Noname manuscript No.
(will be inserted by the editor)

AURORA: Autonomous Real-time On-board Video Analytics

Plamen Angelov · Pouria Sadeghi-Tehran · Christopher Clarke

Accepted: March 2016

Abstract In this paper, we describe the design and implementation of a computationally efficient system for detecting moving objects on a moving platform which can be deployed on small, lightweight, low-cost and power-efficient hardware. The primary application of the payload system is that of performing real-time on-board autonomous object detection of moving objects from videos stream taken from a camera mounted to an Unmanned Aerial Vehicle (UAV). The implemented object detection algorithms utilise Recursive Density Estimation (RDE) and Evolving Local Means (ELM) clustering to perform change and object detection of moving objects without prior knowledge. Furthermore, experiments are presented which demonstrate that the introduced system is able to detect, by on-board processing, any moving objects from a UAV in real-time while at the same time sending only important data to a control station located on the ground with minimal time to setup and become operational.

Keywords Autonomous objects detection · unmanned aerial vehicle · evolving clustering · video analytic · linear motion model

1 Introduction

In this paper, we propose an autonomous real-time and power-efficient approach for on-board moving object detection using a mounted camera on a moving platform. The novelty of the proposed work is the combination of camera motion estimation with a fast, recursive background subtraction technique to enable real-time moving object detection on an aerial platform. The detection of moving objects has been studied extensively due to its role in a variety of applications such as surveillance [14], mobile navigation [28], and traffic control [11]. The study of Unmanned Aerial Vehicles

School of Computing and Communications, Data Science Group
Lancaster University, , United Kingdom, LA1 4WA
Tel.: +44 (0)1524 510391
Fax: +44 (0)1524 510489
E-mail: p.angelov@lancs.ac.uk

(UAV) has been an active topic among researchers because they can be utilised in a wide variety of applications and scenarios which are not accessible by humans. Video footage taken from UAVs has been one of the fastest growing data sources in the last few years, and this is increasing due to the development of small, low-cost high definition cameras.

The most common method for dynamic image analysis is background subtraction which uses background modelling for detecting moving objects. Background subtraction techniques detect objects by detecting statistically significant changes from a background model and can successfully separate foreground objects from the background assuming the camera is stationary or its motion is fully compensated by other means. Background subtraction is mostly used with stationary cameras because camera movements require the background to be remodelled. When the camera is also in motion, pixels with the same nominal positions in two (or more) consecutive image frames are no longer comparable without further computationally complex processing.

A background compensation method proposed in [19] can be used to calculate the background motion from the camera pan and tilt angles. However, the compensation of pan and tilt angles is only restricted to rotation of the camera about the lens centre. Zhang et. al [29] used dense optical flow fields over multiple frames to estimate the camera motion and segmentation of moving objects. However, inconsistent object boundaries in optical flow methods cause distortion of the moving objects or split articular motion of a moving object into more than one object. On the other hand, in [12] a hybrid model based on colour segmentation and motion based regions is used. The method provides good results but it is computationally expensive and is not real-time.

Several approaches have been proposed for object detection on UAV platforms; however, many of them are limited to only track pre-defined objects such as vehicles [22] or humans [26]. The advantage of the proposed system is that it requires no previous knowledge of the objects to be detected. As a result no training is required which enables fast deployment in a wide range of environments.

Most of the alternate techniques for object detection on UAVs involve sending a video data stream back to a ground station which is then processed on a high-end desktop computer, e.g. [20]. The major drawback of using this technique is the requirement for a fast and reliable data connection between the UAV and the ground station. The video signal is usually transmitted with a considerable delay and the possibility of corrupted data being received at the ground station due to the requirement to send every frame. As a result, this limits the operational range of the UAV because it must always be capable of transmitting the video stream for further analysis.

Computer hardware has become more powerful, smaller, and lighter, which allows for object detection algorithms to be implemented directly on-board the UAV. The advantage of using on-board processing is the significant reduction in the amount of transmitted data and minimisation of the delay between detection and analysis of objects. There is also a reduced risk of interception, increased range of operation, and reduction in workload for human operators located on the ground because they are only presented with video frames that may contain items of interest.

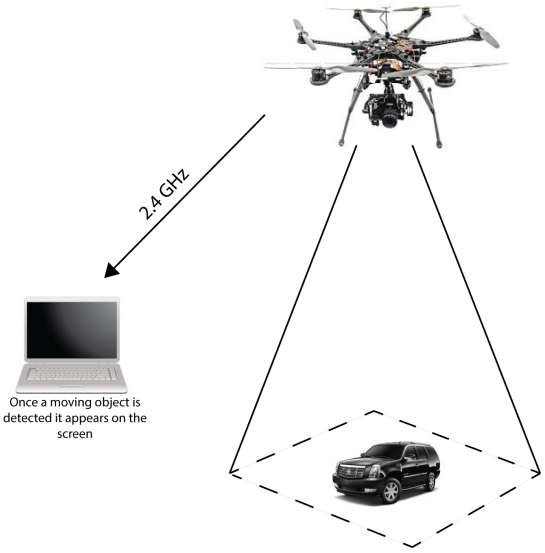

Fig. 1 Schematic representation of the proposed AURORA approach

The main focus of this paper is to implement a system that is able to perform real-time moving object detection on-board the UAV by using fast and recursive algorithms [1,2] without the need for prior knowledge of the objects to be detected. Such a system should be able to consistently detect moving objects under different scenarios before downlinking the processed and analysed image to a control station located on the ground once, and only if, moving objects are identified (Figure 1).

2 Methodology overview

The main challenge to address when detecting moving objects from an airborne UAV is the need to differentiate between the changes in the frame caused by the movement of the UAV from those caused by moving objects. This problem is not limited to aerial platforms, however it represents an additional difficulty because of the increase in the number of degrees of freedom in comparison to ground based vehicles.

We aimed to overcome these challenges by further developing and improving the recently introduced ARTOD approach, proposed in [21]. The whole process is performed in three main phases and utilizes computationally efficient algorithms to achieve real-time performance without the need for prior knowledge of the objects to be detected. The first phase is used to estimate the motion of the camera using salient features from the image and to compensate for the motion using, for example, homography or an optical flow based method; we use homography derived from optical flow in this work for a balance between execution time and accuracy. The second phase concerns change detection in which novelties must be identified between consecutive images; recursive density estimation (RDE) was chosen due to the reduction

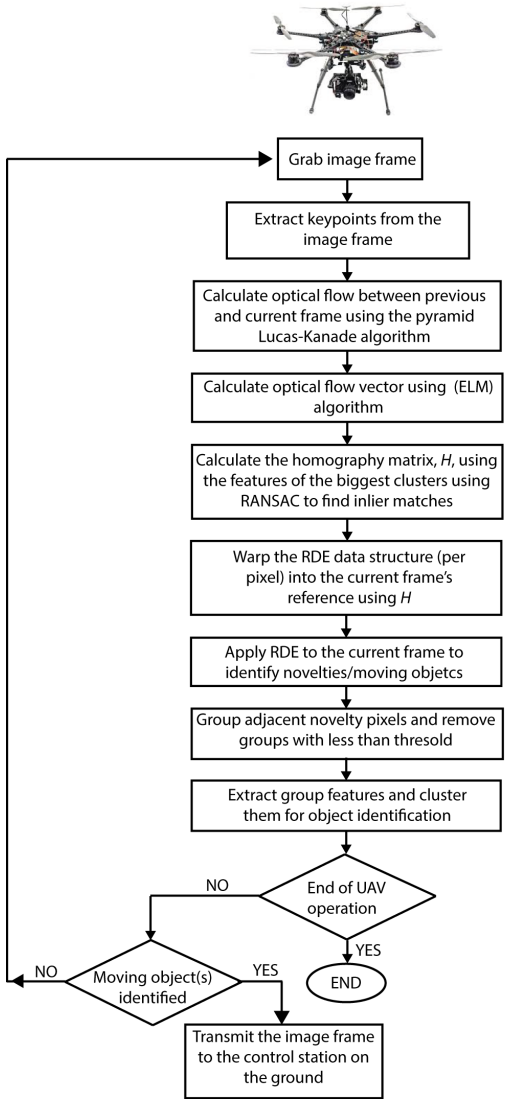

Fig. 2 Flow diagram of the proposed approach

in computational complexity compared with background subtraction techniques using a window of frames or a Gaussian Mixture Model (GMM) [18]. In the third phase objects are detected based on the changes detected in the second phase; a recursive clustering algorithm, Evolving Local Means (ELM), combined with neighbouring adjacent pixels is used to group novelties and detect moving objects from the video frame (Figure 2).

Autonomous video analytics can be seen as a hierarchical layered process (Figure 3). In this paper, we focus on the first layer (lower layer) for autonomous real-time on-

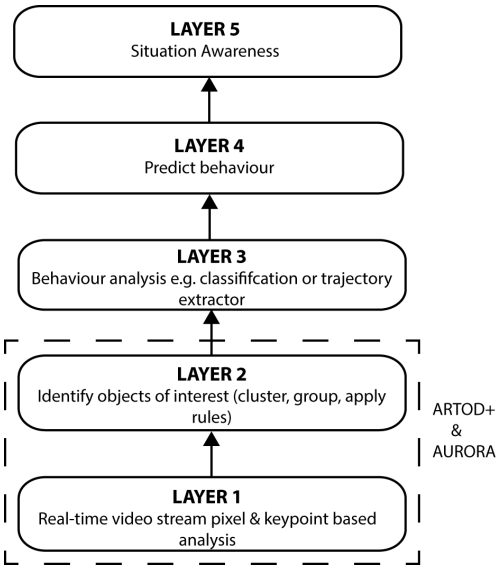

Fig. 3 Autonomous video analytics schematic structure

board video analytics. It would be possible to implement the higher-level layers once the ground station had received the video frames containing the objects of interest, once again reducing the computational burden due to only a subset of frames needing further processing.

2.1 Camera Motion Estimation and Compensation

In order to to apply background subtraction methods to moving camera videos several techniques have been proposed which compensate for the camera motion e.g. [8, 13, 27]. Some compensate for the camera motion offline and only perform well when there are no significant camera orientation changes [25]. These assumptions severely limit the operational range of these techniques and are not applicable to images captured from a UAV with more degrees of freedom.

The RDE algorithm is able to estimate the density of a pixel from the current image frame based on the similarity of the pixel at the respective position to **all** previous frames [3] or as many or as little as desired (RDE can be restarted or refreshed when required). We differentiate restarting and refreshing RDE as follows: restarting RDE resets all the values and the algorithm starts afresh with no prior data; refreshing RDE allows for a minimum number of prior frames to be used when the algorithm is reset. In order to refresh RDE there are two instances of the RDE algorithm running in parallel but only one is used for detection at any given time. The refresh value, n, is the minimum number of prior frames to be used for detection when the algorithm is to be reset. Every n frames the instance of RDE used for detection is changed to the other instance and the former RDE instance restarts. This guarantees at least n

frames worth of data are present for detection (where available). This is beneficial for the AURORA system due to the errors introduced by the homography process.

In the previously proposed ARTOD method [21], a homography is calculated, derived from the camera motion of each consecutive frame. This can be used to transform the new frame to the background model?s coordinate frame. Scale Invariant Feature Transform (SIFT) [17] was used to detect feature points in image sequences and to estimate the transformation homography between two consecutive frames. SIFT is invariant to image scaling, rotation illumination changes and 3D camera viewpoint. Usually, hundreds of keypoint matches are extracted from each frame; however, among extracted keypoint matches some result in mismatches. On the other hand, some of the keypoints can be from moving object(s) which do not represent the static background scene. In order to remove the incorrect matches and refine outliers, the RANSAC algorithm was used to find inliers by estimating the homography matrix and to ignore outliers [10].

In AURORA we developed the ARTOD approach further into ARTOD+, in which we use optical flow to calculate the homography matrix which is significantly quicker than filtering matching keypoints. The proposed approach is best suited for small camera motions due to the underlying background subtraction method which relies upon previous images in order to update the background model. The pyramidal Lucas-Kanade optical flow algorithm [6] is used to extract the optical flow displacement vectors between the previous image and the current image based on features extracted using the Good Features to Track (GFT) [23] algorithm. The proposed method is a much more computationally efficient alternative to SIFT whilst maintaining the accuracy of keypoints between frames. The GFT algorithm was proposed to be used in ARTOD+ as an optimal solution to extracting keypoints in an image that can be used for the purpose of tracking. GFT uses pure translation to maintain the accuracy and reliability of the keypoints to be tracked. Not all keypoints in an image may contain complete motion information due to horizontal or vertical edges. This is known as the aperture problem. To overcome, this, GFT selects good features to track that take into account the texturedness and dissimilarity of the extracted keypoints to ensure that they correspond to real world features. GFT was selected due to the low computational complexity whilst providing enough accuracy to accurately determine the homography matrix.

Once the keypoints have been extracted from the previous image frame they are passed to the pyramidal Lucas Kanade optical flow algorithm [6] in order to find the positions of the features in the current image frame. The purpose of the algorithm is to track a point u on a frame, I_k where k is the frame number, to another location v on another frame I_{k+1}. The difference between points u and v represent the optical flow vector for that point and describes both the angle of motion and magnitude that translates u from frame I_k onto v from frame I_{k+1}. This is applied to all keypoints extracted from the previous image frame in order to find their corresponding positions in the current image frame.

Once the optical flow displacement vectors have been calculated, they are normalised and clustered using the online ELM clustering algorithm [5]. The x and y co-ordinates of the optical flow vectors, which encompass both the angle of motion and displacement of the optical flow, are used as the features for clustering. The

Evolving Local Means method is a one-pass approach which recursively calculates the local mean and variance of clusters.

The cluster with the largest number of optical flow vectors associated with it represents the motion of the camera with respect to the background. The features contained in the biggest cluster, that represent the camera motion, are then used to calculate the homography matrix using the random consensus sampling algorithm to further improve the robustness and remove any outliers that may remain. The choice of ELM, over alternatives such as the mean shift algorithm, was based on its computational efficiency with both approaches having comparable accuracy. The small errors introduced by ELM are minimised using the random sampling consensus (RANSAC) algorithm when calculating the homography. The minimum size of the object is limited by the clustering process, we remove clusters of less than 10 pixels to remove noise. If a moving object becomes stationary for a prolonged period then the approach will stop detecting it because RDE is a motion based detector. Pure rotation is difficult for the proposed approach to handle due to the use of optical flow for the calculation of the homogrpahy matrix.

2.2 Computationally Efficient Background Subtraction using RDE and RTSDE

Once the motion of the camera is estimated and a new image frame is aligned to the coordinate system of the background model, the next step is to identify novelties in the scene. We used a recursive algorithm which is fast and computationally efficient [1,2] and has been proven to have a fast response to environmental changes without requiring any threshold or pre-setting of any parameters for static [3] or moving cameras [21].

$$D\left(s_k^{i,j}\right) = \frac{1}{1+\|s_k^{i,j}-\mu_k^{i,j}\|^2+S_k^{i,j}-\|\mu_k^{i,j}\|^2} \tag{1}$$

where D denotes density;$s = \left[R_k^{i,j}, G_k^{i,j}, B_k^{i,j}\right]^T$ and represents the colour of the pixel in RGB form, S denotes scalar product of the data samples, μ is the mean, k is the frame number, i is the x coordinate in the frame and j is the y coordinate.

Both, the mean, μ and the scalar product, S can be updated recursively as follows [1,2]:

$$\mu_k^{i,j} = \frac{k-1}{k}\mu_{k-1}^{i,j} + \frac{1}{k}s_k^{i,j}, \quad \mu_1^{i,j} = s_1^{i,j} \tag{2}$$

$$S_k^{i,j} = \frac{k-1}{k}S_{k-1}^{i,j} + \frac{1}{k}\left\|s_k^{i,j}\right\|^2, \quad S_1^{i,j} = \left\|s_1^{i,j}\right\|^2 \tag{3}$$

Recursive density estimation technique is non-parametric and it represents the distance from a data sample to ***all***, or as many as needed, previous samples in the feature (e.g. colour) space. Density defined in this way is not exactly the probability-based density function [2]. Having the value of the density updated for the current k^{th}

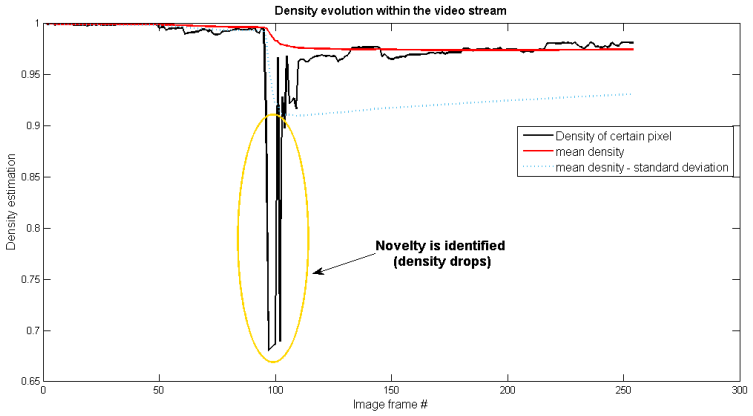

Fig. 4 The evolution of the density for a pixel throughout the video stream. The frames for which the value of the density drops below the value of $\bar{D}_k - (\sigma_k^D)^2$ are highlighted by a yellow circle. This indicates the presence of a foreground pixel.

frame for each $(i,j)^{th}$ pixel we can also calculate and update the mean density, $\bar{D}_k$ [1, 2]:

$$\bar{D}_k^{i,j} = \frac{k-1}{k}\bar{D}_{k-1}^{i,j} + \frac{1}{k}D_k^{i,j}, \quad \bar{D}_1^{i,j} = D_1^{i,j} \tag{4}$$

One can also recursively update the variance of the density as follows:

$$(\sigma_k^D)^2 = \frac{k-1}{k}(\sigma_{k-1}^D)^2 + \frac{1}{k}(D_k^{i,j} - \bar{D}_k^{i,j}), \quad (\sigma_1^D)^2 = 1 \tag{5}$$

Using equations (1-5) novelties (foreground) in the scene can be detected when the density of a pixel drops below the value of $\bar{D}_k^{i,j} - (\sigma_k^D)^2$ (Fig 4).

In order to deploy autonomous object detection algorithms on light-weight and low-power hardware, a new approach called RTSDE has been recently introduced [4]. This is aimed at further reducing the computational complexity of RDE by using integer only arithmetic with no division or floating point numbers calculations. As opposed to RDE, which uses a Cauchy-type kernel to calculate the inverse mean distance between a current observation and all past observation, RTSDE skips the kernel entirely. Instead RTSDE calculates a Total Sum Distance (TSD) or accumulated proximity, π for each colour dimension, which can be expressed recursively per feature, d as [4]:

$$\pi_{kd}^{i,j} = \left(s_{kd}^{i,j} - \mu_{kd}^{i,j}\right)^2 + S_{kd}^{i,j} - \left\|\mu_{kd}^{i,j}\right\|^2 \tag{6}$$

where π is the TSD or Total Sum Distance for the pixel at position (i,j) in the current k^{th} frame. s_k is the feature (e.g. colour) vector for the pixel, $d = \{1;2;3\}$ (e.g. colour, RGB) vector for the pixel. In the RTSDE approach [4] we multiply all elements in Eq. 2 by k, but assume it is written in terms of the mean of π which is $\bar{\pi}$ rather than in terms of the mean of s (Eq. 7).

$$k\bar{\pi}_{kd}^{i,j} = (k-1)\,\bar{\pi}_{(k-1)d}^{i,j} + \pi_{kd}^{i,j}, \quad \bar{\pi}_{1d}^{i,j} = \pi_{1d}^{i,j} \tag{7}$$

Subtracting this from (Eq. 6) indicates the deviation of the current π from the mean π over all previous image frames. The pixel can then be determined to be a novelty if the following is true for any feature dimension, d ($d = [R,G,B]$):

$$k\pi_{kd}^{i,j} - k\bar{\pi}_{kd}^{i,j} > k^2\xi \tag{8}$$

where $\bar{\pi}$ is the mean π value for feature d of the pixels at position (i,j) in all previous frames, and ξ is a predefined sensitivity factor, which can be proportional to the variance, σ. From this, we have two values that must be persistent across frames; s_l and $\bar{\pi}$.

Thus, we are able to recursively update the persistent values using just a single integer addition. This is the primary improvement over the original RDE algorithm, which requires floating point divisions to update μ and S, as in Eq. 2 and Eq. 3, in addition to a very expensive floating point square root calculation when recursively updating the standard deviation of the density, σ_D (Eq. 5).

In the state-of-the-art methods [27] the homography matrix is used to warp the current frame into the previous one to compensate for the moving camera (Figure 5). However, in the proposed ARTOD+ approach, the RDE data structures are warped into the new frames perspective which allows the RDE algorithm to be continuously updated, see Figure 6. When the data structures generated by RDE are warped some of the pixels will not have been processed before because of the camera movement. These pixels are initialised whilst the remaining pixels (those that have been processed before) are updated. The ability to continuously update pixels that have been seen by the RDE algorithm reduces the effect of the errors introduced by the warping process because the data from the previous frames are held in the data structures derived by RDE and, therefore, small changes are not detected as novelties.

2.3 ELM clustering of novelties in the scene

Once the RDE algorithm has detected the novelty pixels possibly corresponding to moving objects they must be clustered in order to detect individual objects as opposed to pixels. The disadvantage of background subtraction techniques from the point of view of object detection is that the number of novelty pixels detected can be an order of magnitude larger than other techniques, such as keypoints matching, due to the fact a feature is made up of a number of pixels. The increased number of pixels that must be clustered for the purpose of object detection can introduce a significant overhead because the clustering algorithms must assign each pixel to a cluster. In addition to this, there is a chance that pixels belonging to the same object, or those that are adjacent to other novelty pixels, are assigned to different clusters.

To overcome this limitation a simple technique is used for grouping novelty pixels utilising 8-connected pixels. This assumes that a novelty pixel adjacent to another novelty pixel forms part of the same object which may not always be the case in practical applications due to object occlusion, however it provides a quick and efficient

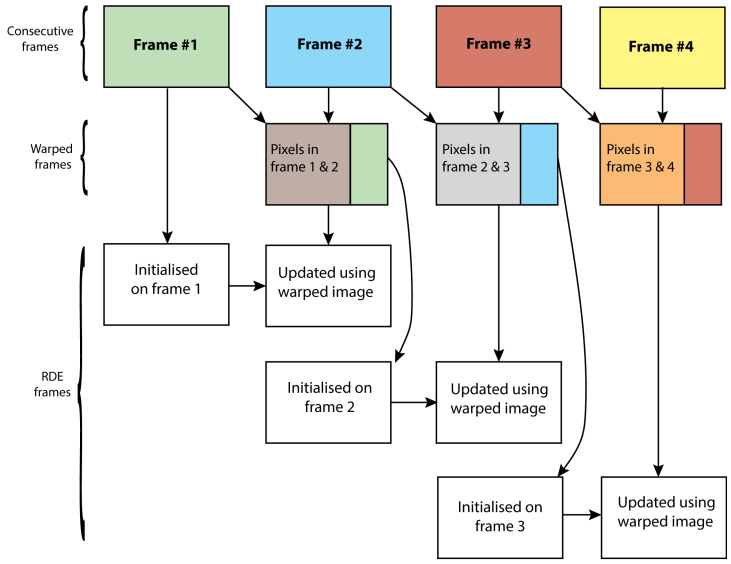

Fig. 5 State-of-the-art methods to compensate a moving camera and create a mosaic image

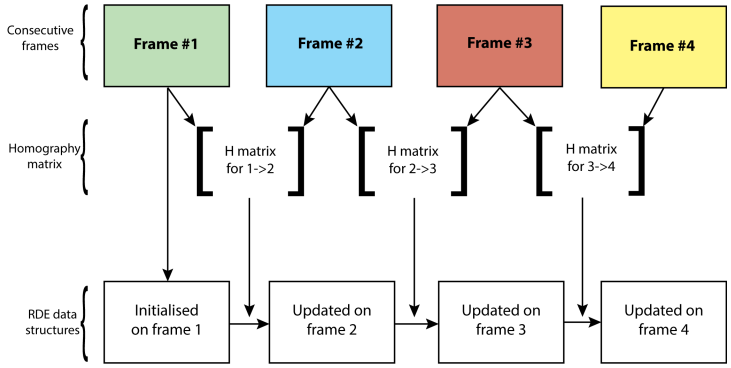

Fig. 6 Schematic representation of ARTOD+ used in AURORA

method to reduce the number of novelty pixels that must be clustered and significantly speeding up the clustering process. Once all 8-connected novelty pixels have been grouped a subset is then passed to the clustering algorithm which determines whether groups of pixels should be clustered together. For example, if two parts of an object are detected but are not grouped together then the clustering will identify that they belong to the same object if provided with a sufficient threshold. However, there may be some instances where grouped clusters belonging to the same group are assigned to different clusters. An example of this is shown in Figure 7. In this case, clusters are merged based on the groups to ensure that all pixels belonging to the same group belong to the same cluster [7].

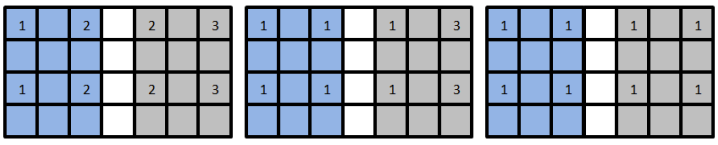

Fig. 7 Example of conflict between pixel groups and clusters. The coloured regions represent different novelty groups and the numbers represent different clusters. In (a) the pixels are clustered into three clusters but there are only two groups, (b) the clusters in the blue group are merged which, in turn, changes the identity of the cluster in the grey group, and (c) the clusters in the grey group are merged together

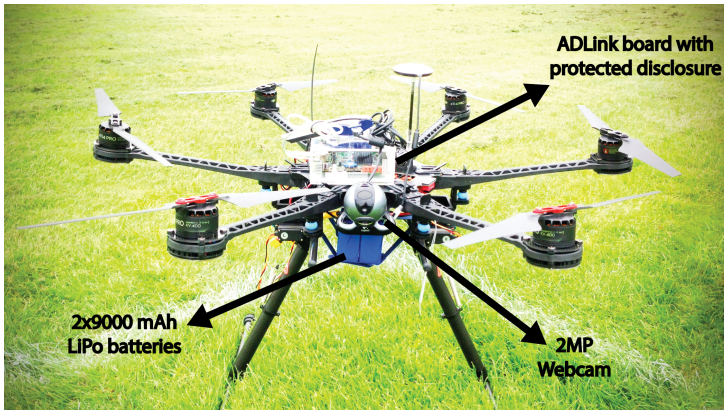

Fig. 8 The Hexacopter S800-EVO owned and operated by the intelligent system lab at the Lancaster University

3 Experimental Results

3.1 Implementation and Payload

In this section, we present real-life experiments for moving object detection to test the proposed algorithm in real-time using a light-weight and low-power ADLINK 850 single-board computer mounted on a DJI hexacopter S800 EVO (Figure 8). The hexacopter DJI S800 Evo, owned and operated by the Data Science Group at the Intelligent System Lab at Lancaster University has a dimension of $1000mm(Length) \times 1180mm(Width) \times 500mm(Height)$ and has 5-7kg of payload carrying capability. The payload consists of an ADLink 850, a USB wireless adaptor, a 2MP Logitech webcam, and two 9000mAh LiPo batteries which provide 12-15 minutes flight time.

The experiment was carried out with two objectives in mind. The first objective was to validate that the developed system was robust enough to detect moving objects during the flight tests and transmit the correct data to the ground station; the second objective was to test that the proposed approach was computationally efficient and capable of detecting moving objects in real-time.

The proposed approach was developed on Linux using C++ and runs on the ADLINK equipped with an Intel Core 2 Duo running at 2.2 GHz with 1GB of RAM.

Fig. 9 Results of the pre-recorded busy traffic scene and high resolution camera

For the following experiments we used the approach of refreshing RDE (page 5) with a refresh value of 10. Three different environments and scenarios were used in order to analyse the performance of the introduced algorithm. The first experiment was tested on pre-recorded video footage from a busy traffic scene recorded with a high resolution camera mounted on a quadcopter flying at high altitude (9). The second and third experiments were real-life experiments carried out at Lancaster University (10) and Copehill down village (11) whereby the processing was performed on-board the UAV and results downlinked to a laptop on the ground when a moving object was detected. The real-time video streams were produced by an external webcam mounted to the UAV with a resolution of 640 x 480 and without any mechanical stability, such as a gyroscopic gimbal or vibration damping. The results and videos obtained during real-life experiments are available on-line: `http://www.lancaster.ac.uk/staff/angelov/AURORA.htm`.

3.2 Results

3.2.1 Evaluation in terms of processing speed

Prior to the hexacopter taking off, the ADLink 850 is connected to a remote server and starts to wirelessly send image frames by executing the steps described in Figure 2. Once a moving object is detected, the frame of interest and/or binary frame is transmitted to the laptop on the ground and appears on the screen (Figure 10). When there is no movement the ADLink continues to process frames as normal, however it does not transmit any data back to the laptop server on the ground. When there are no moving objects in the field of view, a blank screen is displayed on the laptop server with text informing the user that no moving objects have been detected. The

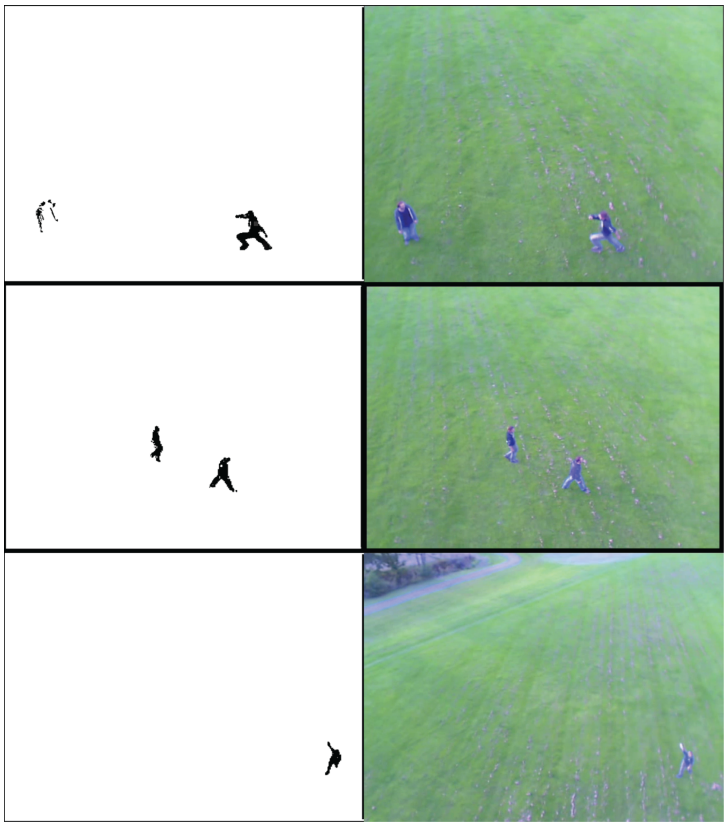

Fig. 10 Results of the real-life experiment (including shaking UAV, wind etc. processed on board UAV) performed at Lancaster University rugby pitch: Screenshot (left) with binary information which appears on the server screen on the ground only when a moving object is detected by the UAV; on the right-the original image (also transmitted, but only those frames with a moving object). The whole video is saved locally on-board for comparison purposes.

maximum distance from the drone to moving objects is dependent on the resolution of the camera and the size of the object relative to the camera. We were enabled to the detect moving objects up to 30 meters above ground from the 2MB on-board webcam. During the experiment the mounted camera was shaking noticeably and the illumination changed, however it did not affect the overall performance of the developed algorithm. Since we do not use any threshold to detect moving objects, the proposed approach quickly adapted to the changing environment and was able to accurately detect the moving objects in the scene.

Over the last few decades, background modelling has been studied significantly and many methods have been proposed. Comparing our proposed method with all the existing techniques is not feasible. In this paper, we compared the computational time of the ARTOD+ with three well-known background modelling algorithms including codebook method (CB) [15], Gaussian Mixture Model (GMM) [24] and SOBS [16] on 640×480 resolution for static and moving camera. The computational time for

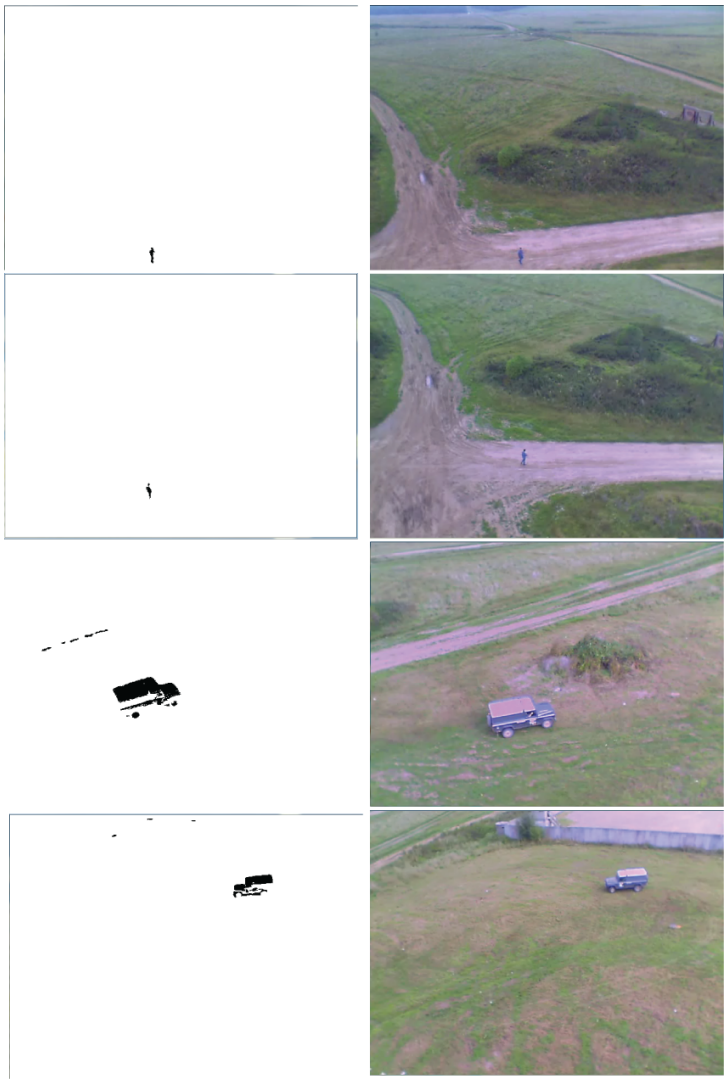

Fig. 11 Results of the real-life experiment (including shaking UAV, wind etc. processed on board UAV) performed at Copehill down village: Screenshot (left) with binary information which appears on the server screen on the ground only when a moving object is detected by the UAV; on the right-the original image (also transmitted, but only those frames with a moving object). The whole video is saved locally on-board for comparison purposes.

three background subtraction algorithms was estimated based on the results published in [9,18]. As shown in the table below, the ARTOD+ method outperformed the other three selected methods and achieved real-time performance of over 5 frames per second (Table 1). There is additional scope for significantly increasing the speed of the ARTOD+ approach by using a GPU implementation and CUDA technology because the algorithms used are highly suited for hardware parallelisation.

Table 1 A performance comparison between ARTOD+ and three well-known background modelling algorithms for a resolution of 640 x 480

Approach	Motion	Average time (ms)
ARTOD+	Static	**2**
	Moving	**191**
CB	Static	59
	Moving	4663
GMM	Static	12
	Moving	965
SOBS	Static	500
	Moving	39792

3.2.2 Evaluation in terms of power transmission efficiency

The proposed approach is also power efficient in terms of transmitting data. For example, a 12 minute 41 second original video with resolution 640×480 required 34.1MB memory to transmit the data with a bandwidth of 64Kbs and will keep it engaged all the time. On the other hand, the binary data for the same resolution and video length is only 7.17MB with a bandwidth of 16Kbs. For a comparison, the binary frames when objects were detected required only 0.37KB of memory which contains all the important information (Table 2). It should be noted that this comparison is, of course, application specific and is dependent on the number of objects detected.

Table 2 Data transmission comparison between the original and binary video file

	Video length	Video resolution	Size (MB)	Size of transmitted data per frame (KB)
Original video	12'41"	640x480	34.1	1.95
Binary video	12'41"	640x480	7.17	0.37
Transmitted binary video	2'54"	640x480	1.43	0.37

4 Conclusion

In this paper, we described a novel low-cost and power-efficient system, AURORA, based on the improved original ARTOD algorithm, ARTOD+, for the use in unmanned aerial vehicles for on-board autonomous lower level video analytics tasks.

The described real-time autonomous object detection approach utilises a homography matrix derived from using optical flow, recursive density estimation and evolving clustering algorithms. The approach does not require prior knowledge of the objects to be detected and can dramatically reduce the information being transmitted to the ground in real time resulting in reduced risk of interception, reduced power and energy for transmitting the data and reduced bandwidth. The whole video can also be saved on board in memory and analysed in more detail by human operators post flight. The data transmitted also allows for human operators to focus on areas of importance, as opposed to having to actively search for areas of interest, which reduces the risk of human error and fatigue.

Acknowledgements This work was funded under the MODs Centre for Defence Enterprise themed call for Generic Enablers for Low Size, Weight, Power and Cost (SWAPC), contact reference DSTLX1000082760. RTSDE was developed in EntelSenSys Ltd as a subcontractor in this work. The authors would like to thank Dr. Asmar Khan (C code and first experiments) and Mr. Ashley Wilding (Beagle board implementation) for their contribution to AURORA project.

References

1. Angelov, P.: Anomalous system state identification. patent application, GB1208542.9 (2012)
2. Angelov, P.: Autonomous Learning Systems: From Data Streams to Knowledge in Real Time. John Wiley and Sons (2012)
3. Angelov, P., Sadeghi-Tehran, P., Ramezani, R.: A Real-time Approach to Autonomous Novelty Detection and Object Tracking in Video Stream. International Journal of Intelligent Systems **26**, 189–205 (2011)
4. Angelov, P., Wilding, A.: RTSDE: Recursive Total-Sum-Distances-based Density Estimation Approach and its Application forAutonomous Real-Time Video Analytics. Symposium Series on Computational Intelligence (2015)
5. Baruah, R.D., Angelov, P.: Evolving Local Means Method for Clustering of Streaming Data. IEEE World Congress on Computational Intelligence pp. 2161–2168 (2012)
6. Bouguet, J.Y.: Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm. Intel Corporation, Microprocessor Research Labs (1999)
7. Clarke, C.: Smartphone Application for Real-time Object Detection. Master's thesis, School of Computing and Communications, Lancaster University (2014)
8. Farin, D., de With, P.H.N., Effelsberg, W.A.: Video-object segmentation using multi-sprite background subtraction. IEEE International Conference on Multimedia and Expo, ICME '04. **1**, 343–346 (2004)
9. Fauske, E., Eliassen, L.M., Bakken, R.H.: A Comparison of Learning Based Background Subtraction Techniques Implemented in CUDA . In: NAIS, pp. 181–192 (2009)
10. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM **24**(11), 381–395 (1981)
11. Gao, T., Liu, Z.G., Yue, S.H., Mei, J.Q.: Traffic Video-based moving vehicle detection and tracking in the complex environment pp. 569–588 (2009)
12. Gelgon, M., Bouthemy, P.A.: A region-level motion based graph representation and labeling for tracking a spatial image partition. Pattern Recognition **33**, 725–740 (2000)
13. Hayman, E., Eklundh, J.O.: Statistical background subtraction for a mobile observer. In: IEEE International Conference on Computer Vision, pp. 67–74. IEEE (2003)
14. Huwer, S., Niemann, H.: Adaptive change detection for real-time surveillance applications. In: Third IEEE International Workshop on Visual Surveillance, pp. 37–46. IEEE (2000)
15. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground–background segmentation using codebook model. Real-Time Imaging **11**(3), 172–185 (2005)
16. Li, L., Huang, W., Tian, Q.: A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications. IEEE Transaction on Image Processing **13**(11), 1459–1472 (2004)

17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
18. Morris, G., Angelov, P.: Real-time Novelty Detection in Video using Background Subtraction Techniques: State of the art. In: IEEE International Conference on Systems, Man and Cybernetics (2014)
19. Murray, D., Basu, A.: Motion tracking with an active camera. IEEE Trans. on Pattern Analysis and Machine Intelligence **16**, 449–459 (1994)
20. Price, A., Pyke, J., Ashiri, D., Cornall, T.: Real time object detection for an unmanned aerial vehicle using an FPGA based vision system. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2854–2859 (2006)
21. Sadeghi-Tehran, P., Angelov, P.: ARTOD: Autonomous Real Time Objects Detection by a Moving Camera using Recursive Density Estimation. In: Novel Applications of Intelligent Systemsm Springer Verlag, vol. 586, pp. 123–138. Springer Verlag (2015)
22. Sadlier, D.A., O'Connor, N.E.: Evaluation of a vehicle tracking system for multi-modal UAV-captured video data. SPIE Defense, Security, and Sensing **7668** (2010)
23. Shi, J., Tomasi, C.: Good features to track. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '94., pp. 593–600. IEEE Comput. Soc. Press (1994)
24. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1999)
25. Sugaya, Y., Kanatani, K.: Extracting moving objects from a moving camera video sequence. Symposium on Sensing via Image Information pp. 279–284 (2004)
26. Thornton, S., Hoffelder, M., Morris, D.: Multi-sensor detection and tracking of humans for safe operations with unmanned ground vehicles (2008)
27. Tsinko, E.: Background Subtraction with a Pan/Tilt Camera. Ph.D. thesis, The University Of British Columbia (2010)
28. Watanabe, Y., Fabiani, P., Le Besnerais, G.: Simultaneous visual target tracking and navigation in a GPS-denied environment. In: International Conference on Advanced Robotics, ICAR 2009., pp. 1–6 (2009)
29. Zhang, G., Jia, J., Xiong, W., Wong, T.T., Heng, P.A., Bao, H.: Moving Object Extraction with a Hand-held Camera. IEEE 11th International Conference on Computer Vision, ICCV 2007. pp. 1–8 (2007)