Data Fusion for Unsupervised Video Object Detection, Tracking and Geo-Positioning

Denis Kolev, Garik Markarian
R&D Department, Rinicom Ltd.
Lancaster, Lancashire, UK
{denis_kolev, garik}@rinicom.com

Dmitry Kangin
Data Science Group,
School of Computing and Communications,
Lancaster University, UK
d.kangin@lancaster.ac.uk

Abstract - In this work we describe a system and propose a novel algorithm for moving object detection and tracking based on video feed. Apart of many well-known algorithms, it performs detection in unsupervised style, using velocity criteria for the objects detection. The algorithm utilises data from a single camera and Inertial Measurement Unit (IMU) sensors and performs fusion of video and sensory data captured from the UAV. The algorithm includes object tracking and detection, augmented by object geographical co-ordinates estimation. The algorithm can be generalised for any particular video sensor and is not restricted to any specific applications. For object tracking, Bayesian filter scheme combined with approximate inference is utilised. Object localisation in real-world co-ordinates is based on the tracking results and IMU sensor measurements.

Keywords: Bayesian filters, UAV, object tracking, unsupervised detection, rigid motion segmentation

1 Introduction

Nowadays, there are a plenty of algorithms aimed on object detection and tracking. These algorithms work using data captured in different wavelengths (synthetic aperture radars (SAR), video cameras, thermal imagers). Video camera can serve for a cheap solution for the tracking system. Thermal imager has an advantage that it allows to see discernible object contours either in day- or in night-time conditions.

Our aim is to resolve the problem of automatic moving objects detection on the video feed with moving camera. To do this, we consider unsupervised approach to the object detection, which is independent of the size and form of the object.

Also the positions of the objects are estimated and tracked. It means estimation of the geographical co-ordinates of the object in each moment of time.

Suggested technical implementation is like shown in figure 1. Our system works on board the UAV, where the sensors and camera are mounted.

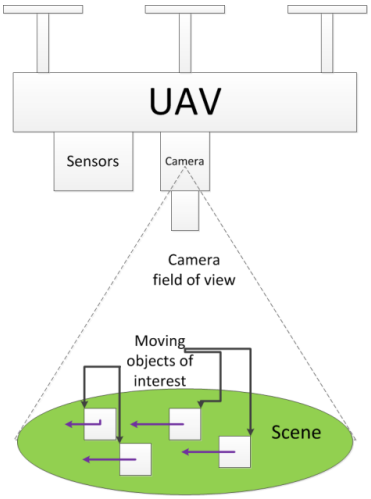

Figure 1. The scheme of the proposed system

2 Literature review

The object detection and tracking problems are widely studied, nevertheless the problem statements vary tremendously, as well as application domains the methods are designed for. We have not found the methods doing exactly the same as the whole proposed system with object tracking, but we have observed the similar tracking methods.

Some of the object detection methods rely on supervised detection techniques [1], [2], which require learning set preparation and taking into account all the varieties of the objects' appearances. More, the range of object appearances can be too various to consider it beforehand.

Another methods abandon the object detection problem at all, relying on a 'human in the loop', i.e. an operator, which points to the object that should be detected.

The method proposed here is based on unsupervised detection with some restrictions imposed on the object appearance, expressed in terms of object speed estimation. The unsupervised methods are proposed in [3] and [4], but they are defined for the dissimilar domains. One of the most widespread particular cases of unsupervised detection relies on static background, that can be resolved by background subtraction techniques [5], [6], but for this problem we think it is inapplicable because we cannot assume the camera to be static.

The tracking methods are well developed, and in contrast to detection methods, some of them perform as domain independent frameworks. Different levels of complexity can be emphasised for these trackers. Optical flow based trackers, like Lucas-Kanade tracker [7], are simple trackers for short term surveillance for the point objects. More complex trackers aggregate information on the points' movement from an optical flow tracker for different points, and apply it to track the sets of points, constituting objects. For example, it can be done by means of rigid motion segmentation [8], where we additionally assume the objects to move consistently. Actually, the method we propose can be referred to this group. However, there are even more complex methods like TLD [9], which are capable of challenging the object partial occlusion, as well as re-appearance after the full occlusion. But it does not support simultaneous tracking and detection, as well as multiple object tracking. Here we believe that the problem of full of partial occlusion in case of 'top-down' UAV view is not so critical, so we do not utilise complex trackers in order to enhance computational efficiency. At last, the objects we propose as a response of our tracker can be treated by such a complex tracker.

Multiple Hypothesis Tracking (MHT) [10], when used for cluttered data, and different variations of Probability Hypothesis Density (PHD) [11] filters perform the idea of simultaneous multiple object detection and tracking based on Bayesian filtering, consonant to the proposed algorithm. One of the examples of the PHD filter application to multi-target tracking is given in [12]. Another way is to exploit custom update equations instead of Bayesian filtering ones like it is done in [13]. The conceptual difference with the proposed algorithm is that all these methods recognise the possibility of occurrence of clutter, whilst we make motion segmentation for all the points and then investigate what of them are actually objects.

One more task is to combine the object appearance on the image with the object position in the real world. For this purpose, telemetry approach should be utilised. The approach resembles SLAM [14] problem statement, but it is not the same. We cannot assume, that the camera has sufficiently large parallax to estimate the distance to the object by feature points matching. Hence, and also to perform tethering to the world co-ordinates, we rely on the external Euler angle sensors to estimate the camera position, rather than on the image feature points, as it is done in many SLAM problem statements like [14]. More, in some cases the background is not sufficiently gradient (e.g. sea, desert) to be reliably matched by feature point extraction algorithm, but the edges appear to be highly discernible for most of the objects.

3 Problem statement

The video tracking problem can be stated as follows. Suppose, that we have a video feed, which can be represented as a (finite, i.e. pre-recorded file, or infinite, i.e. stream) sequence $\{I_1, I_2, \dots I_k \dots\}$, where k is the discrete instant of time, or frame number. Each of the grey-scale frames is a function $I_k: X \to [0,1]$, where X is a frame domain, usually $X = [0,h] \times [0,w] \subset \mathbb{R}^2$ or $X = [1,h] \times [1,w] \subset \mathbb{N}^2$, and w and h are frame's width and height in pixels, respectively. For each of the frames, we intend to determine presence of the objects within it, and consistently follow them through the tracks.

To resolve the problems of the object tracking and detection, we exploit 'rigid motion segmentation' approach [8]. This approach implies that all points of the object presented are moving consistently and synchronously from frame to frame. We track feature points, which are sampled by Harris corner detector with suppression, by Lucas-Kanade tracker [7]. Then we aggregate it into tracks, i.e. sequences of the tracked feature points, which have a size up to some pre-defined maximum (for example, track of the last 10 tracked positions of the point). Then the estimated tracks are clustered in time-consistent manner using Bayesian filtering.

Another problem we tackle is real world object localisation by means of telemetry. Suppose, that we have the frame I_k, tuple $\{\alpha_k, \beta_k, \gamma_k, x_k, y_k, z_k, P_c\}$ containing three Euler angles $\alpha_k, \beta_k, \gamma_k$ of camera rotation in the world co-ordinate system, camera position co-ordinates x_k, y_k, z_k, which can change with time, and camera parameters P_c (focal distance and angles of view), which are assumed to remain constant throughout the algorithm's working time. Based on this data, we can produce a mapping from the image to the world co-ordinates $(x_w, y_w, z_w) \in \mathbb{R}^3$, as well as estimate the geographical co-ordinates $(\lambda, \mu) \in \mathbb{R}^2$ assuming a plain terrain.

Then, we go ahead to the method formulation, which is depicted in figure 2.

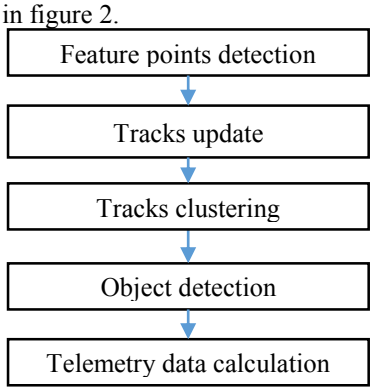

Figure 2 General description of the method.

The method is performed for each frame and starts from the feature points detection. After that, the points are attached to the tracks by the procedure which is outlined in the following sections. At the following step, the tracks are clustered using Bayesian filter approximation approach, ensuring between-frame consistence of the cluster labels. Then, the objects' clusters are selected from the set of clusters which move discernibly relative to the background. Finally, the estimated geographic positions for the detected objects are obtained using telemetry data processing algorithms.

4 Tracking and detection

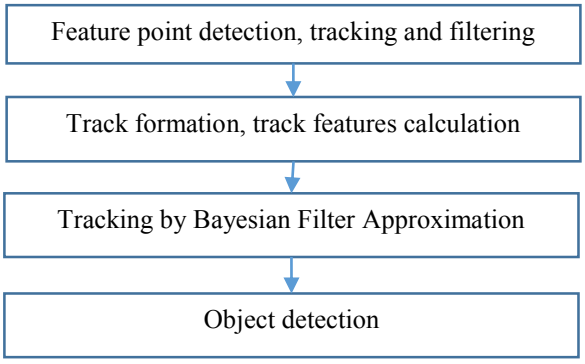

Figure 3. Tracking and detection method

As it is shown in figure 3, at the first stage, the feature points are detected, tracked from frame to frame, and replaced, if there is no reliable correspondence on the subsequent frame.

Then, for each of the frames, the tracks are built up the following way: the points from the new frame are attached to the tracks with the matching terminal points. If there is no match for any point from the previous frame, because it cannot be tracked or the object has left the capturing area, the track is terminated and abolished.

The key part of our approach is a novel tracking and detection algorithm based on the Bayesian filter approximation. As it was mentioned before, it is based on time-consistent update of the frame clusters. Time consistent update means that the form, the parameters and the labels of the clusters are dependent from the results on the previous frame. The global idea is to track the "clusters" of stable points in the frame according to their movement by the application of Bayesian filter approximation approach inspired by Kalman filter [21].

Then, the objects are detected according to the assumption that the camera can have some rotation or parallax from frame to frame, and we can think of the objects as moving if the expected movement of the points does not correspond to the expected movement of the camera. It allows the algorithm to detect the object movement when the camera has background movement itself.

4.1 Feature point detection, tracking and filtering

Feature point detection and tracking are deeply interdependent procedures as the quality of the tracking depends on the selected points. In this work we propose a well-known combination of Forward-Backward Lukas-Kanade (FBLK) [15] tracking of the points selected by modified Harris corner detector (HCD) [16]. HCD is well suitable [17] point detector for FBLK-based tracking because it provides points according to conditioning number of the Hessian matrix [7], which is inverted during optical flow calculation. We utilise a modified HCD for grey-scale image in order to ensure sparsity of the detected points over the frame. This property is achieved by non-maximum suppression and application of sub-frame detectors. Hereafter we consider the "transition" from the frame I_k to I_{k+1}. Here we define the feature points set from the frame I_k by $F_k = \{f_1^k, \dots, f_{n_k}^k\}, f \in \mathbb{R}^2$ – pixel coordinates.

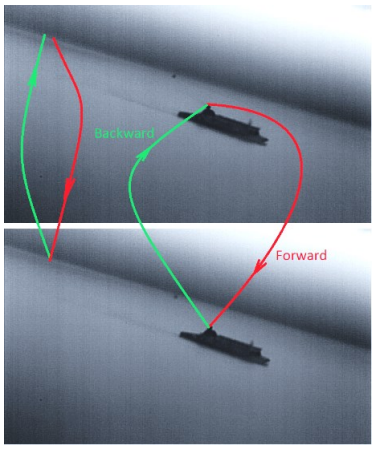

Figure 4. The illustration of forward-backward error.

At each new frame feature points from the previous frame are tracked using FBLK. The prediction of the point f from I_k to I_{k+1} is defined as $p_k^{k+1}(f)$. FBLK provides a measure of the point tracking quality – backward error, which is calculated as

$$FB_k^{k+1}(f) = \left\| p_{k+1}^k\left(p_k^{k+1}(f)\right) - f \right\|. \quad (1)$$

In order to ensure high precision of the algorithm only feature points with backward error lower than a pre-defined threshold T_1. The illustration of FBLK error is presented in the figure 4. On the right side, one can see well tracked point, which gives a reasonable forward-backward error. On the left side, the point is barely discernible, that causes the forward-backward error to be large due to erroneous matching. Let us define the set of tracked points (from the frame I_k) with low error as $G_k \subset F_k, G_k = \{f \in F_k : FB_k^{k+1}(f) < T_1\}$.

If all the points are tracked well, some pre-defined percent of the points is deleted (with highest backward error). Then new feature points are detected using HCD algorithm. The number of newly detected points is equal to the number of deleted feature points. The set of newly detected points is defined as H_k. Thus,

$$F_{k+1} = p_k^{k+1}(G_k) \cup H_{k+1}. \quad (2)$$

4.2 Track formation

Tracks are defined as a sequence of feature points from sequential frames: $t_i^k = \{f_{i_{s_i}}^{s_i}, \dots, f_{i_k}^k\}$ is a track defined for a point from frame s_i, on which the track was initiated, to frame k. Informally, the track defines a "trajectory" of the filmed points from frame to frame. There are two possible options of for each point from the new frame when updating the tracks, depending on whether or not $f_{i_k}^k \in G_k$, i.e. whether the last point of the track was reliably predicted by FBLK.

If the point is reliably detected, then the corresponding point is added to the track (figure 5, track 1):

$$t_i^{k+1} = t_i^k \cup p\left(f_{i_k}^k\right), \text{if } f_{i_k}^k \in G_k. \quad (3)$$

Otherwise, if $f_{i_k}^k \notin G_k$, a track is terminated (figure 5, track 2). For all other feature points from the frame $(k+1)$, the new tracks are created (figure 5, tracks 5 and 6). One particular case is the tracks merging, when several tracks come to the same point (figure 5, tracks 3 and 4). These tracks are to be merged.

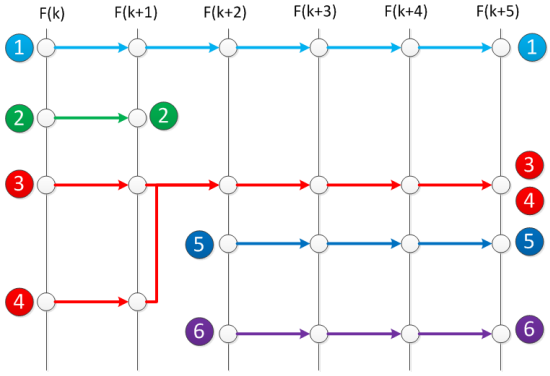

Figure 5. The illustration of the tracks formation.

Then each of the existing tracks are mapped into some "feature space", which characterise the speed and position of the point on the frame for the last few seconds. In this work we use the mean position of the track points for the last a (typically, 10) frames in order to characterise the spatial properties of the track. The information about the tracks which has happened later than $a-1$ tracks ago is to be discarded. A difference between the last point and the point from $a-1$ frames ago characterises the speed. Hence the feature vector is composed of two spatial and two velocity components. The lag of a frames is used in order to ensure the stability of the speed property as a difference between sequential points may be noised. Feature vector related to the track t_i^k is denoted as $d(t_i^k)$ or simply d_i^k. It should be outlined that only those tracks are used which have the length of the points sequence ("age") equal to a: $\left|t_i^k\right| = a$. This limitation is introduced to analyse only such points which are reliably tracked for a number of frames. The 'equality' sign stands here because the points that exist more than a frames are discarded from the track. A track which satisfies the "age" limitation is referred as 'mature'.

4.3 Bayesian filter approximation

As mentioned above, the Bayesian filter model is used in order to model time-consistent development of the mixture of Gaussians (MoG), which is a core part of rigid motion segmentation approach described here. Features of the tracks computed and observed for each frame are considered as a sample generated from the MoG, where each Gaussian represents a separate rigidly moving object, or part of the scene. In order to introduce "smooth" development of the mixture, frame-to-frame changes of the parameters of the MoG — means, covariance matrices and prior probabilities (weights) — are modelled by a dynamic system. Hence, the parameters of the MoG are considered as hidden variables of the Bayesian filter, and the features of the tracks are referred to observed variables. The key difference from the standard approach (and similarity to the PHD filter [11]) is the fact that the model utilises a sample of observed variables at each time step.

Hereafter we define the variables and probability models for the Bayesian filter. Consider first the hidden variables that define the parameters of the MoG at each frame. We denote the mean of the j-th mixture component for frame I_k by $m_j^k \in \mathbb{R}^l$, where l is the features' dimensionality, and corresponding covariance matrix by $\Sigma_j^k \in \mathbb{R}^{l\times l}$ and weights by $w_j^k \in \mathbb{R}$, where $\sum_{i=1}^K w_i^k = 1\ \forall k.\ m^k, w^k$ and Σ^k denote the union of these corresponding parameters.

As pointed before, computed set of the tracks' features is generated from the MoG defined by the hidden variables. Denote the feature vectors of all mature tracks from the frame I_k by $D_k = \{d_1^k, \dots, d_{n_k}^k\}, d_i^k \in \mathbb{R}^l$, which represents observed variables for k-th stage where n_k is the number of the mature tracks in the k-th frame. One of the model assumptions is that the set D_k is i.i.d. and generated by MoG, so

$$d_i^k \sim p\left(d_i^k | m^k, \Sigma^k, w^k\right) = \sum_{j=1}^K w_j^k \mathcal{N}\left(d_i^k | m_j^k, \Sigma_j^k\right), \quad (4)$$

where we denote the normal distribution

$$\mathcal{N}(d|m,\Sigma) = \frac{1}{\left(\sqrt{2\pi}\right)^l |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{(d-m)^T \Sigma^{-1}(d-m)}{2}\right) \quad (5)$$

Therefore, probability distribution of the observed variables given the hidden (likelihood) is defined as

$$p(D_k | m^k, \Sigma^k, w^k) = \prod_{i=1}^{n_k} p\left(d_i^k | m^k, \Sigma^k\right), \quad (6)$$

In order to achieve a "smooth" development in time of each model following update rules for the mean vectors are utilised:

$$\begin{cases} m_j^{k+1} = m_j^k + v_j + \varepsilon_j, \\ \varepsilon_j \sim \mathcal{N}\left(\varepsilon | 0, \Gamma_j^k\right), \end{cases} \quad (7)$$

for $j = 1, \dots, K$. Term v_j denotes the average movement of the points of the tracks from cluster j from the frame I_k to I_{k+1}, and ε_j denotes a random Gaussian noise, Γ_j^k is a covariance matrix of the noise.

The time-propagation model of the covariance matrices Σ^k and prior weights w^k is performed using the heuristic approach, which is described further. All parameters of the MoG are assumed to be independent [19]. Probability distribution for the covariance is imposed over the "precision matrices" $\Lambda_j^k = \left(\Sigma_j^k\right)^{-1}, j = 1, \dots K$ in the form of Wishart distribution, which is conjugate to the Gaussian distribution. Wishart distribution is defined as:

$$\mathcal{W}(\Lambda|\nu,\Psi) = \frac{|\Lambda|^{\nu-l-1} \exp(-\mathrm{tr}(\Psi^{-1},\Lambda)/2)}{2^{\frac{\nu l}{2}} |\Psi|^{\frac{\nu}{2}} \Gamma_p\left(\frac{\nu}{2}\right)}, \quad (8)$$

$$\Lambda \in \mathbb{R}^{l\times l} \Lambda \geq 0, \Psi \geq 0, \nu > (p-1).$$

Here $\Gamma_p(\cdot)$ is the multivariate gamma function. For the parameters w^k Dirichlet distribution is used in this work. Dirichlet distribution is denoted as

$$\mathrm{Dir}(w|\alpha);\ w,\alpha\ \in \mathbb{R}^l;\ \langle w,\mathbf{1}\rangle = 1;\ w,\alpha\ \geq 0. \qquad (9)$$

We exploit the forward filtering approach for probabilistic inference in Bayesian filter where at each moment of time k probability distributions of the hidden variables given *all previous* observed variables $D_1,\dots,D_k$ are estimated. Thus, the model estimates following distributions (due to parameter independence):

$$p\left(m_j^k\middle|D_1,\dots,D_k\right);\ p(\Lambda_j^k|D_1,\dots,D_k);\ p\left(w_j^k\middle|D_1,\dots,D_k\right). \qquad (10)$$

Usually forward filtering is decomposed into two steps: prediction and update. We consider these two stages in more detail further.

4.3.1 Prediction

The prediction step aims to estimate the distributions of the hidden variables at stage k given the observed data $D_1,\dots,D_{k-1}$. It is done using the estimated distributions from the stage $k-1$ and "transition" pdf defined by the dynamic system (7). For the parameters m_j^k the solution is obtained analytically. If

$$p\left(m_j^{k-1}\middle|D_1,\dots,D_{k-1}\right) = \mathcal{N}\left(m_j^{k-1}\middle|\mu_j^{k-1},\Xi_j^{k-1}\right), \qquad (11)$$

then, using the statements in the dynamic system

$$p\left(m_j^k|D_1,\dots,D_{k-1}\right) = \mathcal{N}\left(m_j^k\middle|\mu_j^{k-1}+v_j^k,\Xi_j^{k-1}+\Gamma_j^k\right). \qquad (12)$$

Parameter v_j^k may be computed using the information about the changes of the track features from the previous frame. Let $IS_j^k = \{i_1^{j,k},\dots,i_{N_{jk}}^{j,k}\}$ be a set of indices of adult tracks from frame I_{k-1} clustered to the j-th mixture component, which are presented on frame I_k. Then v_j^k may be computed as

$$v_j^k = \frac{1}{N_{jk}}\sum_{i\in IS_j^k}\left[d\left(t_i^k\right) - d\left(t_i^{k-1}\right)\right]. \qquad (13)$$

Covariance matrix Γ_j^k is estimated as sample covariance for the difference of features of tracks with indices from IS_j^k.

Consider the prediction step for the covariance matrices and prior weights. For both of the parameters it is hard to select "transition" pdf $p\left(\Lambda_j^k\middle|\Lambda_j^{k-1}\right), p\left(w_j^k\middle|w_j^{k-1}\right)$, which after prediction step preserve the form and family of the distribution. For that reason in this work we select the prediction distribution for Λ_j^k, w_j^k in heuristic manner.

If $p(\Lambda_j^{k-1}|D_1,\dots,D_{k-1}) = \mathcal{W}\left(\Lambda_j^{k-1}\middle|v_j^{k-1},\Psi_j^{k-1}\right)$, predictive density is assigned as follows:

$$p\left(\Lambda_j^k\mid D_1,\dots,D_{k-1}\right) = \mathcal{W}\left(\Lambda_j^k\middle|\frac{v_j^{k-1}}{\rho_j^k},\rho_j^k\Psi_j^{k-1}\right), \rho_j^k > 1. \qquad (14)$$

This modification preserves the expected value of the inverse covariance of the cluster, but enlarges the variance allowing the adaptation of the posterior distribution for the new data in the update step in more flexible manner. Larger values of the parameter ρ_j^k cause more flexible and less stable update on the update step in comparison to the predictive density.

It was experimentally established that most convenient and efficient predictive distribution $p(w^k|D_1,\dots,D_{k-1})$ should be the same at each moment of time, i.e.

$$p(w^k|D_1,\dots,D_{k-1}) = \mathrm{Dir}(w^k|\eta\times\mathbf{1}), \qquad (15)$$

where η is a small constant, $\mathbf{1}\in\mathbb{R}^l$ – vector with all components equal to 1.

Thus, assuming that means and covariance matrices of different components are independent, overall predictive density is as follows:

$$p(w^k,\Lambda^k,m^k|D_1,\dots,D_{k-1}) = \prod_{j=1}^K\left[\mathcal{N}\left(m_j^k\middle|\mu_j^{k-1}+v_j^k,\Xi_j^{k-1}+\Gamma_j^k\right)\times \mathcal{W}\left(\Lambda_j^k\middle|\frac{v_j^{k-1}}{\rho_j^k},\rho_j^k\Psi_j^{k-1}\right)\right]\times \mathrm{Dir}(w^k|\eta\times\mathbf{1}). \qquad (16)$$

4.3.2 Update

The update step aims to compute distributions of the hidden variables at stage k given observed data $D_1,\dots,D_k$. Using Bayes rule we obtain:

$$p(w^k,m^k,\Lambda^k|D_1,\dots,D_k) \propto \propto p(w^k,m^k,\Lambda^k|D_1,\dots,D_{k-1})\times \times p(D_k|m^k,\Lambda^k,w^k) = \mathcal{L}(m^k,\Lambda^k,w^k) \qquad (17)$$

Here the likelihood of the observed variables D_k is a product of MoG, as in (6). Therefore, direct inference is computationally infeasible and the exact distribution becomes more complex from step to step. For this reason an approximation of the posterior distribution is used. The posterior is estimated in the form, that preserves the initial structure of the distribution (16), i.e.:

$$p(m^k,\Lambda^k,w^k|D_1,\dots,D_k) = \prod_{j=1}^K\left[\mathcal{N}\left(m_j^k\middle|\mu_j^k,\Xi_j^k\right)\times \mathcal{W}\left(\Lambda_j^k\middle|\nu_j^k,\Psi_j^k\right)\right]\times \mathrm{Dir}(w^k|\alpha^k) = p(m^k|D_1,\dots,D_k)p\left(\Lambda_j^k\middle|D_1,\dots,D_k\right)p(w^k|D_1,\dots,D_k)$$

This distribution structure requires parameters μ_j^k, ν_j^k, Ψ_j^k and α^k to be defined. For this purpose we use the Laplacian approximation for $p(m^k|D_1,\dots,D_k)$, and similar heuristic approach for distributions of Λ_j^k and w^k, utilising the mode of distribution (17), which gives:

$$\Psi_{j,}^k = \frac{\Lambda_{j,MAP}^k}{\nu_j^k},\quad \mu_j^k = m_{j,MAP}^k,\quad \alpha^k = w_{MAP}^k. \qquad (18)$$

Here $\Lambda_{j,MAP}^k$, $m_{j,MAP}^k$, w_{MAP}^k stay for mode of (17). These parameters can be estimated using Expectation-Maximisation algorithm [24]. Parameter ν_j^k is selected as a sum of posterior probabilities of j-th cluster over feature vector in D_k.

Parameter Ξ_j^k is updated according to Laplacian approximation approach for $p(m^k|D_1,\dots,D_k)$, i.e.:

$$\Xi_j^k = -\left[\nabla_{m_j^k}\nabla_{m_j^k}\log(\mathcal{L}(\mu^k,\Lambda^k,w^k))\right]^{-1} \qquad (19)$$

In other words, Ξ_j^k is assigned to the negative inverse Hessian of the logarithm of the posterior likelihood at the mode of the distribution.

The update step may be interpreted as Laplace-like approximation of the posterior distribution.

4.4 Object detection

Kabsch algorithm [25] is used for the object detection. This algorithm aims to establish the estimated rotation matrix and translation vector, using the following assumption:

$$\hat{G}_k^T = UG_{k-1}^T + P,$$
$$\sum_{1 \le i \le n_k} \left| G_{k\,i}^T - \hat{G}_{k\,i}^T \right|_{L^2}^2 \to \min_{U,P}, \quad (20)$$

where G_{k-1} are the tracked points from the previous frame, G_k are the matching points from the new frame and $\hat{G}_k$ is its estimation, U is a rotation matrix, which is supposed to be orthogonal, and P is the translation vector, n_k is the count of the segmented tracks. The matrices are represented in the following way:

$$\hat{G}_k = \begin{pmatrix} \hat{x}_1 & \hat{y}_1 & h \\ \hat{x}_2 & \hat{y}_2 & h \\ \dots & \dots & \dots \\ \hat{x}_{n_k} & \hat{y}_{n_k} & h \end{pmatrix}, G_k = \begin{pmatrix} x_1 & y_1 & h \\ x_2 & y_2 & h \\ \dots & \dots & \dots \\ x_{n_k} & y_{n_k} & h \end{pmatrix},$$

G_{k_i}, $\hat{G}_{k_i}$ are the i-th rows of the matrices G_k and $\hat{G}_k$ respectively.

$U \in \mathbb{R}^{3\times3}$ is an orthogonal matrix, $P = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}$,

where h is some pre-defined height constant, i.e. 1.

To make the estimation more accurate, it is repeated N times, typically $N = 3$, and at each iteration for the best-matched points from the previous iteration up to some quantile η ($\eta = 0.5 - 0.9$) are selected. The matching between the points is estimated according to L^2 metric as

$$E_k = \left[(\hat{G}_k - G_k) \otimes (\hat{G}_k - G_k) \right] \mathbb{I}_{3\times1}, \quad (21)$$

where $\mathbb{I}_{3\times1}$ is an all-ones matrix, $\otimes$ stays for an element-wise multiplication.

To distinguish between the object and the background points, the threshold is calculated dynamically using the following simple heuristics. The points are sorted by their L^2 errors, and the standard deviation S of difference between the neighbouring errors in the sorted array is calculated. The error threshold T is stated as an average between two smallest elements of the errors sequence, which difference from the previous error is more than τS, where τ is some parameter (typically $3 \le \tau \le 20$). The scheme, illustrating this method, is summarised in figure 6.

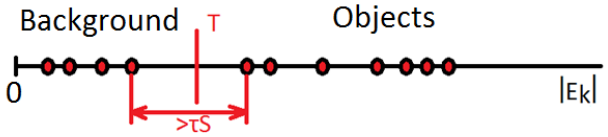

Figure 6. Illustration the data thresholding method. The red points on the line are the points from G_k.

Then, for each cluster from the Bayesian filter tracker, the median error is estimated and checked by the thresholding with threshold T. All the clusters, which average error is more than threshold, are treated as objects, and as background otherwise. The method can be implemented for the subsequent frames as well as for the first and the last frame from the track to ensure robust work of the method.

5 Geographical co-ordinates estimation

Here we propose a geographical co-ordinates estimation method for any point of the image given video footage and synchronised IMU sensor data.

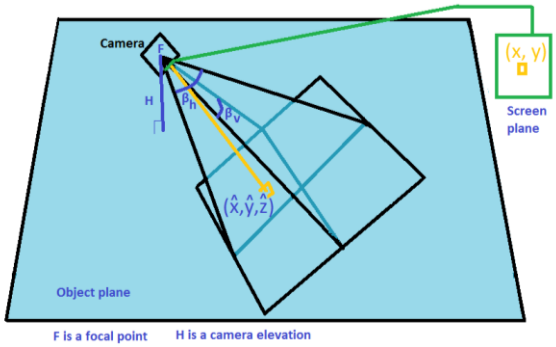

Figure 7. Scene scheme

The scheme of the scene is depicted in figure 7. It is assumed that the surface is ideally horizontal, i.e. the camera has known height above the plain ground. It is pretty correct for instance for sea of field surface. Using geometrical assumptions, we try to estimate the distance to the object, given screen plane inclination, screen object co-ordinates, and camera focal point geographical co-ordinates and height. The screen plane inclination is described by Euler angles [26] provided by IMU. We use pinhole camera model to estimate the distance to the point and estimate the coordinates in world co-ordinate system:

1) normalise screen position of the point (x, y) on the image I_k, having pixel width w and height h:

$$x_n = x - \frac{w}{2}, \; y_n = \frac{h}{2} - y. \quad (22)$$

2) estimate the camera direction in the north-east-down (NED) co-ordinate system [27] using homogeneous image coordinates:

$$n = (\hat{x}, \hat{y}, \hat{z}) = A \begin{pmatrix} \frac{2\tan\left(\frac{\beta_h}{2}\right) x_n}{h} & \frac{2\tan\left(\frac{\beta_v}{2}\right) y_n}{w} & 1 \end{pmatrix}^T, \quad (23)$$

where A is the rotation matrix of the camera which can be derived from the Euler angles captured from sensors, β_h, β_v are horizontal and vertical angles of view, built on the Euler angles obtained from sensors.

3) calculate the scale factor for the normal vector using congruent triangles proportions as $-\frac{H}{\hat{z}}$, where H is a height of the camera relative to the ground.

4) The $\hat{x}$ and $\hat{y}$ components of vector n are used within the Vincenty algorithm [28], along with providing us with the location of the object in the geographical (GPS) co-ordinates.

The most critical assumption that influence the quality of the proposed method is the planarity of the terrain, The measurement error, introduced by discrete pixel

measurements, depends on the angle of view. In the worst case, this kind of error depends on the distance from OOI to the horizon line on the image as $O\left(\frac{HN}{n(n+1)}\right)$, where n is distance (i.e. number of pixels) to the horizon line, N is the "frame size" (height or width). However, in case of "top-down" filming such an error is insignificant.

6 Experiments

To prove the practical applicability of the method, tests with VIVID PETS 2005 data set [29] were carried out, as well as the comparison on the same data set with the alternative multi-target tracking method described in [13]. The data set consists of several video sequences, containing different patterns of multiple vehicles appearance in the video (figure 8). The data set is augmented with the ground truth data for the positions of the objects but only for one of the objects and only on every tenth frame.

Figure 8. VIVID PETS 2005 data set sample frames.

In this comparison we reproduce an experiment from [13], and because of this circumstance, the metrics are chosen the same. 'Match' metric means the part of the frame where presented ground truth data is contained within the bounding box of the object. 'Size ratio' metric means the average ratio between the actually detected bounding box and the ground truth one (ideally 1). The results are given in the table 1. These experiments was carried out with the number of clusters $K = 30$, the object detection parameter $\tau = 10$.

	Match	Size Ratio	Match (Method and data from [13])	Size Ratio (Method and data from [13])
EgTest01	**0.9828**	2.57	0.9500	1.00
EgTest02	**0.9302**	2.47	**0.9302**	1.23
EgTest03	**0.9337**	2.06	0.8588	0.78
EgTest04	**0.9302**	3.51	0.6000	1.19
EgTest05	**0.9080**	0.49	0.8889	0.88

Table 1 Results of the algorithm comparison

These results show the robust match of the localised pattern with the ground truth data. For all the data sets the detection rate exceeds 90%. Higher size ratio means that the detected bounding boxes are larger due to relatively large optical flow near the object and because of the cluster is tightening for several frames on the appearing object, while the rival algorithm relies on the region detection of the object. One should mention that we have an object size ratio less than 1 only in the data set EgTest05 that means underestimation of the bounding box size. In all other cases, the targets are small enough to make even slight bounding box sizes change the bounding box ratio dramatically (figure 9).

Figure 9. Object detection algorithm output.

7 Conclusion

In this work we proposed an algorithm for UAV video analytics capable of the following functionality:

- time-consistent multiple object detection and tracking of the object via newly-proposed Bayesian filter approximation;
- geographical co-ordinates estimation, giving a possibility to locate the objects in the world.

The automatic object detection is performed with the restriction that the object is discernibly moving. The possibility of multiple object tracking enable the algorithm to notify on the appearance of the moving object even in case if another object is already being tracked. The geographical co-ordinates mapping enables us to match the object position on the image with its position on ground. The algorithm delivers robust and accurate results comparing to the rival method for automatic multiple objects detection and tracking, as it was shown in the experimental section.

The following algorithm improvements are considered for future work:

- integration with stereo vision approach for better geo-positioning;
- different prediction and update stage approximations within the Bayesian filter scheme;
- integration with sophisticated trackers capable of occlusions.

Acknowledgment

The research leading to these results has received funding from the EU's Seventh Framework Programme under grant agreement N°607400. The research has been carried out within the TRAX project.

References

[1] P. Viola and M. J. Jones (2011), "Rapid Object Detection using a Boosted Cascade of Simple Features", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Volume: 1, pp.511–518.

[2] L. Sirovich and M. Kirby (1987). "Low-dimensional procedure for the characterization of human faces". *Journal of the Optical Society of America A* **4** (3): 519–524. doi:10.1364/JOSAA.4.000519.

[3] G. Friedrich and Y. Yeshurun (2003), "Seeing people in the dark: Face recognition in infrared images." BMVC.

[4] M. Yang, & G.Zhang 2012). Unsupervised target detection in SAR images using scattering center model and mean shift clustering algorithm. *Progress In Electromagnetics Research Letters*, *35*, 11-18.

[5] M. Piccardi (October 2004). "Background subtraction techniques: a review". IEEE International Conference on Systems, Man and Cybernetics 4. pp. 3099–3104. doi:10.1109/icsmc.2004.1400815

[6] T. Bouwmans (March 2012). "Background Subtraction For Visual Surveillance: A Fuzzy Approach". Chapter 5 in Handbook on Soft Computing for Video Surveillance: 103–134. doi:10.1201/b11631-6.

[7] B. D. Lucas and T. Kanade (1981), An iterative image registration technique with an application to stereo vision. Proceedings of Imaging Understanding Workshop, pages 121—130

[8] F. Flores-Mangas, A. D. Jepson (2013), Fast Rigid Motion Segmentation via Incrementally-Complex Local Models. CVPR 2013: 2259-2266

[9] Z. Kalal, K. Mikolajczyk, and J. Matas (2011), "Tracking-Learning-Detection," *Pattern Analysis and Machine Intelligence* 2011.

[10] Y. Bar-Shalom (Ed.)(1990), Multitarget-Multisensor Tracking: Advanced Applications, ArtechHouse, 096483122, Nonvood, MA.

[11] R. Mahler (2000, June), "A theoretical foundation for the Stein-Winter Probability Hypothesis Density (PHD) multi-target tracking approach," Proc. MSS Nat'l Symp. on Sensor and Data Fusion, Vol. I (Unclassified), San Antonio TX.

[12] M. Schikora, A. Gning, L. Mihaylova, D. Cremers, & W. Koch (2012, July). Box-particle PHD filter for multi-target tracking. In Information Fusion (FUSION), 2012 15th International Conference on (pp. 106-113). IEEE.

[13] Mao, H., Yang, C., Abousleman, G. P., & Si, J. (2014). Automatic detection and tracking of multiple interacting targets from a moving platform. *Optical Engineering*, *53*(1), 013102-013102.

[14] Durrant-Whyte, H.; Bailey, T. (2006). "Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms". *Robotics and Automation Magazine* **13** (2): 99–110. doi:10.1109/MRA.2006.1638022. Retrieved 2008-04-08.

[15] Kalal, Z., Mikolajczyk, K., Matas, J. (2010): Forward-Backward Error: Automatic Detection of Tracking Failures. ICPR 2010: 2756-2759

[16] Harris, C. and Stephens, M. (1988). "A combined corner and edge detector". Proceedings of the 4th Alvey Vision Conference. pp. 147–151.

[17] Kalal, Z., (2001) "Tracking-Learning-Detection" University of Surrey, April 2011 Phd Thesis.

[18] McLachlan, G., and D. Peel (2000). *Finite Mixture Models*. Hoboken, NJ: John Wiley & Sons, Inc..

[19] Wishart, J. (1928). "The generalised product moment distribution in samples from a normal multivariate population". Biometrika 20A (1–2): 32–52. doi:10.1093/biomet/20A.1-2.32. JFM 54.0565.02. JSTOR 2331939.

[20] Gonzalez, R. and Woods, R. (1992) Digital Image Processing, Addison-Wesley Publishing Company, pp 518 - 519, 549.

[21] Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems". Journal of Basic Engineering 82 (1): 35–45. doi:10.1115/1.3662552.

[22] Cevher, V., Guerra, R., Bower, B., Savitsky, T. (2008), Laplace Approximation Rice University, September 11

[23] Stratonovich, R.L. (1960). "Conditional Markov Processes". *Theory of Probability and its Applications* **5**: 156–178. doi:10.1137/1105015.

[24] Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". *Journal of the Royal Statistical Society, Series B* **39** (1): 1–38. JSTOR 2984875. MR 0501537.

[25] Kabsch, W., (1976) "A solution for the best rotation to relate two sets of vectors", *Acta Crystallographica* **32**:922. doi:10.1107/S0567739476001873 with a correction in Kabsch, W., (1978) "A discussion of the solution for the best rotation to relate two sets of vectors", "Acta Crystallographica", "A34", 827–828 doi:10.1107/S0567739478001680.

[26] Euler, L. (1776). Novi Commentarii academiae scientiarum Petropolitanae 20, pp. 189–207 (E478)

[27] Cai, Guowei, Ben M. Chen, and Tong Heng Lee (2011). "Coordinate systems and transformations." *Unmanned Rotorcraft Systems*. Springer London. 23-34.

[28] Vincenty, T. (April 1975a). "Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations". Survey Review. XXIII (misprinted as XXII) (176): 88–93. Retrieved 2009-07-11.

[29] Collins, R.T., Zhou, X., and Seng, K. T. (January 2005) An Open Source Tracking Testbed and Evaluation Web Site. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005).

http://vision.cse.psu.edu/data/vividEval/main.html