Objective Evaluation of Pedestrian and Vehicle Tracking on the CLEAR Surveillance Dataset

Murtaza Taj, Emilio Maggio, and Andrea Cavallaro

Queen Mary, University of London
Mile End Road, London E1 4NS (United Kingdom)
{murtaza.taj,emilio.maggio,andrea.cavallaro}@elec.qmul.ac.uk
http://www.elec.qmul.ac.uk/staffinfo/andrea/

Abstract. Video object detection and tracking in surveillance scenarios is a difficult task due to several challenges caused by environmental variations, scene dynamics and noise introduced by the CCTV camera itself. In this paper, we analyse the performance of an object detector and tracker based on background subtraction followed by a graph matching procedure for data association. The analysis is performed based on the CLEAR dataset. In particular, we discuss a set of solutions to improve the robustness of the detector in case of various types of natural light changes, sensor noise, missed detection and merged objects. The proposed solutions and various parameter settings are analysed and compared based on 1 hour 21 minutes of CCTV surveillance footage and its associated ground truth and the CLEAR evaluation metrics.

1 Introduction

People and vehicle tracking in surveillance scenarios is an important requirement for many applications like traffic analysis, behaviour monitoring and event detection. The tracking task is usually performed in two steps: first objects of interest (targets) are detected in each frame of the sequence, next the detections are linked from frame to frame in order to obtain the track of each targets.

In real-world surveillance scenarios the biggest challenges are due to sensor noise, inter-target occlusions and natural environmental changes in the scene. The environmental changes are usually caused by global illumination variations due to the night-and-day cycle, passage of clouds, cast and self shadows, vehicle headlights and street lamps. Also, movement of vegetation due to wind, rain and snow fall can have a major impact on the reliability of an object detector.

1.1 Object Detection

Object detectors can be divided into two main classes, namely *background model* based and *object model* based. In the first class the detection is performed by learning a model of the background and then by classifying as objects of interest connected image regions (blobs) that do not fit the model [1,2,3]. This solution is mainly used to detect moving objects in the scene. In the second class the

R. Stiefelhagen et al. (Eds.): CLEAR 2007 and RT 2007, LNCS 4625, pp. 160–173, 2008.

detector learns a model of the objects of interest and then the model is used by a classifier that is generally applied to each frame of the sequence [4]. Although this approach is also appropriate in applications with non-static cameras, it can only detect object classes belonging to the training dataset.

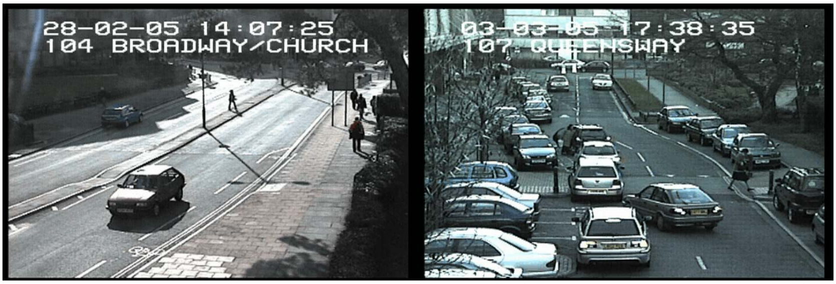
A popular *background model* based adaptive method uses Gaussian Mixture Models (GMM) [5,6,7]. The distribution of a colour of each pixel is approximated by a Gaussian mixture where the parameters are updated to cope with slow changes in natural light conditions. However, when an object becomes static it is gradually assimilated into the background model. The update speed for the parametric model is usually a trade-off between a fast update required to cope with sudden illumination changes and a slow update necessary to allow the detection of slow or stopping objects. A possible solution is to modify the learning rate in the region around a moving object depending on its speed [5]. Also, edge information can help detecting objects when they become static [6]. Once the edge structure of the background is learned, a pixel is classified as foreground by comparing its gradient vector with the gradient distribution of the background model.

A major problem with background-based detection algorithms is the difficulty to deal with object interactions, such as object proximity and occlusions. In such a case, two objects that are close are likely to generate a merged foreground region that produces one detection only, instead of multiple detections. However, when an occlusion is partial, projection histograms can be used to split the merged objects [5]. Also, motion prediction based on trajectory data can help to estimate the likelihood of an occlusion thus allowing a single blob to represent two objects [8].

Unlike background model based methods, *object model* based techniques [4,9] learn local representative features of the object appearance and perform detection by searching for similar features in each frame. Edgelets [10] or Haar wavelets [11] are used in Adaboost algorithms as weak object classifiers that combined in a cascade form a strong classifier [12]. Approaches based on learned classifiers are also used after background subtraction to categorize the detections (i.e. to differentiate pedestrians from vehicles) [13]. Similarly, Support Vector Machines using simple object features, such as object size and object width-height ratio, can be used [8].

1.2 Object Tracking

Once object detection is performed, data association is needed to link different instances of the same object over time. Generating trajectories requires an estimate of the number of targets and of their position in the scene. As modelling of all the possible object interactions is (in theory) necessary, the tracking problem has a complexity that is exponential with the number of targets in the scene. Joint probabilistic data association filter (JPDA) [14] is a widely used data association technique. An alternative is to model the problem with a graph [15]



(a) (b)

Fig. 1. CLEAR dataset scenarios for the pedestrian and vehicle tracking task. (a) Scenario 1: Broadway Church (BC). (b) Scenario 2: Queensway (QW).

where the nodes are associated to the detections and the edges represent the likelihood that two detections in consecutive frames are generated by the same object. Smoothing or target state estimation can be performed by initializing a Kalman Filter for each target [5] and by assuming Gaussianity of the posterior density at every time step. This limiting assumption can be alleviated by using Particle Filters [7]. An alternative to probabilistic methods is Mean Shift (MS), a non-parametric kernel-based method used for target localization [8]. Smoothing and clutter filtering can also be performed prior to data association using a Probability Hypothesis Density (PHD) filter [16], a Bayesian recursive method with linear complexity (with the number of targets). The PHD filter approximates the multi-target statistics by propagating only the first order moments of the posterior probability.

1.3 Detection and Tracking Algorithm Under Evaluation

The detection and tracking algorithm we evaluate in this paper [2] performs object detection using a statistical background model [3] and data association using graph matching [15]. Because performance varies depending on different environmental conditions, the testing and evaluation of a detection and tracking algorithm requires a large amount of (annotated) data from real word scenarios. The CLEAR dataset provides a large testbed making it easier to evaluate how different features impact on the final detection and tracking results.

In this paper we analyse the results of a total of 13 runs on the complete CLEAR dataset consisting of 50 sequences with ground truth annotation, for a total of 121.354 frames per run (i.e., approximately 1 hour 21 minutes of recorded video). To reduce the computational time we processed the sequences at a half the original resolution (i.e., 360x240 pixels). The dataset consists of outdoor surveillance sequences of urban areas (Fig. 1) and the annotation provides the bounding boxes of pedestrians and vehicles in the scene. The complexity of the

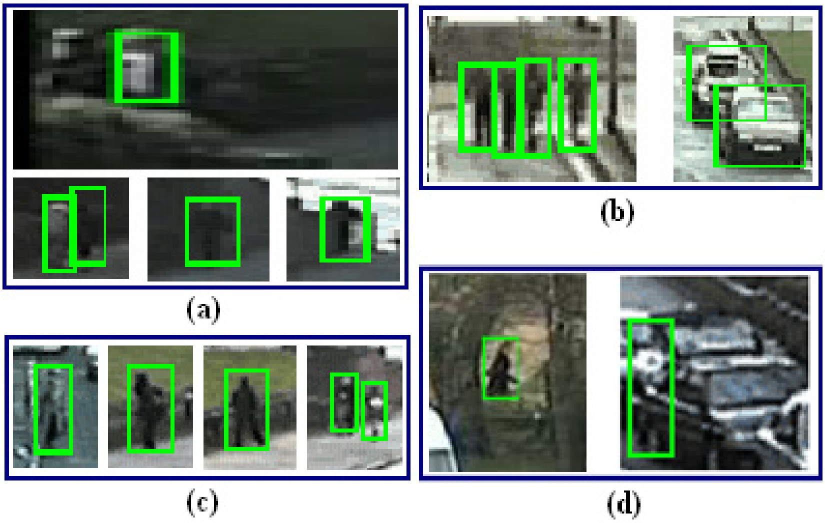


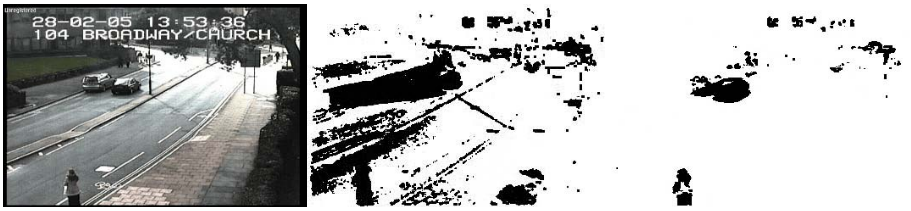
Fig. 2. Examples of challenging situations for the pedestrian and vehicle detection and tracking task in the CLEAR dataset (the ground-truth detection are shown in green). (a) Objects in low visibility regions. (b) Objects in close proximity. (c) Objects with low contrast compared to the background. (d) Occluded objects.

CLEAR dataset is related to the challenges discussed earlier in this section. A set of samples illustrating these difficult situations is shown in Fig. 2: objects with low visibility located in the shade generated by a building (Fig. 2 (a)); merged detections (Fig. 2 (b)) due to either the physical closeness or to the camera perspective view; objects with little contrast compared to the background (Fig. 2 (c)); partial and total occlusions (Fig. 2 (d)).

The detection and tracking performance is measured by means of a set of scores (i.e., Multi-Object Detection Precision (MODP), Detection Accuracy (MODA), Tracking Precision (MOTP) and Tracking Accuracy (MOTA)) defined by the CLEAR evaluation protocol [17]. These scores give a weighted summary of the detection and tracking performance in terms of False Positives (FP), False Negatives (FN) and object identity switches.

1.4 Organization of the Paper

This paper is organized as follows. Section 2 discusses the improvements in the detection algorithm under natural environmental changes (using background model update and edge analysis), illumination flickering (using spatio-temporal filtering), sensor noise (using noise modelling), miss-detections and clutter (using the PHD filter), and merged objects (using projection histograms). Finally, in Section 3 we discuss the results and we draw the conclusions.



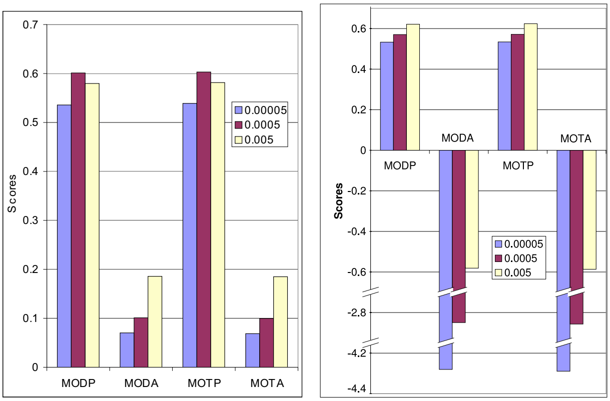
(a) (b) (c)

Fig. 3. Comparison of background subtraction results with and without update of the background model. (a) Original scene, (b) sample result without background update and (c) sample result with background update.

2 Performance Evaluation

2.1 Natural Environmental Changes

As described in Section 1, rapidly changing illumination conditions and inappropriate background modelling can lead to a situation where most of the pixels are classified as foreground pixels (Fig. 3). For background modelling, we use a linear update strategy with a fixed update factor. At time t the background model $I_{t-1}^{(bk)}$ is updated as $I_t^{(bk)} = \alpha I_t + (1-\alpha) I_{t-1}^{(bk)}$, where I_{t-1} is the previous frame and α is the update factor. The choice of α depends on a trade-off between update capabilities and resilience to merging stopped or slow foreground objects in the background model. Figure 4 shows the performance comparison varying α in the range $[0.00005, 0.005]$. Increasing α from $\alpha = 0.00005$ to $\alpha = 0.0005$ precision and accuracy improve (Fig. 4); FPs are reduced without a significative



(a) (b)

Fig. 4. Comparison of tracking results with different update factors for the background model. (a) Pedestrian tracking. (b) Vehicle tracking.

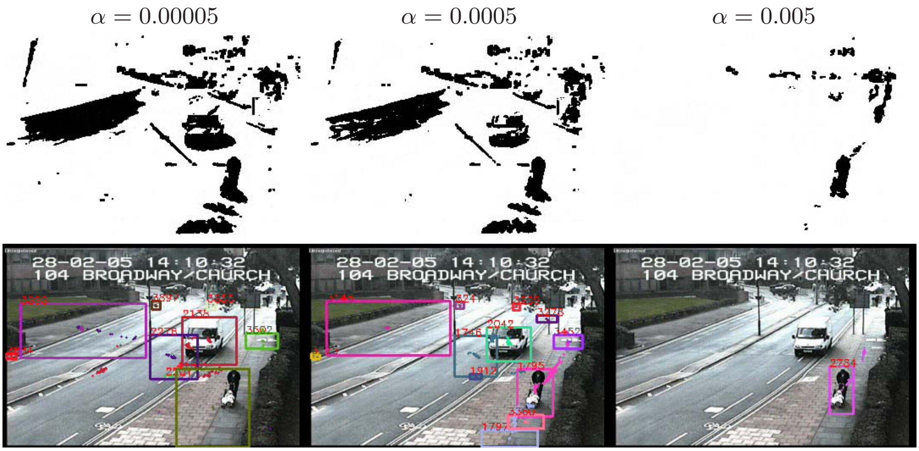

$\alpha = 0.00005$ $\alpha = 0.0005$ $\alpha = 0.005$

Fig. 5. Sample tracking results with different update factors for the background model. A reduction of False Positives is observed by increasing α (from left to right). When $\alpha = 0.005$ no False Positives are returned at the cost of one False Negative.

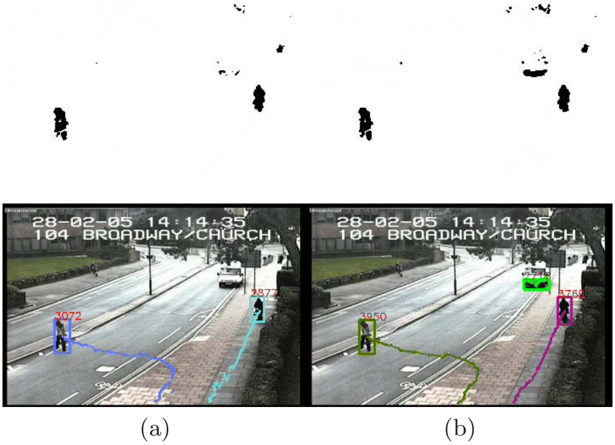

(a) (b)

Fig. 6. Sample tracking results with and without change detection enhanced by edge analysis (EA). (a) Without edge analysis, (b) with edge analysis.

increase of FNs. However, when increasing α to 0.005, the accuracy improves but the precision decreases for pedestrian tracking. Figure 5 shows how the model update manages to reduce FPs; however, the car that stopped on the road becomes part of the background model thus producing a FN. A value of $\alpha = 0.005$ is therefore a good compromise between accuracy and precision.

To avoid erroneously including slow moving objects into the background, we use Edge Analysis (EA). EA enhances the difference image obtained after background subtraction using an edge detector. In our implementation we compute

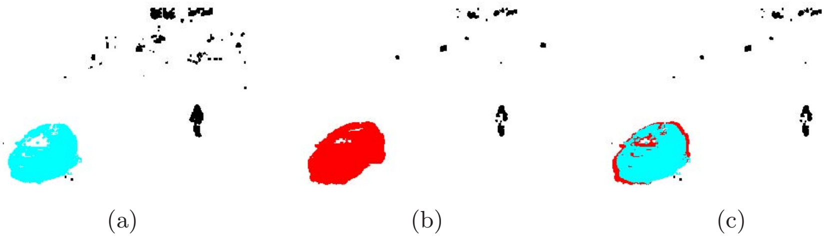


Fig. 7. Comparison of background subtraction results obtained with and without edge analysis. (a) Without edge analysis the results contain a large number of spurious blobs. (b) With edge analysis the spurious blobs are partially removed, however this generates holes in the pedestrian and an enlarged mask for the vehicle. (c) Superimposed result showing the extra pixels (halo) around the vehicle.

the edges by taking the difference between consecutive frames. Figure 6 shows an example of a correct detection of a vehicle despite it had stopped. The price to pay for these correct detections is an artificial enlargement of the blobs produced by fast moving objects (Fig. 7).

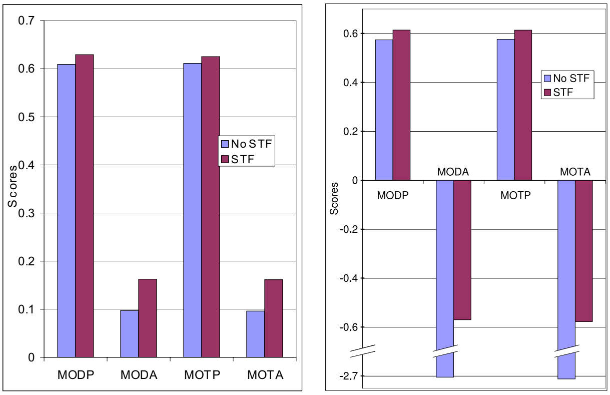
2.2 Flickering Illumination

To reduce the effect on the object detector of short-term illumination variations we use a spatio-temporal filtering (STF) on the result of the frame difference. An n-frame window is used to smooth the output using past and future information. Figure 8 shows the comparative results using pixel-wise temporal filtering. The improvements in terms of accuracy and precision are of 58% and 4%, respectively, for vehicle detection and of 2% and 7%, respectively, for pedestrian tracking.

2.3 Sensor Noise

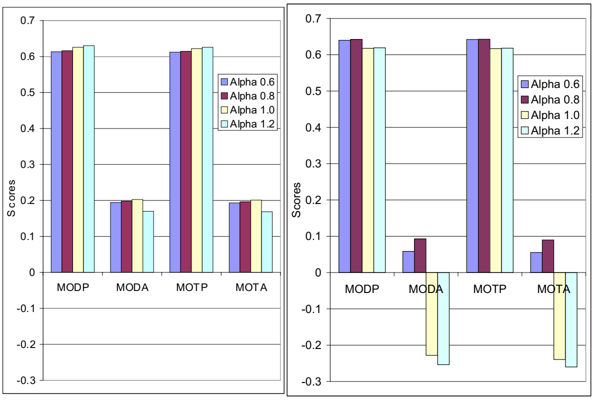
The video acquisition process introduces noise components due to the CCTV cameras themselves. To reduce the effect of the sensor noise, the simplest solution is to threshold the frame difference, either using luminance information only or using the three colour channels. The problem with using a fixed threshold is the inability of the algorithm to adapt to different illumination conditions and therefore is not appropriate for long sequences or across different sequences, as manual tuning is necessary. An alternative is to model the noise assuming that its distribution is Gaussian [3,18] or Laplacian [19].

In this work we performed object detection assuming additive white Gaussian noise on each frame and using a spatial observation window [3]. To account for camera perspective and to preserve small blobs associated to objects in regions far from the camera (top of the frame), unlike our previous work [2], we learn or adapt σ according to the spatial location [1]. We divide the image into three horizontal regions and apply three different multipliers to σ, namely 0.75, 1 and



(a) (b)

Fig. 8. Comparison of tracking results with and without spatio-temporal filtering (STF). (a) Pedestrian tracking, (b) vehicle tracking. The scores show a significant improvement especially in terms of accuracy.



(a) (b)

Fig. 9. Comparison of tracking results by changing the model parameter of the sensor noise. (a) Pedestrian tracking. (b) Vehicle tracking.

1.25. The amplitude of the noise ($\sigma = 0.8$) was estimated experimentally. Figure 9 shows the impact of σ on vehicle and pedestrian tracking. The value $\sigma = 1.0$ produces better results for pedestrians but also an important performance decrease in terms of accuracy for vehicle tracking. Figure 10 shows sample detection results: the highest value of σ does not allow the detection of the pedestrians, whereas with $\sigma = 0.8$ the classification of most of the pixels belonging to the object is correct.

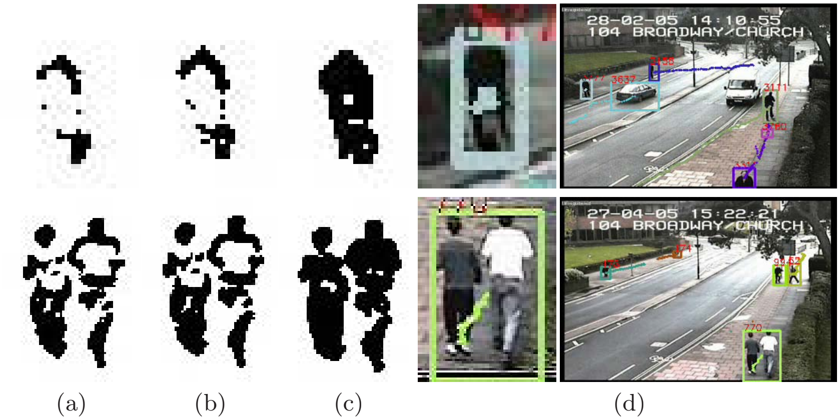


Fig. 10. Comparison of background masks of pedestrians by changing the model of the sensor noise. (a) $\sigma = 1.2$, (b) $\sigma = 1.0$, (c) $\sigma = 0.8$. (d) Sample tracking results.

2.4 Filtering Clutter and Miss-Detections

To mange the intrinsic exponential complexity of the multi-target tracking problem we recently proposed to use the PHD filter, a tracking algorithm that achieves linear complexity by propagating only the first order statistics of the multi-target posterior. This spatio-temporal filter is able to model birth and death of targets, background clutter (i.e., FP), miss detections (i.e., FN) and the spatial noise on the detections. Figure 11 shows the comparative results of introducing the PHD spatio-temporal filtering stage at the detection level. In vehicle tracking, the PHD filter allows for a 6% improvement in accuracy and

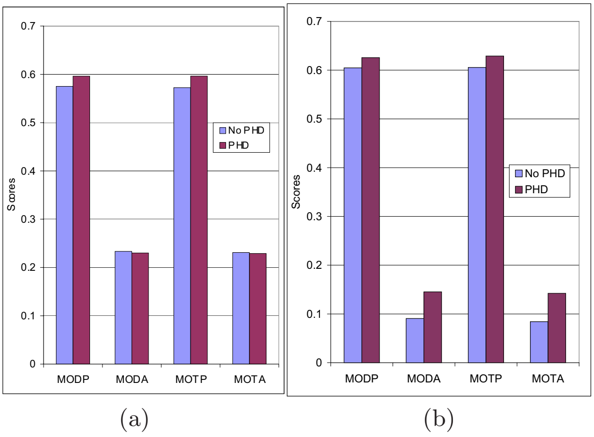
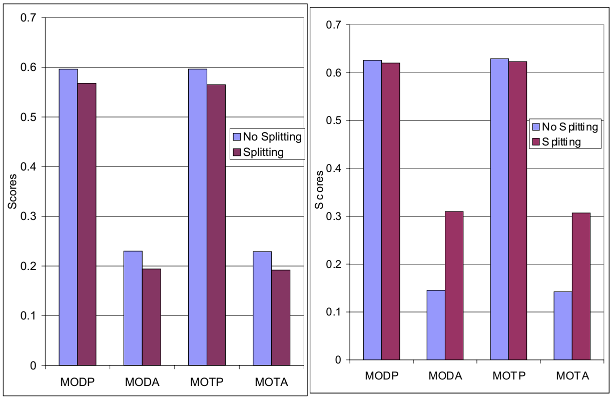


Fig. 11. Comparison of tracking results with and without PHD Filter. (a) Pedestrian tracking: enabling the PHD filter the tracker achieves higher precision scores. (b) Vehicle tracking: the tracker achieves both higher precision and accuracy scores.



(a) (b)

Fig. 12. Comparison of tracking results obtained by splitting the blobs associated to more than one target using projection histograms. (a) Pedestrian tracking: small decrease in the scores. (b) Vehicle tracking: large accuracy improvement.

2% improvement in precision. In pedestrian tracking, there is 2% increase in precision whereas there is no significant change in accuracy.

2.5 Merged Objects

Multiple objects in proximity to each other may be grouped into one blob only by background subtraction based detection algorithms. In order to maintain a separate identity for these objects, a possible solution is to analyse the histograms

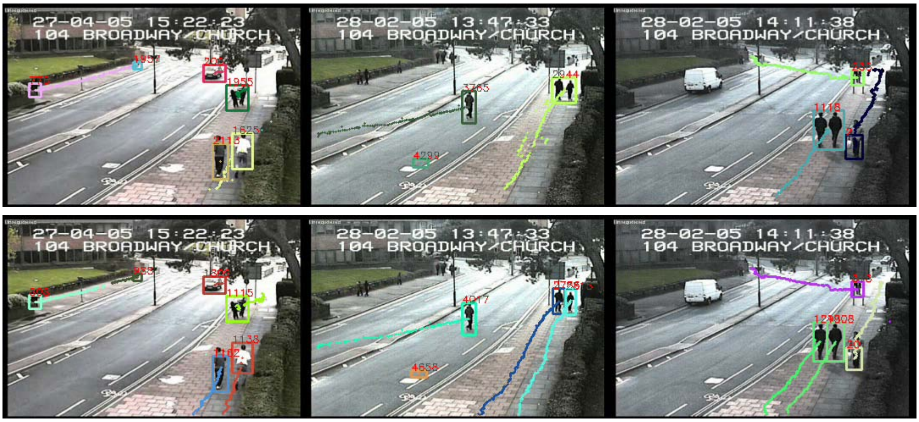


Fig. 13. Sample tracking results obtained with and without blob spitting using projection histograms. Top row: without blob splitting. Bottom row: with blob splitting.

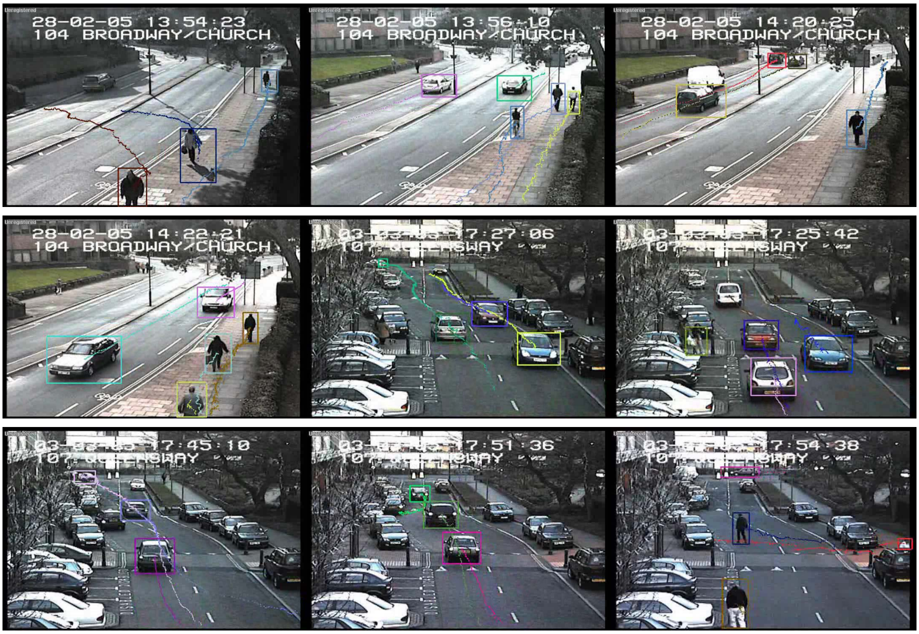


Fig. 14. Sample tracking results on Broadway Church (BC) and Queensway(QW) scenarios

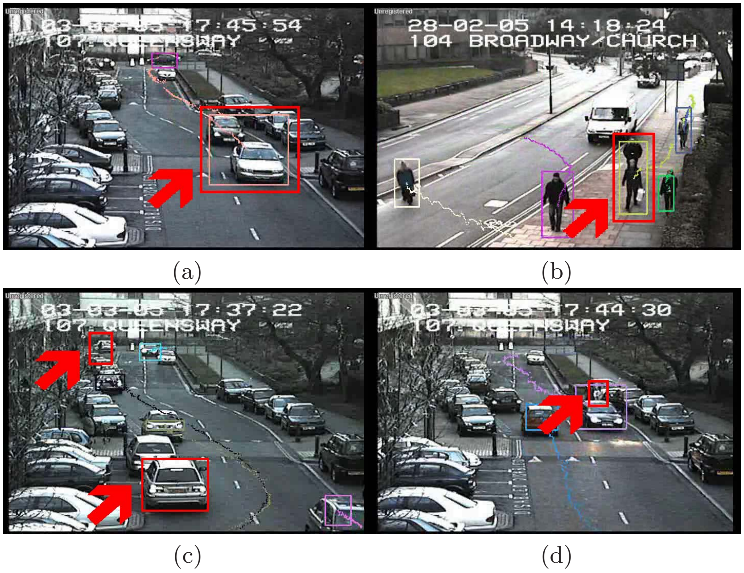
of the pixels of a blob projected on one of the two Cartesian coordinates [20]. This solution assumes that the modes of the histogram correspond to the different pedestrians that can be split by separating the modes.

Figure 12 shows the tracking performance comparison with and without the use of the projection histograms based blob splitting. The impact of this procedure on the scores is biased by the vehicle-pedestrian classification. As the classification depends on the width-height ratio of the bounding boxes, the splitting allows to assign the correct label to group of pedestrians and therefore the accuracy of vehicle tracking increases by 16%. However, previous errors assigned to the vehicle tracking scores are now transferred to pedestrian tracking. An example of tracks obtained with and without splitting is shown in Figure 13. The merged blobs associated to the two pedestrians on the right are constantly split by analysing the projection histograms.

To conclude, Fig. 14 shows sample detection and tracking results generated by the proposed framework on the BC and QW scenarios under different illumination conditions.

2.6 Failure Modes

Figure 15 shows three failure modalities of the proposed tracker. In Fig. 15 (a and b) two objects are merged and the use of the projection histograms based



(a)

(b)

(c)

(d)

Fig. 15. Sample failure modalities on the CLEAR Broadway Church (BC) and Queensway (QW) scenarios (red boxes indicate the areas of the frame where the failure occurred). Top row: merged objects. Bottom left: missed detections caused by small objects. Bottom-right: object incorporated into the background model.

splitting does not help as the objects are not merged along the horizontal axis. A possible solution could be the use of a body part detector to estimate the number of targets in a blob. Figure 15 (c) shows missed detections caused by (i) static, (ii) small and (iii) similar-to-the-background objects. Figure 15(d) shows a failure due to an occlusion. To overcome this problem, information from multiple cameras could help disambiguating the occlusion.

3 Conclusions

In this paper we analysed major challenges of video tracking in real-world surveillance scenarios. Starting from this analysis and a well-tested tracking platform, we evaluated the inclusion of a set of new features into the framework. The main added features are a background model update strategy, a spatio-temporal filtering, edge analysis, a PHD filtering step and splitting blobs containing nearby objects by means of projection histograms. The evaluation was performed on the CLEAR dataset and showed that the new features improve the accuracy and precision of the tracker by 76% and 50%, respectively. Further work includes the improvement of the vehicle-pedestrian classification step by means of a dedicated object classifier.

Acknowledgements

The authors acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC), under grant EP/D033772/1.

References

1. Stauffer, C., Grimson, W.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Machine Intell. 22, 747–757 (2000)
2. Taj, M., Maggio, E., Cavallaro, A.: Multi-feature graph-based object tracking. In: Stiefelhagen, R., Garofolo, J.S. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 190–199. Springer, Heidelberg (2007)
3. Cavallaro, A., Ebrahimi, T.: Interaction between high-level and low-level image analysis for semantic video object extraction. EURASIP Journal on Applied Signal Processing 6, 786–797 (2004)
4. Wu, B., X., Kuman, V., Nevatia, R.: Evaluation of USC Human Tracking System for Surveillance Videos. In: Stiefelhagen, R., Garofolo, J.S. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 183–189. Springer, Heidelberg (2007)
5. Pnevmatikakis, A., Polymenakos, L., Mylonakis, V.: The ait outdoors tracking system for pedestrians and vehicles. In: Stiefelhagen, R., Garofolo, J.S. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 171–182. Springer, Heidelberg (2007)
6. Zhai, Y., Berkowitz, P., Miller, A., Shafique, K., Vartak, A., White, B., Shah, M.: Multiple vehicle tracking in surveillance video. In: Stiefelhagen, R., Garofolo, J.S. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 200–208. Springer, Heidelberg (2007)
7. Abd-Almageed, W., Davis, L.: Robust appearance modeling for pedestrian and vehicle tracking. In: Stiefelhagen, R., Garofolo, J.S. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 209–215. Springer, Heidelberg (2007)
8. Song, X., Nevatia, R.: Robust vehicle blob tracking with split/merge handling. In: Stiefelhagen, R., Garofolo, J.S. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 216–222. Springer, Heidelberg (2007)
9. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, Hawaii, pp. 511–518 (2001)
10. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: Proc. of IEEE Int. Conf. on Computer Vision, pp. 90–97. IEEE Computer Society Press, Washington (2005)
11. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: Proc. of Int. Conf. on Computer Vision Systems, vol. 2, pp. 734–741 (2003)
12. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Technical report, Department of Statistics, Stanford University (1998)
13. Munder, S., Gavrila, D.: An experimental study on pedestrian classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(11), 1863–1868 (2006)
14. Herman, S.: A Particle Filtering Approach to Joint Passive Radar Tracking and Target Classification. PhD thesis, University of Illinois at Urbana Champaign (2005)

15. Shafique, K., Shah, M.: A noniterative greedy algorithm for multiframe point correspondence. IEEE Trans. Pattern Anal. Machine Intell. 27, 51–65 (2005)
16. Maggio, E., Piccardo, E., Regazzoni, C., Cavallaro, A.: Particle phd filter for multitarget visual tracking. In: Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Honolulu, USA (2007)
17. Kasturi, R.: Performance evaluation protocol for face, person and vehicle detection & tracking in video analysis and content extraction (VACE-II). Computer Science & Engineering University of South Florida, Tampa (2006)
18. Li, W., Unbehauen, J.L.R.: Wavelet based nonlinear image enhancement for gaussian and uniform noise. In: Proc. of IEEE Int. Conf. on Image Processing, Chicago, Illinois, USA, vol. 1, pp. 550–554 (1998)
19. Aiazzi, B., Baronti, S., Alparone, L.: Multiresolution adaptive filtering of signal-dependent noise based on a generalized laplacian pyramid. In: Proc. of IEEE Int. Conf. on Image Processing, Washington, DC, USA, vol. 1, pp. 381–384 (1997)
20. Hu, W., Hu, M., Zhou, X., Lou, J.: Principal axis-based correspondence between multiple cameras for people tracking. IEEE Trans. Pattern Anal. Machine Intell. 28(4), 663 (2006)