Passive Multitarget Tracking with Cameras

Dann Laneuville, Adrien Nègre

DCNS Research
DCNS
40-42 rue du Docteur Finlay
75732 Paris
dann.laneuville@dcnsgroup.com
adrien.negre@dcnsgroup.com

Abstract:

This paper considers large areas surveillance and 3D tracking with passive data, obtained here by geographically distributed cameras. The first step, at a camera level, is to detect moving objects in the video sequence and we propose a very simple, fast and efficient approach: a pixel level background subtraction technique to segment foreground pixels and a region level process where segmented pixels are connected into objects. Experiments on real costal environment videos of this method demonstrate similar results compare to more sophisticated approaches with a very low processing time, which allows processing high resolution images. The second step is then to obtain 3D tracks by merging the elementary detections issued by the cameras and we use a suitably modified Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter approach in a centralized fusion scheme. We present some results with simulated data obtained on a realistic test scenario.

1 Introduction

The aim of this paper is to discuss and exhibit solutions to the problem of large areas image-based surveillance, here and without loss of generality, in a maritime context. In the last decade, increased concern about terrorism, drug smuggling or stowaways has lead to study harbour or coastal area surveillance systems. In such systems, information delivered by a group of sensors is fused to achieve a situation picture (classified tracks) augmented by an anomalous behaviour detection process. We focus here on surface targets and propose to use geographically distributed cameras as a complement to the traditional AIS receivers or coastal radars that are commonly used to monitor maritime traffic. Our approach is motivated by the fact that small boats may be hard to detect with radars, especially in coastal environment and, if evil-minded, will not transmit their AIS position. Furthermore, cameras are, in addition of the surveillance task, able to zoom and to provide detailed images of some particular targets that will be the basis of any identification process. We restrict our attention in this paper on the *detection* process at a camera level and the *tracking* process at a centralized fusion level.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4

A lot of work has been done in the last decade on visual tracking and object extraction is mainly performed through two approaches: background subtraction or colour-based tracking also named kernel-based object tracking. In the latter (see for instance [CRM03], [ZB09] and [Pe02]), the search of a window whose colour or/and texture content matches a reference histogram model is performed whereas in the former, each pixel of the scene is classified as belonging to the foreground, i.e. correspond to moving objects, or to the background (see for instance [ZS03], [EHD00] and [SG99]).

Section 2 will outline the detection process which aims to detect the different moving objects, and we show some results obtained on real images with our fast background subtraction algorithm. Section 3 is devoted to the second process which consists in obtaining 3D tracks by processing the detections obtained from different cameras and we show some results obtained with a centralized GM-PHD approach on simulated data.

2 Video extraction of moving objects

As colour-based approaches require a manually defined initial region on the object we want to track (though [Pe02] presents an automatic procedure in a specific case), we use the background subtraction approach to segment all moving objects. To deal with variable backgrounds this technique has suitably been modified and [ZS03] presents a segmentation method where the background of the scene is modeled by a dynamic texture with a first order Auto Regressive Moving Average (ARMA) model. Another approach is the use a Gaussian mixture or a Gaussian kernel to model the density of each pixel ([SG99], [EHD00]) to deal with cluttered environment (swaying trees or bushes).

2.1 Background subtraction: foreground mask

In a costal environment application, the sea represents the variable background and the different objects evolving on the sea are the foreground objects of interest. In our case of surveillance, the need of a large Field of View (FOV) to cover a wide area combined with the fact that small objects are the targets of interest requires images with a sufficient resolution such as 1920×1080, precluding the use of the algorithm described in [ZS03]. Indeed, [ZS03] shows result obtained with a resolution of 160×120 with a processing time of several seconds per frame and is thus not able to process high resolution images in real time on a reasonable computer. The idea is then to use the simplest background subtraction method at a pixel level and we propose a robust Kalman filter with an EM step to estimate the background.

Let $I(t)$, $I(t) \in \mathbb{R}^m$, be the gray scale image of size m at time t of an incoming video which contains moving objects. Let $i \in \{1..m\}$ be a pixel location. Our segmenting algorithm is based on a simple pixel independent approach. We propose to model the gray scale behavior of each pixel of the background by the following scalar linear dynamic system:

$$\begin{cases} x(t+1) = a(t)x(t) + v(t) \\ z(t) = x(t) + w(t) \end{cases} \quad (1)$$

where $x(t)$ is the pixel value at location i, $a(t)$ is the model parameter, $v(t)$ is the process noise, assumed to be $N(0, q(t))$ and $w(t)$ the measurement noise $\sim N(0, r(t))$. This is the scalar counterpart of the background model described in [ZS03] that will be denoted as "ZS method" in the sequel. In order to estimate on line the parameters a, q, r of model (1), we add an EM step and to avoid updating this model by pixels coming from the foreground, which will appear as outliers, we use a robust Kalman filter formulation as presented in [TTS07]. To take into account lighting changes, the EM step is processed on the last N frames. The final detection is done by comparing to a threshold the difference between the background model and the current pixel value. Pixels which overtake the $k\sigma$ threshold (typical $k = 3$ to 5) are considered as moving pixels and form the foreground mask. The entire method is illustrated by the following figure:

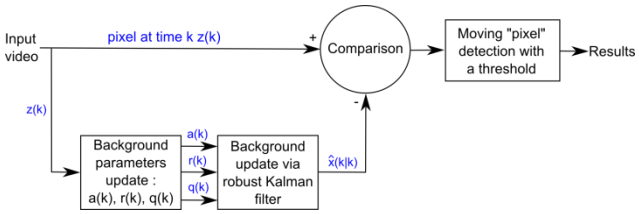

Fig1. Video extractor block diagram

This very simple method, though it may be a little less performing, avoids the drawbacks of the complex global background subtraction techniques which take into account the pixels' correlation. First, these methods require the use of a training video to learn the parameters of the background model. This training video has the constraint to be "empty", i.e. must not contain any moving objects. But the major drawback is the complexity of this global approach, contrary to the pixel independent approach, that requires inversion of very large matrices (inverse of $m \times m$ matrices, m being the size of the image), precluding its utilization on high resolution images. Compare to the Gaussian mixture approach, we will have some false detections in reflecting conditions or great sea but will cancel most of them after the connecting phase with a threshold on the number of pixels at the region level.

2.3 Example of result on real images

In the experiments, the image size is set to $m = 320 \times 240$ which is a low resolution, but is a maximum value for the ZS method on our computer (PC 3.00 Ghz and 2 Go RAM) with a Matlab implementation. Detection results are a binary image (the foreground mask) with white pixels representing moving ones. The ZS method was run with parameter $n = 100$, n being the number of principal components of the state vector that models the background (the authors in [ZS03] used smaller images with $m = 160 \times 120$ and $n = 180$). To initialize this method, we have to use a specific empty training video (without moving objects) and the CPU time per frame is 2.5 sec (it would be around 5 sec if we used $n = 180$). In comparison, our algorithm is simply initialized with the first image of the incoming video and only requires 0.025 sec per frame.

Though we did not carry out detailed performance analysis, the different cases we have studied and this example show that our segmentation algorithm is fairly good with a two magnitude of order reduction in the computation time. Indeed, our method allows for processing high resolution image. The following figure shows an example obtained on a video with a 720×576 resolution. Some sea clutter, easily canceled after the connecting phase with size based filtering, is visible on the foreground mask on Fig.3 below. Computation time is then about 0.13 sec per frame. On a 1920×1080 resolution, we obtain a 0.67 sec processing time (Matlab implementation).

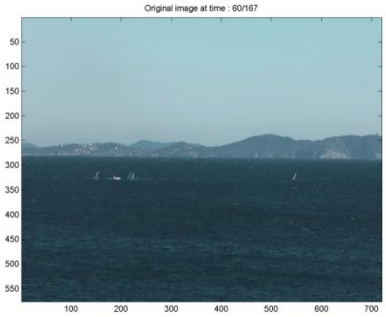

Fig2. Original image

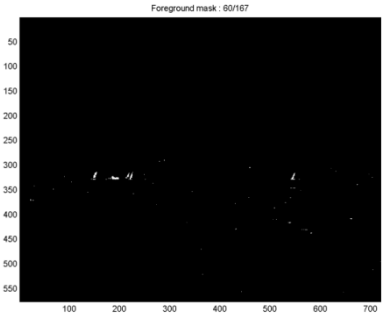

Fig3. Foreground mask

3 Data fusion

This section outlines the GM-PHD tracking process presented in [LH10] to fuse the elementary detections in a centralized fusion scheme. In our surveillance application, the target state consists of 3D position and velocity: $X = [x\ y\ z\ v_x v_y v_z]'$ and we simply use the classical near constant velocity (CV) model to describe its evolution with dt the time interval between two reports at the fusion node coming from the different cameras S_i. The measurement consists of two angles (az, ele) as being target centers obtained from the foreground mask after the connecting phase to group pixels into region. Due to the non linear nature of the measurement model consisting of two angles related to a 3D Cartesian state, we use the UKF update step to compute, for each Gaussian density, the predicted measurement, its covariance matrix and the filter gain.
As commonly done in passive applications (like in BOT), where no range measurement is available, we choose an a priori distance d_0 with a large uncertainty σ_{d0} to be able to initialize a Cartesian state on a single detection $z = (az, ele)$ from a camera. This is achieved by means of a UT transform where (az, ele, d_0) and the associated covariance matrix are converted to Cartesian coordinates, state and covariance. This distance is set to $d_0 = 2\ km$ with $\sigma_{d0} = 500\ m$. This concerned the position part of the state vector and its covariance matrix. The velocity is initialized to zero with a large covariance, set to $\sigma_{vx0} = \sigma_{vy0} = \sigma_{vz0} = 10\ ms^{-1}$. Finally, this amounts to start the state vector with

$$X_0 = [x_0\ y_0\ z_0\ 0\ 0\ 0]' \text{ and with } P_0 = \begin{bmatrix} P_{xyz} & 0_{3\times3} \\ 0_{3\times3} & diag(\sigma_{vx0}^2\ \sigma_{vy0}^2\ \sigma_{vz0}^2) \end{bmatrix}$$

3.1 Scenario

The scenario under consideration is shown on Fig.4, here below. Targets are plotted with a number at their starting point and the three cameras are indicated with a star marker. Targets are moving with a speed between 10 and 15 knots and may slowly maneuver with a turn rate of 2°/s. Note we did not take advantage that the targets are surface targets, except for the initial covariance velocity. It could have been as well aerial targets.

3.2 Measurements simulation

Three cameras are used, one located at the origin and at an altitude of $50\,m$ and the two others symmetrically disposed at $(+/-2000\,m, 500\,m, 50\,m)$. The bore sights angles of the cameras are respectively (0°,-5°), (20°,-5°) and (-20°,-5°) with a FOV of 120° × 30°. The measurements consist of azimuth and elevation angles with $\sigma_a = \sigma_e = 10^{-3} rad$ and are taken with a frame rate of $T = 3$ sec for each sensor and sent without delay at the fusion node. There is one second of time lag between each camera report so that the fusion node receives a report every second ($dt = 1\,s$). Each camera report consists of target detections, if detected, and false alarms. The false alarms, corresponding to "waves" detection in the video extraction process of our application, are supposed to be uniformly distributed in the image frame and are set to a number of $\lambda = 20$ per image, which is quite a high value.

3.2 Example of results

The process noise intensity is set to $\sigma_v^2 = (0.1)^2$. The parameters of the PHD filter are set to: Initial weight $w_0^Y = 2.10^{-6}$, Probability of detection $P_d = 0.95$, Probability of survival $P_s = 1$, Pruning threshold $\tau_p = 1.10^{-6}$, Merging threshold $U = 25$, Confirmation threshold $\tau_c = 0.9$.

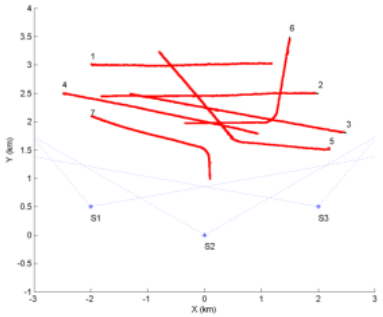

Fig4. Estimated tracks (True: thin black, estimated: thick red)

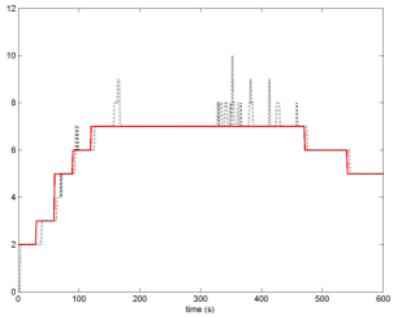

Fig5. Cardinality

Fig.4 shows the estimated tracks and the ground truth. The filter globally performs very well, though a zoom on certain parts (crossings) could show some discrepancies that are visible on the cardinality plot shown on Fig.5.

4 Conclusion

The proposed segmentation and fusion algorithms in this paper are a promising candidate to the challenging problem of large area passive surveillance and multitarget tracking with cameras. For the detection process, we have used a simple and efficient segmentation algorithm which allows processing of high resolution images in a reasonable time with fairly good results. Then we have used a modified GM-PHD filter to fuse elementary detections coming from the different cameras. Future work will consist in extending the background subtraction to color imagery and thus improve the detection process. A wake suppression process is also under study: this is the shadow analogue problem in people or car tracking applications. Improvements for the GM-PHD are also under study to take into account the fact that an object may be not simultaneously visible by every camera.

Bibliography

[CRM03] Comaniciu, D.; Ramesh, V.; Meer, P.: Kernel-Based Object Tracking, IEEE Trans. Pattern Ana. Machine Intell., 25(5):564-577, 2003.

[EHD00] Elgammal, A.; Harwood, D.; Davis, L.: Non-parametric model for background subtraction, Proc. Europ. Conf. Computer Vision, 2000.

[LH10] Laneuville, D.; Houssineau, J.: Passive Multi Target Tracking with GM-PHD Filter, Proc. Int. Conf. on Information Fusion, 2010.

[Pe02] Pérez, P.; Hue, C.; Vermaak, J.; Gangnet, M.: Color-Based Probabilistic Tracking, Proc. Europ. Conf. Computer Vision, 2002.

[SG99] Stauffer, C.; Grimson, W.E.L.: Adaptive background mixture models for real-time tracking, Proc. Computer Vision and Pattern Recognition, 2003.

[TTS07] Ting, J-A.; Theodorou, E.; Schaal, S.: Learning an Outlier-Robust Kalman Filter, CLMC Technical Report Number: TR-CLMC-2007-1, University of Southern California, 2007.

[ZB09] Zhang, S.; Bar-Shalom, Y.: Robust Kernel-Based Object Tracking with Multiple Kernel Centers, Proc. Int. Conf. on Information Fusion, 2009.

[ZS03] Zhong, J.; Sclaroff, S.: Segmenting foreground objects from a dynamic textured background via a robust Kalman filter: Proc. Int. Conf. Computer Vision, 2003.