Real-time ghost removal for foreground segmentation methods

Fei Yin, Dimitrios Makris, Sergio Velastin

Real-time ghost removal for foreground segmentation methods

Fei Yin, Dimitrios Makris, Sergio A Velastin
Digital Imaging Research Centre (DIRC)
Faculty of Computing, Information Systems and Mathematics
Kingston University, Surrey, KT1 2EE, UK
{fei.yin, d.makris, sergio.velastin}@kingston.ac.uk

Abstract

Among all the challenges for foreground detection, we focus on detection of ghosts caused by the starting or stopping of objects. We propose a fast and effective method for ghost detection that compares the similarity between the edges of the detected foreground objects and those of the current frame based on object-level knowledge of moving objects. Finally we demonstrate the performance of ghost detection algorithm using a series of urban traffic video sequences including car parking and departure which normally cause ghost problems. The performance evaluation results show that the proposed method can detect and reduce ghost objects efficiently, brings little computational load, and no significant side-effect to the surveillance system.

1. Introduction

Visual surveillance in dynamic scenes, especially for tracking humans and vehicles moving through the field of view (FOV) of a CCTV camera has received much attention in the past decades [1]. Because detection of moving objects is a key preliminary, it ultimately limits successful tracking in surveillance applications such as traffic monitoring and analysis, access control in special areas, human and vehicle identification, and detection of anomalous behaviours.

The most common approach for detecting moving objects is background subtraction, in which each frame is compared against a background model. The pixels in the current frame that have significant difference from the background are then considered as moving pixels or foreground pixels. These foreground pixels will be grouped to form objects which are then tracked. A large number of background subtraction algorithms have been proposed so far [1], but problems still remain for moving objects identification under certain conditions.

One of the toughest problems is that background subtraction causes the detection of false objects when an object that belongs to the background starts to move away, often referred to as “ghost”. It is important to address the problem because ghost objects will adversely affect many tasks such as object classification, tracking and event analysis (e.g. abandoned item detection). Therefore, ghost object needs to be separated from other objects and eliminated.

This paper focuses on the problem of ghost identification and elimination. We used a state-of-the-art industrial tracker [6] which includes basic background subtraction and object tracking. Then we included our ghost detection algorithm into the basic tracker to identify and eliminate ghosts. Finally, we systematically evaluate and compare performance on some urban traffic video sequences. The ground truth for all videos was manually generated using Viper GT [8].

This paper is organized as follows: Section 2 defines the problem of ghost detection and surveys some previous methods in the literature. Section 3 describes the methodology of ghost detection. Results are presented and discussed in section 4. Finally section 5 concludes the paper.

2. Background

Generally, a robust background model should be able to automatically recover and update itself from a dynamic sequence and be insensitive to illumination changes, shadows, weather conditions such as rain or snow. One of the most popular existing background modelling algorithms was proposed by Stauffer and Grimson [2] in 1998 and further refined e.g. in [9].

Besides background modelling, the accuracy of motion detection also depends on rules of motion segmentation and object classification. Normally, motion segmentation methods are based on the notion that differences in consecutive images arise from moving objects. The motion is measured using either image differencing or flow. There are some drawbacks of using a background model. One problem related to our work is that, to account for changing illumination conditions, objects that become stationary are relatively quickly incorporated into the background and when they move again they leave behind an area of foreground (or "ghost") that can be mistakenly taken as a new object. So the ghost problem is a direct result of the approach to background modelling. An approach that would feedback high level object detection/tracking information to the background model might be able to reduce these effects, for example by enforcing constraints such as that slow or stopped objects cannot simply disappear (merged into the background).

Cucchiara et al [3] proposed a general frame work of background subtraction called Sakbot which combines statistical assumptions to detect moving objects, apparent objects (ghosts), and shadows with the object level knowledge of those from previous frames. For ghost detection, they calculate the average optical flow, over all the detected moving pixels. They assumed that moving objects should have significant motion, while ghosts should have a near-to-zero average optical flow since their motion is only apparent. However, optical flow is computational expensive for real time processing and also there is a danger of inaccurate classification of stationary object into ghost.

Cheung et al [4], proposed a different method to detect ghost in traffic surveillance. They used a frame difference mask which was computed using the incoming frame and its previous frame. Objects with no corresponding blob in the frame difference mask were identified as ghosts. However, the method can not distinguish between ghosts and abandoned objects.

In Guler's method [5], a specific processing layer is used for detection of objects that become stationary. The detected stationary foreground objects are maintained in the specific layer, and the motion history of the immobile objects are recorded. However, the method needs motion history of the object which is not available if the object belonged to background from the start of the video sequence.

Lu et al [10] proposed to recover the "real" background by using a revised image inpainting method. Thus they could differentiate between ghost and abandoned objects: The object will be declared as ghost if there is no difference between an incoming frame and the "real" background image, otherwise it is an abandoned item. However, image inpainting is computational expensive for real time processing, and their method has not been evaluated quantitatively.

Desurmont et al [12] proposed a general framework for video surveillance systems and calculated the edge mean square gradient (EMSG) for each blob. They find out ghost blobs based on the assumption that in the background the gradient is typically low. Therefore, if the EMSG for a blob is very small (lower than a threshold), the blob will be classified as ghost, otherwise it will be classified as real object. However, this is not always true, e.g. a cluttered scene may result in a high textured background with high EMSG.

In this paper, we propose using edge comparison between extracted regions from multiple images to find out whether the detected foreground belongs to the current image or not, and furthermore to identify detected foreground objects as real objects (whether moving or stationary) or ghosts. The method is discussed in detail in section 3 and evaluated in section 4.

3. Methodology

We have developed a framework that we use to detect ghosts during tracking and furthermore to eliminate them.

Ghosts mainly appear in two cases: In the first case, when a moving object becomes stationary, it will be adapted (merged) into the background, and then, when it starts to move again some time later, there will be a ghost left behind. In the second case, an existing object that belongs to the background starts to move (e.g parked vehicle) and will also cause a ghost problem. In the first case, we may have the history of object trajectory, but in the second case, we may not have any previous information about the object. So it is preferable to make the tracker able to recognise ghosts in time without additional history information rather than analyse the history of the object.

We consider that a basic tracker is available with the following assumptions:

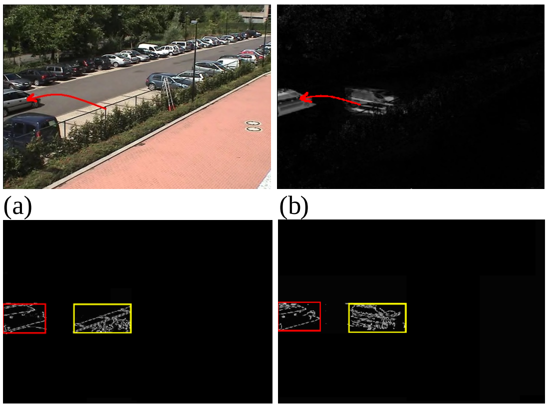
- Tracking is performed at pixel level and no calibration information is needed.
- Tracking is performed on gray-level image sequences (although extension of the method to colour image sequences is relatively simple).
- The basic tracker contains a motion detection module that can provide a difference map which

is the difference of the values of each pixel between the background image and the incoming frame

- The basic tracker is able to estimate the bounding box and the velocity of moving objects at each frame.

Our proposal of ghost detection is described as follows:

- For every k frames (k=12 in our implementation), we check the speed of each track. If the track's speed is below a threshold T_v (e.g. T_v= 0.1 pixel/frame), we proceed further to the ghost detection.
- We apply Gaussian smoothing and Canny edge detection in regions of the bounding boxes (bbox) on both the difference map and the incoming frame. and we get two edge maps for every track at the same time (see Figure 1).



(a) (b) (c) (d)

Figure 1 KND2-JULY-EAST-S2.avi sequence frame 5880
a) a car starts to move out of its parked area along the red arrow
b) the detected moving area by background subtraction c) canny edge detection of "(a)" in track regions d) canny edge detection of "(b)" in track regions Note that red and yellow boxes indicate the regions of two different tracks

- Then, we compare the difference between the edges from the two maps (incoming frame and the frame difference map) in the area of the bounding boxes (bbox) for each track respectively. (e.g. in figure 1 we compare the right yellow bbox with the left yellow bbox since they are of the same track). We get a score for edge similarity (S) between the edges for each track in each frame, and then set up a threshold for making the decision of whether they are the same or different (usually, the score is from 0% (do not match at all) to 100% (perfectly matched).

Suppose for track i, frame j, we get a set of edge points from the difference map (see Figure 2):

$$D_{i,j} = \{(x_1^D, y_1^D), (x_2^D, y_2^D) \ldots (x_n^D, y_n^D)\} \quad (1)$$

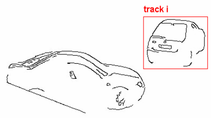


Figure 2 edges of difference map

and another set of edge points from the current image for track i (see Figure 3):

$$C_{i,j} = \{(x_1^C, y_1^C), (x_2^C, y_2^C) \ldots (x_m^C, y_m^C)\} \quad (2)$$

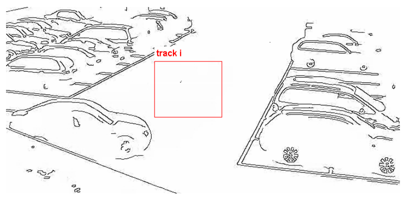


Figure 3 edges of current image

We calculate the score of S_{ij} between two edge sets as follows:

$$S_{i,j} = \frac{size(D_{i,j} \cap C_{i,j})}{size(D_{i,j} \cup C_{i,j})} \quad S_{i,j} \in [0 \sim 1] \quad (3)$$

- Normally, similarity scores are very high when the tracks are from real existing objects (typically much higher than 30%). If $S_{i,j}$ is larger than a threshold T (set to 30% according to experiments), a ghost condition has been detected for track ***i***.
- We accumulate evidence over time using a counter G_i which indicates how many times the track i has been detected as ghost. The initial value for G_i is zero.

If $S_{i,j} > T, \quad G_i \leftarrow G_i + 1$ (4)

As long as $G_i > 0$, we label the track as "ghost", the track is prevented from being "valid" or "real moving object" and does not trigger a false alarm. After several times of ghost suspicion (using the criterion of equation 4), if G_i is larger than a predefined number E (set to 5 in our work), then we confirm the track as "ghost" and vanish the track (i.e. feedback the confirmation to the motion detector so that the motion detector enforces the ghost blobs to become part of the background).

- In order to makes sure that we do not detect a "real" object as ghost and vanish it by mistake, we also take the motion of the track into consideration. Let (x_i^O, y_i^O) be the coordinates of the centroid of track i where it appears firstly. $(x_{i,j}, y_{i,j})$ be the coordinates of the centroid of track i in current frame j. $(W_i\ ,H_i\ )$ be the average width and height of track i during its life time. Then,

$$\left.\begin{array}{l} V_i = \dfrac{\left(x_{i,j}-x_i^O\right)^2+\left(y_{i,j}-y_i^O\right)^2}{W_i^2+H_i^2} >= 1 \\ \qquad G_i > 0 \end{array}\right\} \quad G_i \leftarrow G_i - 1 \tag{5}$$

If the track has moved a long enough distance from where it appeared, then it is not be identified as "ghost". The counter G will be decremented to zero gradually.

Our ghost detection algorithm is summarized in Figure 4:

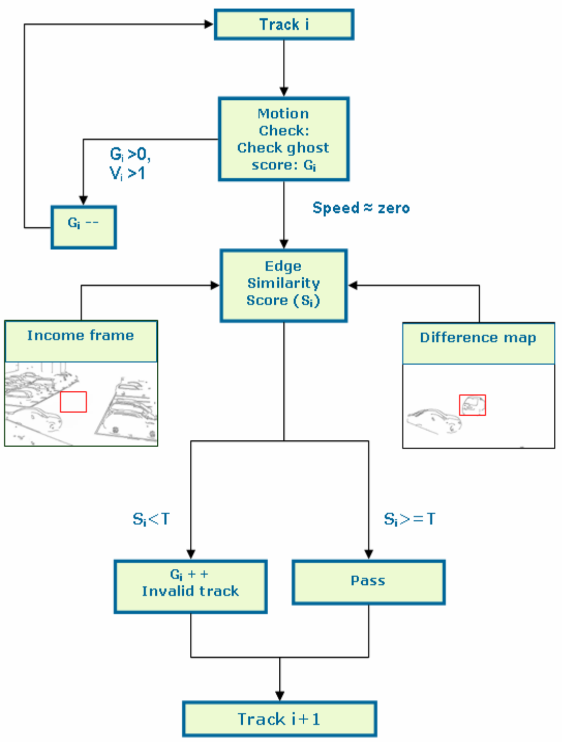


Figure 4 Framework of ghost detection algorithm

4. Results

We demonstrate the practical value of the proposed method by using an experimental industrial tracker from BARCO [11] called the "basic tracker" here and then the same tracker with the addition of our ghost detection algorithm. We run the two trackers on a standard PC with Pentium IV 3.4 GHz processor with 1GB RAM. The tracking and ghost detection were carried out on 4 video sequences with 6 ghost events and 2 abandoned items. We managed to remove all the 6 ghost objects, and avoid all false alarms caused by ghosts. (see Figures 5,6,7)

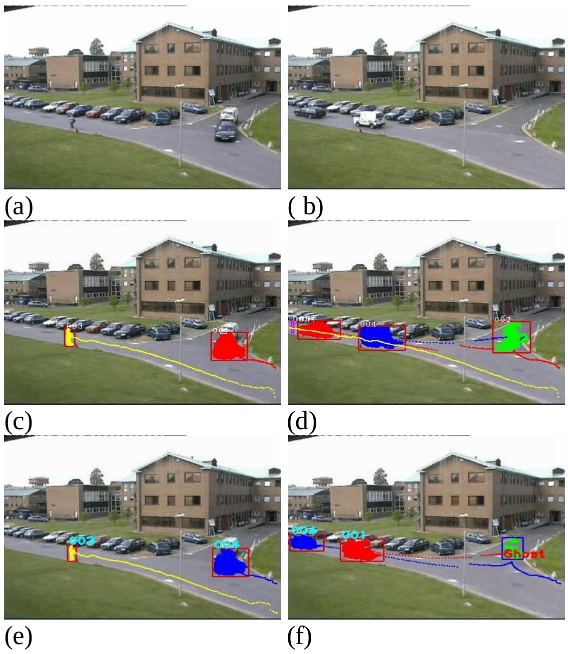
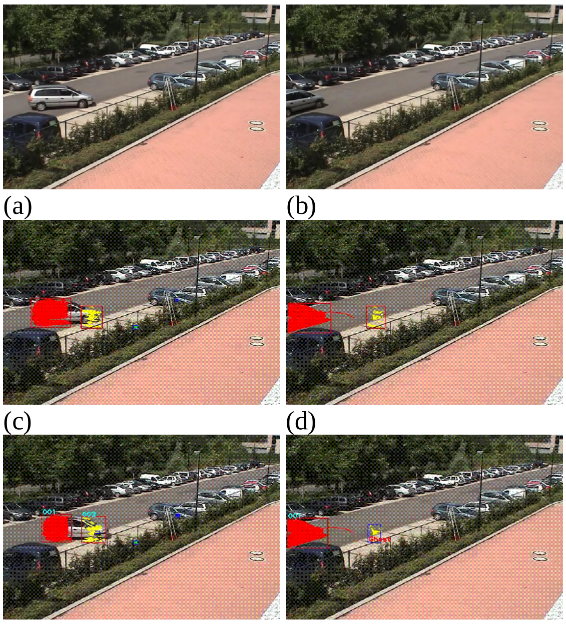


Figure 5 PETS2001 PetsD1TeC1.avi sequence is 2686 frames (00:01:29) long and depicts 4 persons, 2 groups of persons and 3 vehicles. Its main challenge is the multiple object intersections and ghosts. a) frame 2360 b) frame 2635 c) basic tracker at frame 2360 d) basic tracker at frame 2635 e) tracker with ghost detection at frame 2360 f) tracker with ghost detection at frame 2635

Figure 5 presents visual results from the basic motion tracker as well as from the tracker with ghost detection algorithm using a PETS2001 sequence. Red bboxes represent the valid tracks generated by the tracker, while blue bboxes represent invalid tracks (ghost tracks). The coloured patches inside each bbox present the foreground pixels which are detected as moving pixels (the colours of each patch are randomly assigned by the tracker). In Figures 5c and 5d, the white minibus moved away from where it had stopped, leaving a ghost (red bbox with green patch in Figure 5d) behind. In Figures 5e and 5f, by using the proposed ghost detection method, we successfully identified the ghost object (blue bounding bbox with green patch with the text “ghost” on it).

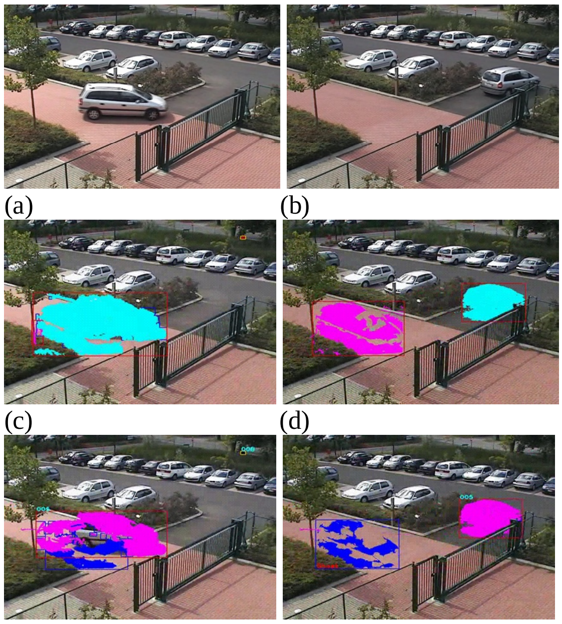
Figure 6 presents results from the KND2-JULY-EAST-S2.avi sequence. In Figure 6c frame 5856 a car is about to start. After a while, in frame 5880 (Figure 6d), the car has left its parked area and a ghost track has been formed (red box with yellow patch). Figures 6e and 6f present the tracker with ghost detection: in frame 5880 (Figure 6f), the car left its parked area, however, the ghost patch has been successfully identified (blue bbox with yellow patch) and furthermore prevented from becoming valid and eliminated after a few frames.

Figures 7c and 7d present the results by the basic motion tracker and Figures 7e and 7f are produced by Ghost detection tracker. As we can see Figure7d, after the vehicle (frame 9556, red bbox with light blue patch) has left, a ghost track was left behind (red bbox with purple patch). With ghost detection in Figures 7e and 7f, the ghost object (blue bbox with blue patch) has been successfully detected.



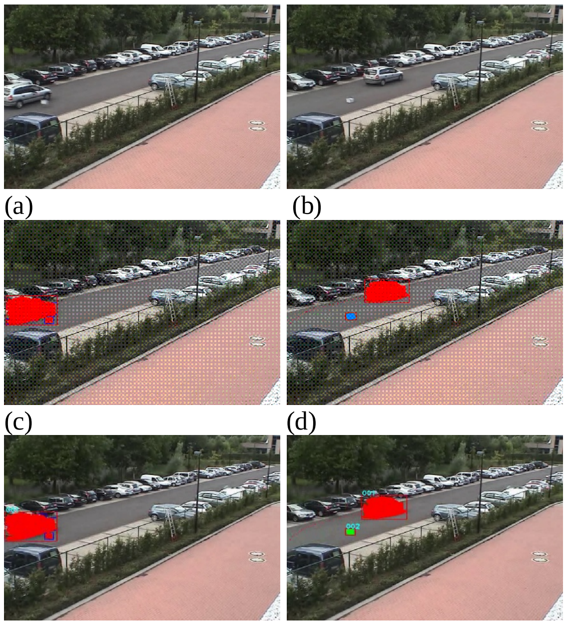
(a) (b) (c) (d) (e) (f)

Figure 6 KND2-JULY-EAST-S2.avi sequence is 21794 frames (00:12:55) long and depicts 23 objects. The main challenges are quick illumination changes and 2 ghosts
a) frame 5856 b) frame 5880 c) basic tracker at frame 5856
d) basic tracker at frame 5880 e) tracker with ghost detection at frame 5856 f) tracker with ghost detection at frame 5880



(a) (b) (c) (d) (e) (f)

Figure 7 KND2-JULY-GATE1-S2.avi sequence is 18180 frames (00:12:32) long and depicts 29 objects. The main challenges are quick illumination changes and 3 ghosts
a) frame 9496 b) frame 9556 c) basic tracker at frame 9496
d) basic tracker at frame 9556 e) tracker with ghost detection at frame 9496 f) tracker with ghost detection at frame 9556



(a) (b) (c) (d) (e) (f)

Figure 8 KND2-JULY-EAST-S3.avi sequence is 18102 frames (00:10:04) long and depicts 29 objects. The main challenges are quick illumination changes, abandoned bags.
a) frame 13460 b) frame 13510 c) basic tracker at frame13460
d) basic tracker at frame 13510 e) tracker with ghost detection at frame 13460 f) tracker with ghost detection at frame 13510

Figures 8e and 8f present the result from the Ghost detection tracker. A box (frame 13510, red bbox with green patch) was dropped from the car, and our algorithm can still successfully detect the abandoned item without miss classifying as a ghost.

From the figures shown above, we can see that ghost detection is effective enough to detect ghost objects, to prevent the tracks from becoming “valid”, and furthermore to avoid false alarms. At the same time, the algorithm can differentiate between ghost objects and abandoned items.

We experimented with the frequency of ghost detection by using different values of k (e.g. k = 1, 2, 5, 10, 12, 15, 20…) and testing our algorithm. There is no effect on the performance as long as k is under 20. Our decision to set k to 12 ensures fast and reliable ghost detection.

We also present some quantitative evaluation results (Tables 1 - 3) based on the performance evaluation framework described in detail in [6]. The improvement is mainly reflected by the metric: False alarm tracks (FATs). In the PETS2001 sequence (table 1), there are two false alarms caused by ghost objects in the results of the basic tracker. In the results of ghost detection tracker, the 2 false alarms caused by ghosts have been avoided. Similarly, in KND2-JULY-EAST-S2 sequence (table 2), we had 2 cases of false alarms caused by ghosts, but the tracker with ghost detection avoided these 2 FATs. Hence, the number of FAT is reduced from 5 to 3. Similarly, in KND2-JULY-GATE1-S2 sequence (table 3), we had 3 cases of FATs caused by ghosts, but the tracker with ghost detection avoided these 3 FATs. Hence, the number of FAT is reduced from 14 to 11.

The speeds of the basic tracker and ghost detection tracker are fast enough to operate in real-time, although there is about 10% to 15% decrease in speed when adding ghost detection into the basic tracker. In addition, except false alarm track, other metrics such as correct detected track, track detection failure, track fragmentation, average latency, distance error remain more or less the same for both basic tracker and ghost detection tracker, which means that the ghost detection does not cause any undesirable side-effects on tracking the objects.

According to the overall evaluation results, the ghost detection algorithm can effectively eliminate ghost objects and brings no significant side-effects caused by the ghost detection algorithm to the surveillance system.

Table 1 Performance evaluation for Pets2001.avi

PetsD1TeC1.avi	Basic tracker	Ghost detection
Total num of frames	2686	2686
Speed	77.4 f/s	63 f/s
GT Tracks	9	9
System Tracks	12	11
CorrectDetectedTrack	9	9
FalseAlarmTrack	3	1
TrackDetectionFailure	0	0
Trackfragmentation	3	4
ID Change	5	6
AverageLatency	46	49
AverageOverlap	0.47	0.55
OverlapDeviation	0.24	0.29
DistanceError	15.75	16.84
DistanceErrorDeviation	23.64	24.23
AverageCompleteness	0.67	0.72

Table 2 Performance evaluation for KND2-JULY-EAST-S2.avi

KND2-JULY-EAST-S2.avi	Basic tracker	Ghost detection
Total num of frames	21794	21794
Speed	28.9 f/s	25.5 f/s
GT Tracks	23	23
System Tracks	30	28
CorrectDetectedTrack	19	20
FalseAlarmTrack	5	3
TrackDetectionFailure	4	3
Trackfragmentation	1	2
ID Change	1	0
AverageLatency	25	29
AverageOverlap	0.69	0.70
OverlapDeviation	0.29	0.29
DistanceError	10.65	9.92
DistanceErrorDeviation	19.97	21.66
AverageCompleteness	0.52	0.51

Table 3 Performance evaluation for KND2-JULY-GATE1-S2.avi

KND2-JULY-GATE1-S2.avi	Basic tracker	Ghost detection
Total num of frames	18180	18180
Speed	25.2 f/s	21.6 f/s
GT Tracks	29	29
System Tracks	58	57
CorrectDetectedTrack	26	26
FalseAlarmTrack	14	11
TrackDetectionFailure	3	4
Trackfragmentation	5	6
ID Change	0	0
AverageLatency	37	34
AverageOverlap	0.58	0.57
OverlapDeviation	0.25	0.24
DistanceError	27.92	27.21
DistanceErrorDeviation	55.64	54.08
AverageCompleteness	0.75	0.75

5. Conclusion

We have presented an efficient ghost detection and removal algorithm that can be applied to any motion detection and tracking system that produces a difference map per frame and bounding boxes for foreground objects.

According to the visual results and performance evaluation results, we can effectively avoid false alarms caused by ghost objects while still differentiating between ghost objects and abandoned objects. The method works well in several scenarios: PETS2001 and another two car park scenarios. The Ghost removal version of basic tracker runs almost as fast as the original and do not have any negative side-effect on detection and tracking.

However, our method is limited by the performance of the tracker, e.g. if the tracker can not recognise individual objects in crowd scenes, our method will not be able to detect ghosts either.

In future work, we want to test our algorithm in videos with textured background to check if that would cause confusion between the edges of the pattern and the object.

6. Acknowledgements

The authors would like to acknowledge financial support from BARCO View, a major multinational manufacturer of CCTV display systems, whose headquarters are in Belgium. They are also grateful to the Engineering and Physical Sciences Research Council's REASON project for providing access to data.

7. References

[1] W. Hu, T. Tan, L. Wang, and S. Maybank. “A survey on visual surveillance of object motion and behaviors”. IEEE Trans. on Systems, Man, and Cybernetics - Part C, 34(3):334-352, August 2004

[2] C. Stauffer, W.E.L. Grimson, “Adaptive background mixture models for real-time tracking”, in: Computer Vision and Pattern Recognition, vol 2, p.2246, Santa Barbara, CA, June, 1998

[3] R. Cucchiara, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," IEEE Transactions on Pattern Analysis and Machine Intelligence 25, pp. 1337-1342, Oct 2003

[4] Sen-Ching S. Cheung, Chandrika Kamath, “Robust Background Subtraction with Fore-ground Validation for Urban Traffic Video”, EURASIP Journal on Applied Signal Processing, pp. 2330-2340, 2005.

[5] Sadiye Guler, Jason A. Silverstein, Ian H. Pushee, "Stationary Objects in Multiple Object Tracking", IEEE International Conference on AVSS, pp.248-253, 2007

[6] Fei Yin, D. Makris, S.A. Velastin, "Performance Evaluation of Object Tracking Algorithms", 10th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS2007), October, Rio de Janeiro, Brazil, 2007

[7] OpenCV Computer Vision Library http://www.intel.com/technology/computing/ opencv/index.htm, [Last accessed: August 2007]

[8] Guide to Authoring Media Ground Truth with ViPERGT, http://viper-toolkit.sourceforge.net/docs/gt/, [Last accessed: July 2007]

[9] D.S. Lee, “Effective Gaussian mixture learning for video background substraction”, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (5) pp.827–832, 2005

[10] Sijun Lu Jian Zhang Feng, D. "An efficient method for detecting ghost and left objects in surveillance video",

IEEE Conference on Advanced Video and Signal Based Surveillance, pp.540-545, London, Sept. 2007

[11] Barco Display and visualization solutions, http://www.barco.com/ [Last accessed: May 2008]

[12] X. Desurmont, C. Chaudy, A. Bastide, C. Parisot, J.F. Delaigle, B. Macq, "Image analysis architectures and techniques for intelligent systems", IEE Proceedings on Vision, Image and Signal Processing, Special issue on Intelligent Distributed Surveillance Systems, pp. 224-231, April 2005.