http://dx.doi.org/10.1117/12.2195224

Detecting Abandoned Objects using Interacting Multiple Models

Stefan Becker, David Münch, Hilke Kieritz, Wolfgang Hübner, and Michael Arens

Fraunhofer Institute for Optronics, System Technologies, and Image Exploitation IOSB
Gutleuthausstr. 1, 76275 Ettlingen, Germany

ABSTRACT

In recent years, the wide use of video surveillance systems has caused an enormous increase in the amount of data that has to be stored, monitored, and processed. As a consequence, it is crucial to support human operators with automated surveillance applications. Towards this end an intelligent video analysis module for real-time alerting in case of abandoned objects in public spaces is proposed. The overall processing pipeline consists of two major parts. First, person motion is modeled using an Interacting Multiple Model (IMM) filter. The IMM filter estimates the state of a person according to a finite-state, discrete-time Markov chain. Second, the location of persons that stay at a fixed position defines a region of interest, in which a nonparametric background model with dynamic per-pixel state variables identifies abandoned objects. In case of a detected abandoned object, an alarm event is triggered. The effectiveness of the proposed system is evaluated on the PETS 2006 dataset and the i-Lids dataset, both reflecting prototypical surveillance scenarios.

Keywords: Interacting Multiple Models, Abandoned Object Detection, Visual Surveillance

1. INTRODUCTION

One challenge that arises due to the increase of surveillance cameras, is the overload of visual information. Consequently, keeping a constant situation awareness for human operators is very hard and accordingly an active support by automated video analysis is needed. In recent years, many methods have been proposed for automated abandoned objects detection. These methods not only differ in their target application like traffic monitoring, retail or public safety, but in addition how such an event is defined. For example, Beynon et al.[1] define an abandoned object as stationary object away from anyone responsible for it. In the work of Bird et al.[2] an abandoned object is defined as being stationary for some time threshold and not being touched by someone and Ferrando et al.[3] set a split from a person as prerequisite for a non-human, static object to define an abandoned object. Nevertheless, the aforementioned definitions can only cover a few existing real life situations. Here we define similar to Ferrando et al. an abandoned object as carried by a person to scene and then left by this unique assigned person.

Apart from the varying definition, approaches for detecting left-luggage can roughly be divided into two groups. On the one hand, there are methods which rely on tracking information and require detection and tracking as intermediate step, such as the approach of Lu et al.[4] They detect and track moving objects using shape and color features in combination with Kalman-based filtering. Moving objects are classified with a support-vector-machine using eigen-features extracted from the binary shapes. Afterwards, they perform a HMM-based activity recognition for package ownership analysis. On the one hand, there are nontracking-based systems, like the approach of Porikli et al.[5] or Lin et al.[6] In order to detect stationary objects dual background models are used. The two background models, one long-term and one short-term background model, are constructed by setting different learning rates. Apart from these different groups, there are also several approaches, which combine these concepts. For example the framework presented by Tian et al.[7] In their work, the tracking output is used to reduce false positives from the applied background subtraction and foreground analysis.

The proposed approach for automatically detecting abandoned objects in surveillance scenarios is also a combination from the concepts outlined above. The overall processing pipeline consists of two major parts. First, a

Further author information (Send correspondence to Stefan Becker):
E-mail: stefan.becker@iosb.fraunhofer.de

detection and tracking component and second a nonparametric background model. For person tracking an Interacting Multiple Model (IMM) filter is used. The IMM filter serves for modeling varying motions of persons and simultaneously estimates different motion states of tracked persons. The state estimates are used for controlling the pixel-wise update probabilities of a background model and the estimated positions define regions of interest in which drop-off event are detected.
The paper is structured as follows: The proposed processing pipeline is presented in section 2. In section 3 the effectiveness is shown on the public available datasets and section 4 concludes.

2. ABANDONED OBJECT DETECTION

In the following section the processing pipeline of the proposed combination of a detection and tracking component and a nonparametric background model is described in detail.

2.1 Person Detection

In order to detect an abandoned object defined by a split from a person, a person detection component is needed. As all subsequent processing steps are based on this component, it has to provide robust, reliable, and coherent results. For localizing persons in the image, the first step is to select image regions likely to contain persons. After this candidate generation process, the selected regions are classified as containing a person or background. The most common approach is the sliding window based classification. It consists in exhaustively scanning of the image and a binary classification for every window location and scale. The final person location is determined by non-maximum suppression. In this work, without preferring a specific person detector, we use a detector based on the work of Dollár et al, Benenson et al.[8] and Kieritz et al.[9] following this sliding window paradigm. Thus, our classifier is trained with AdaBoost using integral image features. Other person detectors mainly differ in terms of image descriptors, classifiers, and models. Nowadays, descriptors such as Haar, HOG, and LBP and classifiers such as support-vector-machine, random forest, CNNs are very common. For a detailed review on person detectors the reader is referred to Gerónimo et al. and for a state-of-the-art performance evaluation to Dollár et al.[10] or to Enzweiler et al.[11] A person's state, as based on the detector output, can be summarized as rectangular bounding box centered at the person location. Hence, the state of a person can be defined as follows:

$$x_k = (x, y, s, \dot{x}, \dot{y}, \dot{s}, \ddot{x}, \ddot{y}, \ddot{s}). \tag{1}$$

Here x, y is the center position in the image space and s the scale, $\dot{x}, \dot{y}, \dot{s}$ and $\ddot{x}, \ddot{y}, \ddot{s}$ are the velocities and accelerations along the state space dimensions. The detector output is directly used as measurements for the tracking component.

2.2 Tracking

In real world scenarios it is reasonable to assume that the dynamics of persons can change from time to time. For example, a person can stand still, walk, or run. Such varying system characteristics cannot be captured by a single linear model. As a solution a system is considered to be composed of multiple independent models, where the currently active model is one from a discrete set of n options, which are denoted by $M = \{M^1, \ldots, M^n\}$. Some prior probability μ_0^j for each model M^j and the state transition probability between time index $k-1$ and k from model i to model j (denoted by $p_{ij} = P(M_k^j | M_{k-1}^i)$) are assumed to be known. The transition probability matrix p_{ij} can be interpreted as a first order Markov chain characterizing the mode transitions. Hence systems of this type are commonly referred as Markovian switching systems (Bar-Shalom et al.[12]). For an optimal filtering behavior an optimal filter for every possible model sequence is required. Hence, some kind of approximations are needed in practical applications. In the proposed system an Interacting Multiple Model (IMM) filter (Bar-Shalom et al.[12]) is used. IMM filters are a popular choice for the state estimation problem, when the underlying system behavior is describable by finite states. IMM filters can also be used in situations, in which the system structure is unknown or its parameters have to be estimated from a set of candidate models. An overview on other Kalman filter extensions also covering different motion types is given by Li et al.[13] The closed form recursive solution for the state estimation problem of linear systems provided by a discrete-time

Kalman filter[14] can be modified in the following way (see for example Hartikainen et al.[15] or Bar-Shalom et al.[12] for a more detailed derivation) *:

$$x_k = F_k^j x_{k-1} + w_k^j \text{ and} \tag{2}$$
$$y_k = H_k^j x_k + r_k^j. \tag{3}$$

Here, the effective model or mode in time step $k-1$ is denoted by j. F_k is the state transition matrix which applies the effect of each system state parameter at time $k-1$ on the system state at time k. y_k is the measurement and H_k is the measurement model matrix that maps the state parameters into the measurement domain. $w_k \sim N(0, Q_k)$ is the process noise and $r_k \sim N(0, R_k)$ is the measurement noise. Both noise terms are assumed to be zero mean Gaussian white noise with covariance Q_k and R_k respectively. An IMM filter cycle basically consists of three major steps: interaction (mixing), filtering and combination (see figure 1).

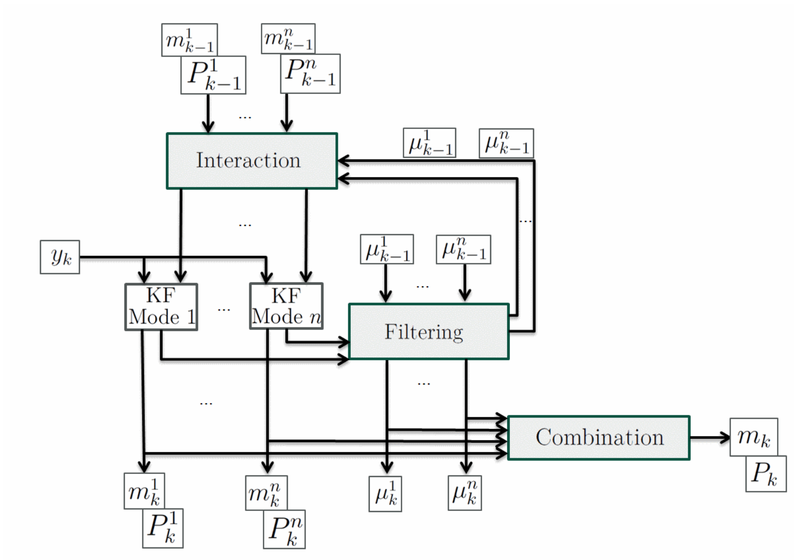

Figure 1. A functional diagram of the IMM filter with the main processing blocks of interaction (mixing), filtering and combination.

In the interaction stage, the initial conditions for a model are obtained by mixing the state estimates produced by all filters conditioned at the best fitting model. A underlying assumption is that a particular model is the right model at the current time step. In detail, the mixing probabilities $\mu_k^{i|j}$ for each model M^i and M^j are calculated as $\mu_k^{i|j} = \frac{1}{\bar{c}_j} p_{ij} \mu_{k-1}^i$ with $\bar{c}_j = \sum_{i=1}^n p_{ij} \mu_{k-1}^i$. Thereby, μ_{k-1}^i is the probability of model M^i in the time step $k-1$ and $\bar{c}_j$ is a normalization factor. Then the mixed inputs (means and covariances) are computed for each filter as

$$m_{k-1}^{0j} = \sum_{i=1}^n \frac{1}{\bar{c}_j} \mu_k^{i|j} m_{k-1}^i \text{ and} \tag{4}$$

$$P_{k-1}^{0j} = \sum_{i=1}^n \mu_k^{i|j} \left(P_{k-1}^i + (m_{k-1}^i - m_{k-1}^{0j})(m_{k-1}^i - m_{k-1}^{0j})^T \right). \tag{5}$$

*The mathematical notation follows the notation used in Särkkä[16]

Here, m^i_{k-1} and P^i_{k-1} are the updated mean and covariance for model i at time step $k-1$.
Then filtering stage consists of applying a standard Kalman filtering ($KF_p(.)$ and $KF_u(.)$) for each individual model. Initialization is done with m^i_{k-1} and P^i_{k-1}.

$$\left[m_k^{-,i}, P_k^{-,i}\right] = KF_p(m_{k-1}^{0j}, P_{k-1}^{0j}, F_k^i, Q_k) \tag{6}$$

$$\left[m_k^{i}, P_k^{i}\right] = KF_u(m_k^{-,i}, P_k^{-,i}, H_k^i, R_k^i) \tag{7}$$

In addition to mean and covariance estimates, the likelihood of the measurement for each filter is computed as

$$\Lambda_k^i = N(v_k^i, 0, S_k^i) = \frac{1}{\sqrt{((2\pi)^l \left|S_k^i\right|)}} exp\left(-\frac{1}{2}(v_k^i)^T (S_k^i)^{-1} v_k^i\right). \tag{8}$$

Here, v_k^i is the innovation or measurement residual and $S_k^i = (H_k^i P_k^{-,i}(H_k^i)^T + R_k^i)$ the corresponding residual covariance for model M^i in the Kalman filter update step. The model probabilities are adapted according to

$$\mu_k^i = \frac{1}{c}\Lambda_k^i \bar{c}_i\ , \tag{9}$$

where $c = \sum_{i=1}^n \Lambda_k^i \bar{c}_i$ is a normalizing factor.
The final step of the IMM filter is combination. The combined estimate for the state mean and covariance is computed as

$$m_k = \sum_{i=1}^n \mu_k^i m_k^i \text{ and} \tag{10}$$

$$P_k = \sum_{i=1}^n \mu_k^i \left(P_k^i + (m_k^i - m_k)(m_k^i - m_k)^T\right). \tag{11}$$

In our experiments, we used a set of three different motion models. A constant position, a constant velocity, and a constant acceleration model. In addition to the observed measurement from the person detector, a color histogram (HSV color space) is used. The histogram is calculated from the detection bounding box. In order to ensure an exclusive assignment of detections and person tracks and thereby consistently maintain person identities over time the Munkres algorithm[17] is applied based on the weighted sum of the Bhattacharyya distance of the histograms and Euclidean distance of the locations. For every assigned detection the IMM estimates the current best fitting model. Thus, the IMM filter does not only help to better describe the varying dynamics of persons, but it further builds the basis to distinguish between persons that stand still, walk, or run using the state estimate from the best fitting model. On the basis of the information provided by the complete detection and tracking component, a nonparametric background model is introduced for detecting abandoned objects.

2.3 Background Model

In this section, the Context Aware Pixel-Based Adaptive Segmenter (CAPBAS) is introduced. The approach follows a non-paramertic paradigm to model the background and extends the work of Hofmann et al.[18] (Pixel-Based Adaptive Segmenter; PBAS), which is built on the approach introduced by Barnich et al.[19] The background is modeled by a history of recently observed pixel values. The basic operation of background subtraction methods is to separate pixels affected by motion defined as foreground (FG; $F(p_i) = 1$) from the static information defined as background (BG; $F(p_i) = 0$). Here, the decision is done by comparing the current image and the background model $B(p_i)$ with a per pixel threshold $R(p_i)$. In order to allow a gradual change of the scene, the background model is updated depending on a per-pixel learning parameter T_{p_i}. Hofmann et al. introduce an update scheme for the decision threshold $R(p_i)$ and the learning parameter $T(p_i)$ which is based on an estimate of the background dynamics. The essential idea of our CAPBAS approach is a coupling between the dynamic update and context information provided by the tracking component described in the previous section.

Figure 2. On the left an example image from the PETS2006 dataset (S1-T1). On the right a visualization of the corresponding local distribution of $T(p_i)$.

Updating means that a value from background model $B_k(p_i)$ from the history of N observed background values is randomly replaced by the current pixel value $I(p_i)$. This update is only performed with probability $prob = 1/T(p_i)$. Otherwise no update is carried out at all. Therefore, the parameter $T(p_i)$ defines the update rate. The higher $T(p_i)$, the less likely a pixel will be updated. This update is done for foreground pixels and for randomly chosen neighborhood pixel. Thus, certain foreground pixels at the boundary will gradually be included into the background model. Hence, depending on the update parameter $T(p_i)$ every foreground object will be eaten-up from the outside after a certain time. The advantage of this property is that erroneous foreground objects will quickly vanish, but this will also assign slowly moving foreground objects to the background. Hofmann et al. alleviate the problem by introducing a dynamic controller for $T(p_i)$, such that the probability of background learning is (slowly) increased when the pixel is background and (slowly) decreased when the pixel is foreground. The controller for measuring the background dynamics is created based on the history of minimal decision distances. The average of these values is the measure of the background dynamics. Figure 2 shows the update parameter $T(p_i)$ for slowly moving, but significantly large foreground object. For the pixels corresponding to the person, a high $T(p_i)$ indicates low update probability. Such an adaption of the update probability, works pretty well for relative large foreground objects, but for our goal of detecting abandoned objects carried by a person a modification is required. Especially, in situation where the person which carries an object stands still or rather is loitering. This context information for preventing a too fast integration into the background model is available due to the tracking component. Thus, depending on the current best fitting dynamic model the pixel-wise update probabilities inside a small region of interest can be adapted. The region is defined by the state of the person, in particular based on the center location and scale. The width is calculated with a fixed aspect ratio from the current estimated scale. The update of the decision threshold $R(p_i)$ remains unaffected.

Figure 3 shows an example image of a resulting foreground mask in case of a slowly moving or rather standing person with and without context information. On the left the resulting foreground mask obtained with no tracking information is shown, and on the right with context information. When the tracking state is not used the person and the carried object are almost fully integrated in the background model. As outlined in section 2.2, a set of three different motion model is used. For adapting the update probability, the constant position model is assigned to a very fast increasing of $T(p_i)$. Hence, an inclusion of carried objects into the background model is prevented by standing or loitering. In case of the other two models, the region covered by the tracked person leads to only a small increase of $T(p_i)$. Compared to Hoffman et al. the strength of the adjustment depends not only on the dynamic estimator $\bar{d}_{min}(p_i)$, and is defined as:

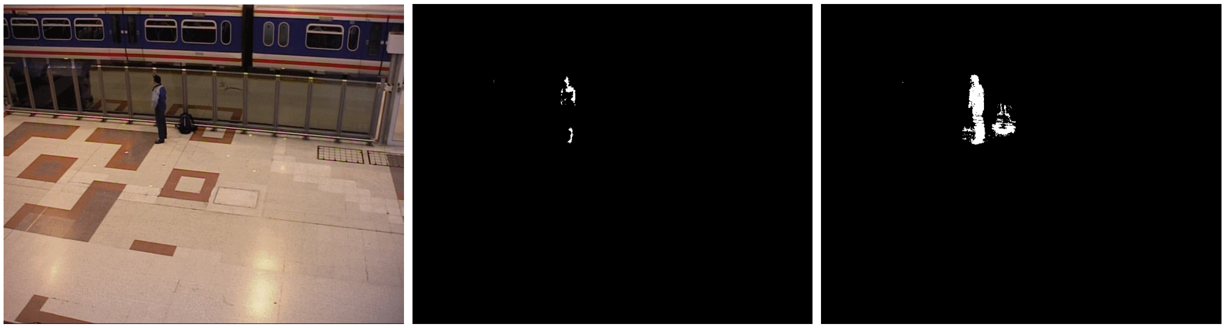

Figure 3. (Left) Sample input image. Foreground masks obtained from the background subtraction algorithm without (Middle) and with context information (Right).

$$T(p_i) = \begin{cases} T(p_i) = T(p_i) + \frac{T_{inc(k)}}{\bar{d}_{min}(p_i)} \text{ if } F(p_i) = 1 \\ T(p_i) = T(p_i) - \frac{T_{dec(k)}}{\bar{d}_{min}(p_i)} \text{ if } F(p_i) = 0. \end{cases} \tag{12}$$

Here, $T_{inc(k)}$ and $T_{dec(k)}$ are not fix parameters, but depend on the states of tracked persons in the scene. Also, the upper and lower bound $T_{lower(k)} < T < T_{upper(k)}$ are state depended. Instead of using dual background models for abandoned object detection, which are constructed by setting different learning rates, the overall processing pipeline uses the introduced CAPBAS.

2.4 Overall System

After introducing the main components of the system, the whole processing pipeline is described. The first step for recognizing a split of non-human, static objects is the detection and tracking of persons in the scene. For every tracked person, the tracking component provides the current state and the most likely motion dynamics. The state of the persons defines a region of interest (ROI) which is uniquely assigned to the corresponding person. Thereby, the size of the ROI is determined by the person location and its scale. In comparison to the unified enclosing bounding box it is slightly increased to capture some uncertainties and small movements. All ROI's are continuous checked if the assigned person is still inside the area and the learning rate of all covered pixels is adapted. In case a person leaves its assigned ROI for a defined time interval and the number of detected foreground pixels inside this region exceeds a threshold an alarm event is triggered. For determining the number of foreground pixels areas covered by other persons are excluded. With the proposed system an abandoned object event follows three rules. Contextual rule: An object is owned and attended by a person who enters the scene until such point that the object is not in physical contact with the person. Spatial rule: An object is unattended when the owner is further away than a defined distance. Temporal rule: If an object has been left unattended by the owner for a specific time period in which the owner has not re-attended to the luggage.

3. EXPERIMENTS

The goal of this section is an evaluation of the proposed system and a discussion of several effects, which can lead to non-optimal results. Evaluation is done on the Pets 2006 dataset,[20] which was designed for testing abandoned object detection algorithms in public spaces and the definition of an abandoned object event, including the contextual, spatial, and temporal rule, is derived from the corresponding PETS 2006 challenge. The PETS 2006 dataset consists of multi-sensor sequences containing several abandoned object scenarios with increasing scene complexity. There are seven different scenarios captured by four cameras from different viewpoints. Since our algorithm is based on a single camera and does not require any 3D information, only one camera is used. This camera view is chosen because the objects appear larger in the videos. Figure 4 shows some exemplary results of detected abandoned object events. The detected abandoned object is highlighted in red. The tracked persons are marked with a bounding box.

Figure 4. (Left) Sample input image. (Right) Example of detected abandoned object. (PETS 2006 dataset).

Due to the fact that an object is owned and attended by a person who enters the scene, it can be expected that a person is at first walking and the initial model probability μ_0^i of the IMM filter is set in favor of the constant velocity model. The diagonal of the transition probability matrix p_{ij} is filled with values close to 1. The process and measurement noise is modeled as white noise. A monitoring of a ROI is initiated when the dynamic model of the IMM filter changes to constant position.

In PETS 2006 scenarios all abandoned events are in conjunction with loitering. Only monitoring persons in a constant position mode, which is associated with loitering and slowly moving, helps to reduce false alarm,s but it is no necessary condition for the system. Overall, the proposed system is able to detect all abandoned

Figure 5. (Left) Sample input image. (Right) Example of detected abandoned object. (i-Lids dataset).

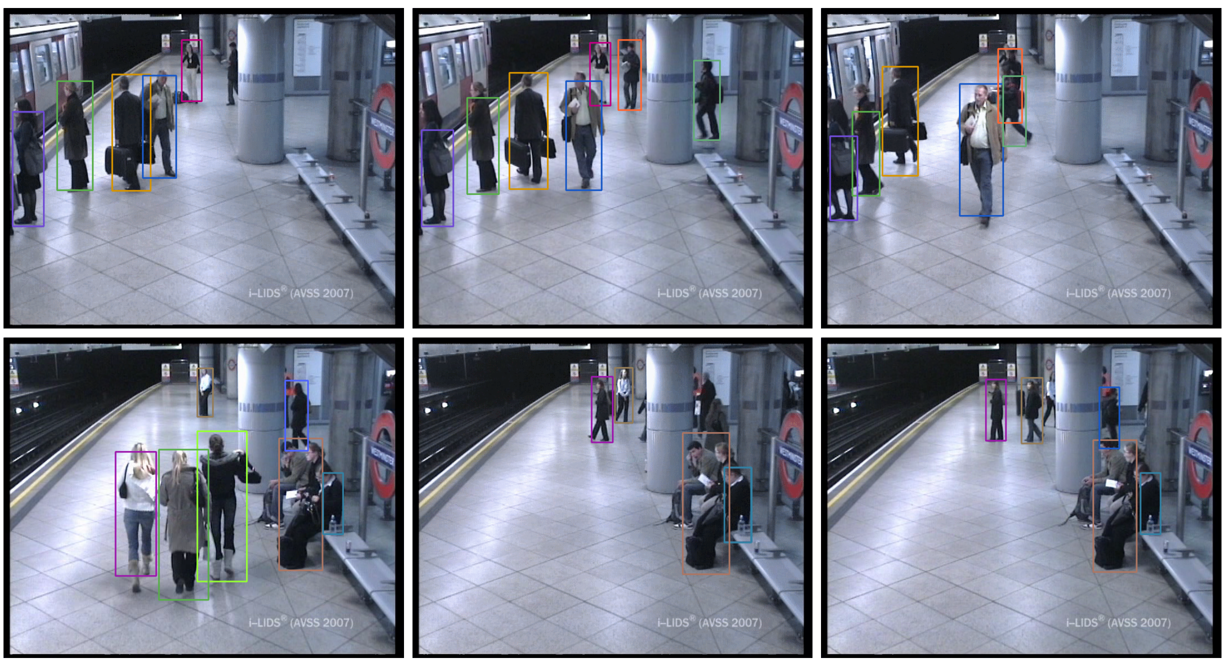

Figure 6. Tracking results on the sequences AVSS_AB_Medium (Top) and AVSS_AB_HARD (Bottom) from the i-Lids dataset. In top sequence the tracking id of the persons carrying the later abandoned object got lost and in the bottom sequence there is an id switch. Thus, the abandoning event cannot correctly be detected.

items in the PETS 2006 challenge. The clear definition of the alarm event enable that a high-level reasoning can eliminate the errors of low-level processing.

However, the aforementioned definition cover only a very limited number of existing real life situations. Moreover, the fact that the system is a combination of two modules which faces open challenges, the system strongly depends on the robustness of these components. Thus strong illumination changes, extremely crowded scenes etc. lead to higher error rates and the limitations of an intermediate tracking step need to be balanced with a possible unique owner assignment for a video forensic system.

Another dataset which includes a definition of abandoning events is the i-Lids dataset.[21] It consist of CCTV footage containing three sequences including an abandoned object event with different difficulty levels according to the luggage size and the density of person groups. This dataset captures also a larger variance of distances of persons to the camera and two actual abandoning object events are with strong occlusion. In such conditions, current dual background or nontracking approaches like[6] are able to introduce a significant improvement. Figure 5 shows a correctly detected abandoned object from the i-Lids dataset, whereas in figure 6 shows two tracking failures. In the top sequence the track of the person carrying the late abandoned object is lost due to occlusion and in the bottom sequence the id (color coded) is changing also due to occlusion. As a result, in both cases the abandoning event is not correctly recognized by the proposed system. Due to an imperfect person tracking the system also produces several false alarms. Although the intermediate tracking has drawbacks, the framework of using an IMM filter estimate for influencing the update probability of a non-parametric background model showed promising results. Future work will focus on an integration in a distributed Cognitive Vision System (dCVS) architecture,[22] where this system will be encapsulated as a self-contained building block.

4. CONCLUSION

In this paper, we presented a video analysis module for abandoned object detection that successfully tackled the challenge proposed by the PETS 2006 workshop. The proposed system is a combination of a person tracking component with an IMM filter and a nonparametric background model with dynamic per-pixel state variables. The state estimates of the tracking component are used as context information for controlling the update probabilities of the introduced CAPBAS. The IMM filter estimates a loitering-person-state according to a finite-state, discrete-time Markov chain. Furthermore, the position of a loitering person defines a region of interest, in which abandoned objects are identified based on the CAPBAS output. Real life situations usually are more complicated than the one presented here, in terms of the number of involved persons and objects and the variation in execution style. In future work, in order to handle such complexities more sophisticated algorithms need to be considered.

Acknowledgment

This work is supported in part by the European Unions Seventh Framework Programme for research, technological development and demonstration under grant agreement FP7-SEC-2012-1-313034 (SAWSOC-Project).

REFERENCES

1. Beynon, M., Van Hook, D., Seibert, M., Peacock, A., and Dudgeon, D., "Detecting abandoned packages in a multi-camera video surveillance system," in [*International Conference on Advanced Video and Signal based Surveillance (AVSS)*], 221–228 (July 2003).
2. Bird, N., Atev, S., Caramelli, N., Martin, R., Masoud, O., and Papanikolopoulos, N., "Real time, online detection of abandoned objects in public areas," in [*International Conference on Robotics and Automation (ICRA)*], 3775–3780 (May 2006).
3. Ferrando, S., Gera, G., and Regazzoni, C., "Classification of unattended and stolen objects in video-surveillance system," in [*International Conference on Advanced Video and Signal based Surveillance (AVSS)*], 21–21 (Nov 2006).
4. Lu, S., Zhang, J., and Feng, D. D., "Detecting unattended packages through human activity recognition and object association," *Pattern Recognition* **40**(8), 2173 – 2184 (2007). Part Special Issue on Visual Information Processing.
5. Porikli, F., Ivanov, Y., and Haga, T., "Robust abandoned object detection using dual foregrounds," *EURASIP J. Adv. Signal Process* (2008).
6. Lin, K., Chen, S.-C., Chen, C.-S., Lin, D.-T., and Hung, Y.-P., "Left-luggage detection from finite-state-machine analysis in static-camera videos," in [*Pattern Recognition (ICPR), 2014 22nd International Conference on*], 4600–4605 (Aug 2014).
7. Tian, Y., Feris, R., Liu, H., Hampapur, A., and Sun, M.-T., "Robust detection of abandoned and removed objects in complex surveillance videos," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **41**(5), 565–576 (2011).
8. Benenson, R., Mathias, M., Tuytelaars, T., and Van Gool, L., "Seeking the strongest rigid detector," in [*CVPR*], (2013).

9. Kieritz, H., Hübner, W., and Arens, M., "Learning transmodal person detectors from single spectral training sets," in [*Proc. SPIE 8901A Optics and Photonics for Counterterrorism, Crime Fighting and Defence IX*], (2013).
10. Dollár, P., Wojek, C., Schiele, B., and Perona, P., "Pedestrian detection: An evaluation of the state of the art," *Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **34**(4) (2012).
11. Enzweiler, M. and Gavrila, D. M., "Monocular pedestrian detection: Survey and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2009).
12. Bar-Shalom, Y., Kirubarajan, T., and Li, X.-R., [*Estimation with Applications to Tracking and Navigation*], John Wiley & Sons, Inc., New York, NY, USA (2002).
13. Li, X. and Jilkov, V., "Survey of maneuvering target tracking. part v. multiple-model methods," *Aerospace and Electronic Systems, IEEE Transactions on* **41**, 1255–1321 (Oct 2005).
14. Kalman, R. E., "A new approach to linear filtering and prediction problems," *ASME Journal of Basic Engineering* (1960).
15. Hartikainen, J. and Särkkä, S., *Optimal filtering with Kalman filters and smoothers a Manual for Matlab toolbox EKF/UKF* (2008).
16. Särkkä, S., *Recursive Bayesian inference on stochastic differential equations*, PhD thesis, University of Technology, Helsinki (2006).
17. Munkres, J., "Algorithms for the assignment and transportation problems," *Journal of the Society of Industrial and Applied Mathematics* **5**(1), 32–38 (1957).
18. Hofmann, M., Tiefenbacher, P., and Rigoll, G., "Background segmentation with feedback: The pixel-based adaptive segmenter," in [*Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*], (2012).
19. Barnich, O. and Van Droogenbroeck, M., "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing* **20**(6), 1709–1724 (2011).
20. PETS, "Benchmark dataset," in [*International Workshop on* P*erformance* E*valuation of* T*racking and* S*urveillance*], (2006). (see `http://www.cvg.rdg.ac.uk/PETS2006/index.html`).
21. iLids, "i-Lids bag and vehicle detection challenge," in [*Advanced Video and Signal based Surveillance (AVSS)*], (2007). (see `http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html`).
22. Münch, D., Grosselfinger, A.-K., Kieritz, H., Hübner, W., and Arens, M., "Architecture for and evaluation of situational analysis in the real world," in [*Proc. of the 9th Future Security Research Conference*], (2014).