods which have reported their results on the datasets considered in this work. Although the metrics in (10) and (11) are usually measured for evaluating the performance of pixel-level segmentation algorithms, some of them can be simply applied to evaluate moving targets detection performance. As an instance, Recall can be defined as the ratio of truly detected targets to the total number of moving targets. Since the proposed algorithm successfully detected all moving targets on the considered datasets (though partially, in some cases), its detection-based Recall is 1.0. Therefore, in this work, we focus on evaluating the results in terms of moving foreground segmentation, for which, the proposed motion detection method shows a comparable and mostly promising performance versus state-of-the-art segmentation methods in the literature. According to Table I, the results of our method outperforms the BMS results on all datasets in terms of Gmean (by 3.5% on average) and we also obtained an average improvement of 33% on F-measure values provided by motion decomposition (MD) method [4]. It is noted that the proposed method also significantly improves the efficiency compared to these methods in terms of very shorter run-time. More specifically, our method generates final results in less than 0.2 sec/frame on average for the considered datasets (see Fig. 9), while BMS takes at least 1.03 sec/frame (disregarding optical flow step) and MD takes 1.89 sec/frame on a 3.4 GHz CPU with 32 GB RAM.

Fig. 7 demonstrates the detailed performance of proposed method in terms of criteria provided in (10) and (11), when applying on a portion of Dataset2. Fig. 7(a) verifies that our proposed method has higher values of Recall and Gmean on average, while F-measure values are higher than those of BMS after a certain point (frame 25). Moreover, the variations of these metrics are lower compared to the BMS method. The provided results are promising, since the BMS algorithm is designed to accurately classify the foreground and background in the pixel level, and hence, the Precision and F-measure of the BMS are expected to be generally higher. It is noted that the Recall performance for the BMS method drops dramatically after frame 25. The reason could be that the front car (at the left) starts to exit the camera view and the algorithm of pixel classification using bi-level segmentation (coarse and fine) cannot handle boundary pixels very well. However, our algorithm can address such issues due to its uniform keypoint extraction via PLK and perspective registration of frames in a sliding window, which is robust to changes in velocity and location of the moving target. Moreover, the third car at the far right is moving at a slower rate, for which, the BMS algorithm did not classify any of its pixels as moving foreground, while our method partially did so. Fig. 7(b) illustrates the qualitative results of the two algorithms on frames 25 to 33 of this dataset for clarification. As shown in this figure, the proposed method can detect 3 out of 3 moving targets, hence its Recall on moving target detection is 1.0, while BMS detects only 2 targets at a lower Recall value. We have highlighted the false positive and false negative regions on the BMS results by magenta and black circles, respectively. Note that in these image series, increased false positives results in greater $Tp + Fp$, which causes Precision value decreases ($Tp/(Tp + Fp)$), while increased false negatives results in smaller Tp, which ends up reducing the Recall value (Tp/p).

We have also provided empirical analyses on the optimal values for the parameters considered in the proposed method. Fig. 8 represents the results of these studies on videos recorded from UAV camera. As shown in Fig. 8(a) the proposed algorithm is not very sensitive to the values of Gaussian kernel

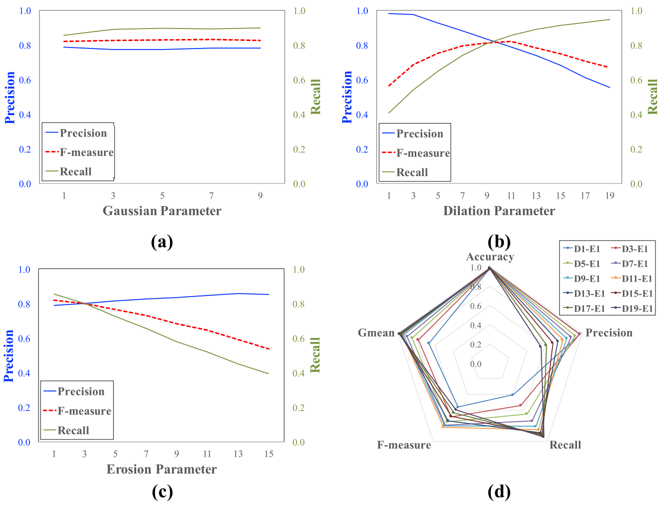


Fig. 8. Sensitivity Analysis of the performance metrics on key parameters: (a) Based on different values of Gaussian kernel parameter; (b) Based on different values of Dilation parameter; (c) Based on different values of Erosion parameter; (d) Two-way analysis based on Dilation and Erosion parameters.

size in the post processing task; hence, in this work, we set n_g between 3 to 7. The graphs in Fig. 8(b) and Fig. 8(c) show the evaluation results based on variation in the morphological operations (i.e. dilation and erosion) parameters. As depicted in the figures, increasing the dilation kernel size reversely affects the Precision performance, while the Recall performance gets improved. The opposite is true for Erosion kernel, where Precision slightly improves by increasing the parameter and Recall performance deteriorated at the same time. Depending on the camera distance (i.e. altitude) and the size of target of interest, the selection of these parameters may vary slightly; however, similar parameter setting and patterns of performance variation apply to Dataset1 and Dataset5, recorded from aerial cameras. Although depending on the application, one of the two performance metrics might be of higher interest for improvement, in this work, parameters with the highest F-measure and Gmean values are considered optimal for the proposed method (e.g. values between 1 to 3 for erosion kernel size). Finally, Fig. 8(d) shows a complete two-way analysis based on dilation and erosion parameters for all different performance metrics in this work. The same discussion about parameter setting based on performance metric of interest (i.e. F-measure and Gmean), holds here.

As a key parameter, the value of detection interval also needs to be adjusted based on the empirical relationship provided in (2). In this work, since the computational complexity of the proposed algorithm only differs based on the image resolution (see Fig. 9), the sensitive parameters for setting the detection interval are altitude and speed of UAV, as well as video streaming rate. Therefore, we initialize the value of Δt at 1 frame (i.e. processing every successive frame in the sliding window) for the initial values of $R_{(c)} = 10(fps)$, $A_{(v)} = 20(m)$ and $S_{(v)} = 5\,(m/s)$. These values are set based on a series of experiments on different videos captured by UAV's onboard camera, to find the best combination for more accurate results. As the values of these parameters change, the detection interval can be adjusted to higher values for robustness. The results of applying different values of Δt have been discussed previously, through Fig. 3 and Fig. 4.

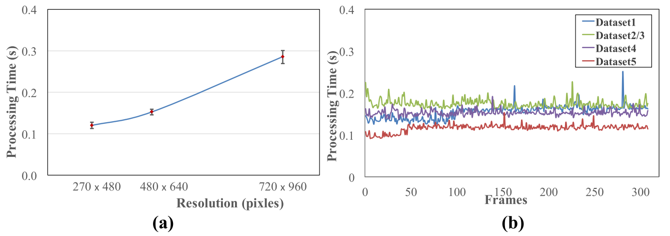


Fig. 9. Computational cost analysis: (a) Mean and standard deviation of the processing time based on the resolution; (b) Time analysis of different datasets.

Finally, the computational complexity of the proposed method is studied for the sake of efficiency evaluation. Fig. 9 shows the time analysis in terms of the processing time per frame based on different image resolutions. As discussed earlier in this section, the average processing time for our method is less than 0.2 sec/frame, which is very promising with respect to the limited computational resources onboard the UAV. As shown in Fig. 9(a), even for resolutions as high as 720 × 960, the proposed method is still efficient for near real-time implementation. Fig. 9(b) also provides the detailed processing times for different datasets considered in this work.

V. Conclusions and Future Work

In this work, we proposed an effective, and efficient method for detecting multiple independently moving targets from a monocular moving camera onboard UAV in a robust manner. A sliding-window framework is considered, where at each time frame t, the keypoints are extracted and tracked onto the next two frames with gap Δt. These frames are then registered through perspective transformation onto frame t. Finally, a local motion history function is applied after post processing operations, to separate the independently moving targets.

We tested our method using videos captured by UAV as well as publically available datasets. The experiments on different scenarios demonstrated promising results based on the quantitative and qualitative evaluations. More specifically, the effectiveness of the proposed method is evaluated by considering results on different camera setups (in terms of altitude, speed, and view-angle) and various applications; its effectiveness is verified through comparisons with ground-truth data as well as state-of-the-art methods, while reporting the achieved performance in terms of common performance metrics; and the method's efficiency is demonstrated by computational time analyses and compared with reported run-times of existing methods. Sensitivity analysis studies have also provided for optimal setting of the key parameters in the proposed method.

As a future research work, we aim at proposing a robust data association algorithm to differentiate and associate multiple detected targets over a sequence of video frames for the application of target tracking through UAVs, while considering various surveillance scenarios.

References

[1] K. A. Joshi and D. G. Thakore, "A survey on moving object detection and tracking in video surveillance system," *Int. J. Soft Comput. Eng.*, vol. 2, no. 3, pp. 44–48, 2012.

[2] G. Zhang, J. Jia, W. Xiong, T.-T. Wong, P.-A. Heng, and H. Bao, "Moving object extraction with a hand-held camera," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[3] J. Kang, I. Cohen, and G. Medioni, "Continuous tracking within and across camera streams," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2003, pp. I-267–I-272.

[4] S. Wu, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 1419–1426.

[5] C.-C. Lin and M. Wolf, "Detecting moving objects using a camera on a moving platform," in *Proc. 20th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2010, pp. 460–463.

[6] S. Dey, V. Reilly, I. Saleemi, and M. Shah, "Detection of independently moving objects in non-planar scenes via multi-frame monocular epipolar constraint," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 860–873.

[7] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2846–2854.

[8] Q. Wang, J. Gao, and Y. Yuan, "A joint convolutional neural networks and context transfer for street scenes labeling," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[9] Q. Wang, J. Gao, and Y. Yuan, "Embedding structured contour and location prior in siamesed fully convolutional networks for road detection," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[10] Y. Yuan, Y. Lu, and Q. Wang, "Tracking as a whole: Multi-target tracking by modeling group behavior with sequential detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3339–3349, Dec. 2017.

[11] S. Minaeian, J. Liu, and Y.-J. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 1005–1016, Jul. 2016.

[12] M. Chen, Q. Wang, and X. Li, "Anchor-based group detection in crowd scenes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 1378–1382.

[13] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov, "Bilayer segmentation of live video," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 53–60.

[14] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1305–1312.

[15] J. Kang, S. Kim, T. J. Oh, and M. J. Chung, "Moving region segmentation using sparse motion cue from a moving camera," in *Intelligent Autonomous Systems 12*. 2013, pp. 257–264.

[16] Y. Jin, L. Tao, H. Di, N. I. Rao, and G. Xu, "Background modeling from a free-moving camera by multi-layer homography algorithm," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 1572–1575.

[17] Y. Wu, X. He, and T. Q. Nguyen, "Moving object detection with a freely moving camera via background motion subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 2, pp. 236–248, Feb. 2017.

[18] S. Minaeian, J. Liu, and Y.-J. Son, "Crowd detection and localization using a team of cooperative UAV/UGVs," in *Proc. IIE Annu. Conf.*, 2015, pp. 595–604.

[19] P. Sand and S. Teller, "Particle video: Long-range motion estimation using point trajectories," *Int. J. Comput. Vis.*, vol. 80, no. 1, pp. 72–91, 2008.

[20] D. Zhang, O. Javed, and M. Shah, "Video object segmentation through spatially accurate and temporally dense extraction of primary object regions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 628–635.

[21] T. Ma and L. J. Latecki, "Maximum weight cliques with mutex constraints for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 670–677.

[22] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1994, pp. 593–600.

[23] J.-Y. Bouguet, "Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm," *Intel Corp.*, vol. 5, no. 4, pp. 1–10, 2001.

[24] R. Collins, X. Zhou, and S. K. Teh, "An open source tracking testbed and evaluation Web site," in *Proc. IEEE Int. Workshop Perform. Eval. Tracking Surveill. (PETS)*, vol. 35. Jan. 2005, pp. 1–8.

[25] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.

[26] University of Central Florida. *UCF Aerial Action Data Set*. Accessed: Jan. 2017. [Online]. Available: http://crcv.ucf.edu/data/UCF_Aerial_Action.php

[27] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1219–1225.

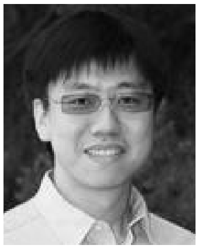
[28] M. Wu, X. Peng, and Q. Zhang, "Segmenting moving objects from a freely moving camera with an effective segmentation cue," *Meas. Sci. Technol.*, vol. 22, no. 2, p. 025108, 2011.

[29] H. Guo and H. L. Viktor, "Learning from imbalanced data sets with boosting and data generation: The DataBoost-IM approach," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 30–39, 2004.

Sara Minaeian received the M.S. degree in industrial engineering from the University of Tehran, Tehran, Iran, in 2008, and the M.S. degree in systems engineering and the Ph.D. degree in systems and industrial engineering from The University of Arizona, Tucson, AZ, USA, in 2015 and 2017, respectively.

Since 2013, she has been a Research Assistant with the Computer Integrated Manufacturing and Simulation Laboratory, The University of Arizona. Her research interests include computer vision, dynamic data driven application systems, and distributed multi-paradigm simulation.

Dr. Minaeian is a member of IISE, INFORMS, and ACM-SIGSIM.



Jian Liu received the B.S. and M.S. degrees in precision instruments and mechanology from Tsinghua University, Beijing, China, in 1999 and 2002, respectively, and the M.S. degree in statistics and the Ph.D. degree in industrial and operations engineering from the University of Michigan, Ann Arbor, MI, USA, in 2006 and 2008, respectively.

He is currently an Associate Professor with the Department of Systems and Industrial Engineering, The University of Arizona. His research interests focus on the integration of manufacturing engineering knowledge, control theory, and advanced statistics for quality, reliability, and productivity improvement.

Dr. Liu is a member of INFORMS and IISE.

Young-Jun Son received the B.S. degree in industrial engineering from Pohang University of Science and Technology, Pohang, South Korea, in 1996, and the M.S. and Ph.D. degrees in industrial and manufacturing engineering from Pennsylvania State University, State College, PA, USA, in 1998 and 2000, respectively.

He is currently a Professor and the Department Head of Systems and Industrial Engineering, The University of Arizona, and the Director of the Advanced Integration of Manufacturing Systems and Technologies Center. His research focuses on the coordination of a multiscale, networked-federated simulation and decision model needed for design and control in manufacturing enterprise, renewable energy network, homeland security, agricultural supply network, and social network.

Dr. Son is a fellow of IISE. He has received several research awards, such as the SME 2004 Outstanding Young Manufacturing Engineer Award, the IIE 2005 Outstanding Young Industrial Engineer Award, the IERC Conference Best Paper Awards in 2005, 2008, 2009, and 2016, and the Best Paper of the Year Award from IJIE in 2007.