Multiple Video Object Tracking Using Variational Inference

Dmitry Kangin, Denis Kolev and Garik Markarian
School of Computing and Communications, Infolab21,
Lancaster University, Lancaster LA1 4YW, U.K.
and
R&D department
Rinicom Ltd.
Riverway House, Morecambe Road, Lancaster, Lancashire LA1 2RX
email: d.kangin@lancaster.ac.uk, g.markarian@lancaster.ac.uk, denis.g.kolev@gmail.com

***Abstract*—In this article a Bayesian filter approximation is proposed for simultaneous multiple target detection and tracking and then applied for object detection on video from moving camera. The inference uses the evidence lower bound optimisation for Gaussian mixtures. The proposed filter is capable of real time data processing and may be used as a basis for data fusion. The method we propose was tested on the video with dynamic background,where the velocity with respect to the background is used to discriminate the objects. The framework does not depend on the feature space, that means that different feature spaces can be unrestrictedly used while preserving the structure of the filter.**

***Keywords*—Multiple object tracking, Bayesian filtering, Variational Gaussian mixtures**

I. Introduction

Up to date, the approximations of the Bayesian filter models gained widely renowned popularity for object tracking. However, the difficulties appear when fully automatic object detection is needed, and even worse, we do not know the number of the objects. In this case, we need to solve multiple object tracking problem with clutter which can be formulated as follows.

Let us have measurements which can be assigned either to one of the objects or to clutter, and each object may include several measurements. The aim is to find the object measurements within the measurement set, to assign them to the objects given pre-defined dependency model, and to determine the characteristics of the objects using the measurements. For each object we assume that the measurements, corresponding to this object, are close to each other in terms of some feature space. The noise measurements are close to each other but not homogeneous, and they are supposed to have substantial differences comparing to the objects' measurements. Also we assume that the noise measurements constitute majority in the measurement set.

The Bayesian filter recursion is decomposed into two steps:

$$p(X^k|Z^{1\dots(k-1)}) = \\ = \int p(X^k|X^{k-1})p(X^{k-1}|Z^{1\dots(k-1))}dX^{k-1}, \quad (1)$$
(Prediction),

$$p(X^k|Z^{1\dots k}) \propto p(Z^k|X^k)p(X^k|Z^{1\dots(k-1)}) \quad (2)$$
(Update).

Here X^k denote hidden variables, or states, of the filter on the k-th step, and Z^k are the visible variables, or measurements, $Z^{1\dots(k)}$ denote the measurements up to the step k. Generally, both hidden variables and measurements can contain the sets of variables, because we can consider many measurements and many targets described by the state. The update stage is usually carried out using either Maximum A Posteriori (MAP) or Minimum Mean Square Error (MMSE) estimate [6].

The stated problem can also be considered as a time-consistent clustering. Given the set of measurements, we assign each of the measurements label, which is either some object or clutter, and consistently update the clustering with the same labels on the same data during the algorithm operation. We emphasise here that in this problem statement we do not assume here point targets, as it is done in many multiple object tracking methods like [1], [2] but perform time consistent clustering.

In section II state-of-the-art methods for multiple object tracking are reviewed. In section III the proposed Bayesian filter is formulated for general, domain unspecific case. After then, in section IV, the algorithm, capable of unsupervised object detection and tracking, is proposed for video tracking. After then, in section V, some experiments were carried out in order to show the capabilities of the algorithm for unsupervised object detection and tracking of the objects from moving camera, followed by the conclusion.

II. State-of-the-art

The variability of the state-of-the-art solutions arises from the different models behind the motion detection, and on miscellaneous approximations even for the same or similar models. For example, for well-known PHD filter there are a lot of recursive approximations based on particle filters and Gaussian mixtures [3], [4], [5].

Classical Bayesian approaches to approximate inference for multi-target tracking include different implementations of Multiple Hypothesis Tracking (MHT) approach [7] and Joint Probabilistic Data Association (JPDA) filter [8]. The problem of tracking in the case of heavy clutter can be solved by R-RANSAC algorithm [9], which extends RANSAC algorithm

[10] to tackle with a case when most of the measurements are clutter.

MHT approach looks for the most probable assignment of the targets to the measurements. It provides a natural solution of the simultaneous tracking and detection problem for the unknown number of targets. However, as the number of hypotheses grow exponentially for each stage, the ways to restrict the count of the hypotheses is needed as well to avoid solving NP-hard problem. One of the possible solutions is to prune the least probable hypotheses [11].

JPDA differs from MHT approach in terms of handling the data association, i.e. unveiling the relation between the targets and measurements. JPDA approach [8] uses the weighted sum of the hypotheses on the association. To make this procedure feasible, gating is applied, which helps to factor out abrupt target state changes which are usually impossible for many problems.

III. The multiple object tracking filter

The multiple object tracking filter proposed in this article propagates time-consistent mixture of Gaussians between the video frames. The clutter and the targets are treated the same way in this framework. Distinguishing between these types of cluster occurs on the detection stage and is not carried out by the tracker.

Here the model is defined within the Bayesian filter framework and then describe the solution based on the variational approximate inference.

First, we define hidden and visible varibles for the formulae (1) and (2).

The visible variables set is the features set $D^k = \{d_1^k, d_2^k, \dots, d_{n_k}^k\}$, built upon the feature point tracks on the k-th frame.

We assume that these visible variables are generated from the mixture of K Gaussians, where the number K is predefined, and the Gaussians are described by the sets μ^k for Gaussian means, Σ^k for Gaussian covariance matrices, and π^k for Gaussian weights within the Gaussian mixture for the k-th frame, where $\mu^k = \{\mu_1^k, \mu_2^k, \dots, \mu_K^k\}$, $\Sigma^k = \{\Sigma_1^k, \Sigma_2^k, \dots, \Sigma_K^k\}$, $\pi^k = \{\pi_1^k, \pi_2^k, \dots \dots \pi_K^k\}, \sum_{i=1}^K \pi_i^k = 1$:

$$d_i^k \sim p(d_i^k|\mu^k, \Sigma^k, \pi^k) = \sum_{j=1}^{K} \pi_j^k \mathcal{N}(d_i^k|\mu_j^k, \Sigma_j^k). \quad (3)$$

We substitute these quantities into the Bayesian recursion as:

$$p(Z^k|X^k) = p(d_i^k|\mu^k, \Sigma^k, \pi^k). \quad (4)$$

The quantity $p(X^k|Z^{1\dots(k-1)})$ is calculated on the prediction state and relies on the probability $p(X^k|X^{k-1}) = p(\mu^k, \Sigma^k, \pi^k|\mu^{k-1}, \Sigma^{k-1}, \pi^{k-1})$ and the previous stage posterior probability $p(Z^{k-1}|X^{k-1})$.

A. Prediction step

The prediction step do not use optimisation techniques and relies on the assumption of the factorisation of the probability

$$\begin{gathered} p(X^k|Z^{1..k-1}) = \prod_{i=1}^K p(\mu_i^k|Z^{1..k-1}) \times \\ \times \prod_{i=1}^K p(\Sigma_i^k|Z^{1..k-1}) \times p(\pi^k|Z^{1..k-1}). \end{gathered} \quad (5)$$

Then, let us consider all these probabilities separately.

We assume the prediction model for means is given by the transition probability

$$p(\mu_i^k|Z^{1..k-1}) = \mathcal{N}(\mu_i^k|U_i\mu_i^{k-1} + T_i, \Psi_i^k), k > 1. \quad (6)$$

Here U_i is the between-frame rotation matrix, T_i is the between-frame transition, both parameters are determined using Kabsch algorithm [16] over the subset of D^k, previously assigned to the k-th cluster, and the details of its application are described further in the object detection section. Ψ_i^k is the covariance matrix over the L^2-errors of the subset of the features set D^k, previously assigned to the k-th cluster, calculated as in formula (12).

The probabilities for the covariance matrices have more convenient representation in terms of the precision matrices. We denote precision matrices $\Lambda_i^k = [\Sigma_i^k]^{-1}$ and assume the following (heuristical) transition

$$p(\Lambda_i^k|Z^{1\dots k-1}) = \mathcal{W}(\Lambda_i^k|W_i^{k-1}, \nu_i^{k-1}), \quad (7)$$

where $\mathcal{W}$ is the Wishart distribution, W_i^{k-1}, ν_i^{k-1} is its parameters, derived from the previous stage, non-negative definite scale matrix and degrees of freedom, correspondingly, $\Lambda_i^{k-1} = (\nu_i^{k-1} - l - 1)W_i^{k-1}$, where l is the feature space dimensionality. The form of the distribution allows to preserve the mean of covariance matrix.

For Gaussian weights, the prediction step is performed as

$$p(\pi^k|Z^{1\dots k-1}) = \mathrm{Dir}\left(\pi^k \left| \frac{n_k \alpha^{k-1}}{\sum_{i=1}^K \alpha_i^{k-1}} \right.\right) \quad (8)$$

Here $\mathrm{Dir}(\cdot)$ is a Dirichlet distribution, and n_k is a number of measurements on the k-th stage.

B. Update step

On the update step, we need to solve the problem of MAP distribution approximation.

We consider

$$\begin{gathered} p(X^k|Z^{1\dots k}) \propto \\ \propto p(Z^k|\mu^k, \Lambda^k, \pi^k) p(\mu^k, \Lambda^k, \pi^k|Z^{1\dots(k-1)}). \end{gathered} \quad (9)$$

To derive this posterior probability, we use approximate inference according to [15]. We consider joint distribution

$$\begin{gathered} p(Z^k, V^k, \pi^k, \mu^k, \Lambda^k) = \\ = p(Z^k|V^k, \mu^k, \Lambda^k) p(V^k|\pi) p(\pi^k) p(\mu^k|\Lambda^k) p(\Lambda^k). \end{gathered} \quad (10)$$

Here V^k, referred as latent variables, are the set of n_k binary vectors of the size $1 \times K$, each summing up to

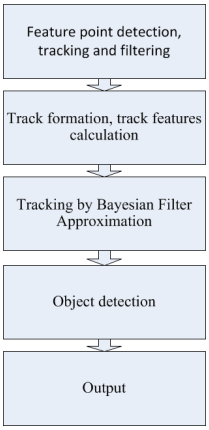

Fig. 1. The algorithm workflow

1, showing which component of the Gaussian mixture the observation is sampled from.

After this stage, one can formulate the variational approximation for the posterior probability factorising between the parameters and latent variables according to [15], which allows to obtain the equations for the iterative update of the parameters.

$$\tilde{p}(V^k, \pi^k, \mu^k, \Lambda^k) = p(V^k)p(\pi, \mu, \Lambda). \quad (11)$$

The solution is provided using Variational Expectation-Maximisation [15] framework.

IV. The video tracking algorithm description

The proposed video tracking filter was applied to multiple target tracking on video. There exist different techniques for video tracking, mostly based on distinction between the clutter and targets, which is a part of tracking model, and featuring data association techniques. Instead of this, here these techniques are avoided, but Mixture of Gaussians model is propagated in a time-consistent way, as a time-consistent clustering. This method allows to decompose object tracking and object detection stages.

The video processing algorithm workflow is depicted in figure 1. First, the feature points detection is carried out, using well known Harris algorithm [12]. Then, the tracks are composed based on the feature points optical flow tracking. Then, the Bayesian filter tracking is carried out, which supports the mixture of Gaussians model update from frame to frame. Then, the object detection is used to factor out the objects. We use the criterion that the object should have discernible movement within the frame using the background velocity model. This model is estimated from the clusters with the largest support. This criterion is pretty straightforward and can be replaced depending on the practical applications. Then, the detected objects are outputted as an outcome from the algorithm. The algorithm 1 shows the overall video analysis and tracking algorithm based variational Bayesian filter approximation.

A. Feature point detection

In this research we used well-known Harris corner point detector [12], combined with sparse pyramidal Lukas-Kanade

Algorithm 1 Variational Bayesian filter approximation algorithm

procedure MAIN
  Tracks = ∅;
  while fetch frame I_k from video stream **do**
    Tracks = calculateAndTrackFeaturePoints (I_k, PreviousTracks);
    Features = calculateFeaturesFromTracks (Tracks);
    FeaturesClusters = clusterTracks (Features, FeaturesClusters);
    detectObjects(Tracks, Features, FeaturesClusters);
  end while
end procedure
procedure NEWTRACKS = CALCULATEANDTRACKFEATUREPOINTS(I_k, PREVTRACKS)
  [FBErr, trackedPoints] = trackLucasKanadeFB(PrevTracks, I_k);
  NewTracks are PrevTracks(FBErr $<$ FBErrThreshold) concatenated with corresponding trackedPoints;
  NewTracksTmp = Detect new points using non-maximum suppression and create new tracks;
  NewTracks = union (NewTracks, NewTracksTmp);
end procedure
procedure FEATURES = CALCULATEFEATURESFROMTRACKS(TRACKS)
  Initialise MatureTracks as tracks with NMature points.
  [Rotation, Shift] = Kabsch (MatureTracks$_{k-NMature+1}$, MatureTracks$_k$), where MatureTracks$_k$ are the points in mature tracks from the $k-$th frame;
  Features = ∅;
  for each track from MatureTracks **do**
    features (track) = [KabschDifference (track, Rotation, Shift), track$_k$];
  end for
end procedure
procedure CLUSTERIDS = CLUSTERTRACKS(FEATURES, FEATURESCLUSTERS)
  for each row in features **do**
    PredictStep (features, featuresClusters); (sec. III.A)
    UpdateStep (features, featuresClusters); (sec. III.B)
  end for
end procedure
procedure DETECTOBJECTS(TRACKS, FEATURES, FEATURESCLUSTERS)
  Initialise MatureTracks as tracks with NMature points.
  [Rotation, Shift] = Kabsch (MatureTracks$_{k-NMature+1}$, MatureTracks$_k$);
  diff = ∅;
  for all t in tracks **do**
    diff(t) = KabschDifference (t, rotation, Shift);
  end for
  VarianceDiff = calculate variance (diff);
  diffSorted = sortAscending(diff);
  calculate velocity threshold T using the formula (14).
  for each cluster k mark it as detected if for most of the cluster's points estimated velocity overcomes the threshold T.
end procedure
procedure DIFF = KABSCHDIFFERENCE (TRACK, ROTATION, SHIFT)
  calculate difference between track$_{k-NMature+1}$ and track$_k$ according to the formula (13).
end procedure

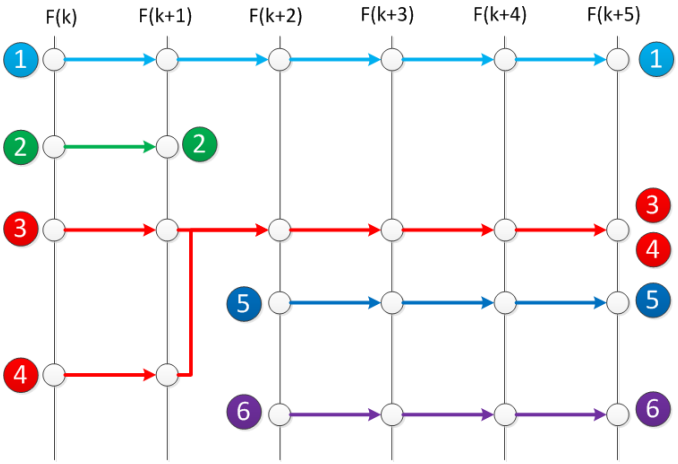

Fig. 2. Track building process

tracker [13]. This combination is reasonable because Harris detector provides points according to Hessian matrix conditioning number, which is inverted when using Lukas-Kanade algorithm for feature point tracking [12], [13].

The quality assessment of the feature point tracking is carried out using forward-backward error concept [14].

To make sure that we can discover new objects we need to prevent the feature points agglomeration in some particular area of the image. For this purpose, non-maximum suppression technique is used over the Harris corner point detector area which allows to eliminate the local maxima close to the highest one.

B. Track formation

Using the information based on Forward-Backward Lukas-Kanade point tracking, we can build tracks representing the movement of the same points between frames.

Tracks are defined as the point sequences $t_i^k = \{f_{j_{s_i}}^{s_i}, \dots, f_{j_k}^k\}$, where s_i is a frame index where the i-th track starts, j_k is the index of the point in the frame feature points list f^k. At each of the stages the points matched by the Lucas-Kanade tracker are attached to the tracks.

In figure 2 all possible track development variants are described.

The track (1) contains the points matched on every stage. The points of the track (2) were matched on the $(k+1)$-th stage, but there were no matches after this stage.

The tracks (3) and (4) have difference below some pre-defined threshold after the stage $(k+2)$, therefore they were merged. Tracks (5) and (6) have appeared from the newly detected points on the stage $(k+2)$ and then were successfully matched.

After each frame the tracks are trimmed to the last $NMature$ points, where $NMature$ is a positive parameter. The tracks which have $NMature$ points are referenced as mature.

C. Object detection

Unlike many state-of-the-art algorithms for object detection with clutter, the proposed algorithm treats clutter within the tracking framework absolutely the same way as the objects itself. The objects are distinguished from the clutter only a posteriori using object model.

To perform the detection of the object, we can rely on the velocity criteria to distinguish the clutter and the object's measurements. The more distinguishable is the velocity of the points cluster from the background the more likely it is to be an object. However, other approach can also be considered, like background model in case for the static camera.

The full list of the criteria is as follows:

- cluster velocity more than some dynamically adjusted threshold, the estimation is described further in this section;
- cluster stability more than some pre-defined threshold, i.e. how many frames the cluster changed less than some threshold (50%) of its points;
- cluster age above some pre-defined threshold, i.e. how many frames the cluster has a support greater than some pre-defined quantity of the points (typically, 0 or 1);
- the cluster size is not greater than some pre-defined threshold (i. e. not larger than $w/2 \times h/2$, where w and h are the width and height of the video frames, respectively).

All the criteria but the first, velocity, look straightforward. Therefore, we concentrate on the description of the first criteria. Consider two matched point sets G^{k-1} and G^k having n_k elements, i. e. the sets with indexed elements where the matching points from the different frames have the same indices. Then we state the least squares problem

$$\begin{array}{l} \sum_{i=0}^{n_k} |G_i^k - \hat{G}_i^k|_{L^2}^2 \to \min_{U,T}, \\ \hat{G}_i^k = G_i^{k-1} U + T, UU^T = 1. \end{array} \tag{12}$$

Here U is an orthogonal rotation matrix, T is a translation vector, $\hat{G}_i^k$ considered as the linear approximation of the movement law from frame to frame.

This problem is widely known and is solved analytically using Kabsch algorithm [16]. This algorithm is deterministic, and to improve this solution, we repeat this procedure several times for the given percentage (e.g. 50%) of the best matched data.

After this stage, we suppose that the rotation and shift model is obtained for the background (we do not consider issues caused by the perspective or other non-linear transformations here). Therefore, we need to distinguish between subtle movement of the background clusters (i.e. clutter) from the significant movement of the object clusters. For this purpose, the following heuristic was developed (figure 3):

- all matched points are sorted by their L^2 error magnitude

$$\epsilon_k = |G_i^k - \hat{G}_i^k|_{L^2}; \tag{13}$$

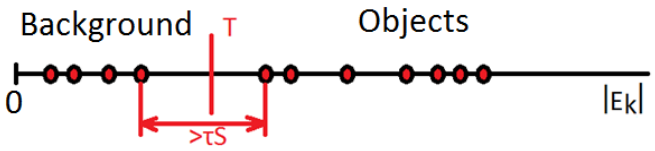

Fig. 3. Threshold adjustment heuristic

Fig. 4. VIVID data set

- the variance S is calculated for the points;
- for $j = 1 \dots n_k$ at the first time when the difference between the neighbouring points' error scalars ϵ_k and ϵ_{k+1} exceeds τS, the error threshold T is initialised as

$$T = (\epsilon_k + \epsilon_{k+1})/2. \tag{14}$$

After this stage, the clusters are selected if the number of the points within the cluster with the estimation error above the threshold is sufficiently large (e.g. $> 50\%$).

V. The Algorithm Performance Evaluation

To prove that the method gives good results comparing to the previous ones, the tests with VIVID PETS 2005 data set were carried out [19]. The data set depicts multiple vehicles which are being tracked and contains marked positions of one vehicle for each 10-th frame. The sample data frames from VIVID data set are depicted in the figure 4.

Fig. 5. Output sample

TABLE I. Results of the Experiments

	EgTest01	EgTest02	EgTest03	EgTest04	EgTest05
Match	0.9717	0.9225	0.8466	0.9005	0.8642
Size ratio	2.59	3.13	1.12	3.63	0.54
Match (method [17])	0.9500	0.9302	0.8588	0.6000	0.8889
Size ratio (method [17])	1.00	1.23	0.78	1.19	0.88

The results of the experiment presented in table I are compared with those from the article [17], which method is based on the set of Kalman filters for each of multiple targets accompanied with data association techniques. The detection is provided by estimating the background movement model based on optical flow.

One can see stable pattern localisation in the proposed algorithm. While the rival algorithm gives only 60% on the EgTest04 data set, given algorithm yields 90%. One of the output samples with marked bounding boxes is shown in figure 5.

VI. Conclusion

The method proposed in this article unites Bayesian filtering approach to simultaneous object detection and tracking with variational approximation. The result shown in the experimental section show the stability and robustness of the algorithm outcome. The proposed Bayesian filter approximation has a good generalisation power. It can be used with different feature spaces, and also can be accompanied with different object detection algorithms.

The research leading to these results has received funding from the EUs Seventh Framework Programme under grant agreement N607400. The research has been carried out within the TRAX project.

References

[1] M. Schikora, A. Gning, L. Mihaylova, D. Cremers, W. Koch, Box-Particle Hypothesis Density Filter for Multi-Target Tracking, IEEE Transactions on Aerospace and Electronic Systems, Vol. 50, No. 3, July, pp. 1660 - 1672, 2014.

[2] D. Salmond. Tracking and guidance with intermittent obscuration and association uncertainty. FUSION 2013: 691-698

[3] Mahler R.; "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, June 2000.

[4] Vo, B.-N.; Ma, W.-K., "The Gaussian mixture Probability Hypothesis Density filter," IEEE Trans. Signal Processing, IEEE Trans. Signal Processing, Vol. 54, No. 11, pp. 4091-4104, 2006.

[5] Clark, D.; Ruiz, I.T.; Petillot, Y.; Bell, J.; Particle PHD filter multiple target tracking in sonar images Aerospace and Electronic Systems, IEEE Transactions on Volume 43, Issue 1, January 2007 Page(s):409 - 416

[6] Jaward, M., L. Mihaylova, N. Canagarajah, and D. Bull. "Multiple object tracking using particle filters." In Aerospace Conference, 2006 IEEE, pp. 8-pp. IEEE, 2006.

[7] Reid D. An algorithm for tracking multiple targets, IEEE Trans. on Automatic Control, vol. 24, issue 6, pp. 423-432, Dec.1979.

[8] T. E. Fortmann, Y. Bar-Shalom, M. Scheffe, Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association, IEEE J. Oceanic Eng., OE-8 (3), 1983

[9] Niedfeldt, Peter C., and Randal W. Beard. "Multiple target tracking using recursive RANSAC." In American Control Conference (ACC), 2014, pp. 3393-3398. IEEE, 2014.

[10] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography" (PDF). Comm. of the ACM 24 (6), June 1981: pp. 381-395.

[11] Roy, Jean. "Towards multiple hypothesis situation analysis." In Information Fusion, 2007 10th International Conference on, pp. 1-8. IEEE, 2007.

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

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

[14] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Forward-backward error: Automatic detection of tracking failures." In Pattern Recognition (ICPR), 2010 20th International Conference on, pp. 2756-2759. IEEE, 2010.

[15] Bishop, C. M. Pattern Recognition and Machine Learning, pp. 474-486, Springer, 2006

[16] Kabsch, W. "A solution for the best rotation to relate two sets of vectors", 1976, Acta Crystallographica 32:922. doi:10.1107/S0567739476001873 with a correction in Kabsch, W., "A discussion of the solution for the best rotation to relate two sets of vectors", "Acta Crystallographica", "A34", 1978, 827828 doi:10.1107/S0567739478001680.

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

[18] D.G. Kolev D. Kangin, G. Markarian. Data Fusion for Unsupervised Video Object Detection, Tracking and Geo-Positioning In Fusion 2015 Conference, Washington D.C., 2015

[19] Collins, R.T., Zhou, X., and Seng, K. T. An Open Source Tracking Testbed and Evaluation Web Site. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005), January 2005. http://vision.cse.psu.edu/data/vividEval/main.html