Temporally Static Region Detection in Multi-Camera Systems

Fatih Porikli
Mitsubishi Electric Research Labs
Cambridge, MA 02139

Zhaozheng Yin
The Pennsylvania State University
University Park, PA 16802

Abstract

Traditional approaches consider left behind object detection as a tracking application and heavily depend on accurate initialization of objects, which is a performance bottleneck. Here, we present a pixel-based solution that employs dual foregrounds of different scene modalities. We construct separate long- and short-term backgrounds modeled as multilayer, multivariate Gaussian distributions. These backgrounds are adapted online using a Bayesian update mechanism at different learning rates that can be imposed as different frame processing frequencies. In addition, the formulation for color background can be easily extended for the gradient and feature point representations. By comparing the current frame with the background modes, we construct dual foregrounds. We aggregate evidence scores at each camera to provide temporal consistency on the hypotheses inferred from the foregrounds. We fuse the evidence from multiple cameras on a ground plane with the associated confidence scores to eliminate the individual camera failures due to the lighting artifacts. Our method does not require object initialization, tracking, or offline training. It accurately segments objects even if they are fully occluded. Its computational load is low and it readily lends itself to parallelization if further speed improvements are necessary.

1. Introduction

Left behind item detection approaches can be grouped as motion detectors [1, 2, 3], object classifiers [4], and tracking based analytics approaches [5, 6, 7, 8, 9, 10].

In [2], a dense optical flow map is estimated to infer the foreground objects moving in opposite directions, moving in a group, and stay stationary by predetermined rules. In [3], a pixel-based method for characterizing objects introduced into the static scene by comparing the background image estimated from the current frame with the previous ones is described. This approach requires storing of as many backgrounds as the minimum detection duration in the memory and causes ghost detections even after the left-behind item is removed from the scene.

Recently, an online classifier that incorporates a boosting based feature selection to label image blocks as background, valid objects, and unidentified regions is presented in [4]. This method adapts itself to the depicted scene, however, fails short of discriminating moving objects from stationary ones. Classifier based methods face with the challenge of dealing with unknown left-behind object type as such objects can vary from small luggage to ski bags.

A considerable amount of effort has been devoted to hypothesize left-behind items by analyzing object trajectories [5, 6, 7, 9, 10] in multi camera setups. In principle, these methods require solving a harder problem of object initialization and tracking as an intermediate step in order to identify the parts of the video frames corresponding to a left behind object. Object detection in crowded scenes, especially for uncontrolled real-life situations, is problematic due to the partial occlusions, heavy shadows, people entering the scene together, etc. Moreover, object appearance is often indiscriminative as people tend to dress in similar colors, which leads inaccurate tracking results.

For static camera setups, background subtraction provides strong cues for apparent motion statistics. Various background generation methods have been employed in a quest for a system that is robust to changing illumination conditions, appearance variations, shadows, camera jitter, and severe noise. Parametric mixture models are employed to handle such variations. Stauffer and Grimson [11] propose an expectation maximization (EM) based adaptation method to learn a mixture of Gaussians with predetermined number of models at each pixel using fixed learning parameters. The online EM update causes a weak model, which has a larger variance, to be dissolved into a dominant model, which has a smaller variance, in case the mean value of the weak model is close to the mean of the dominant one. To address this issue, Porikli and Tuzel [12] develop an online Bayesian update mechanism for adaptation multivariate Gaussian distributions. This method estimates the number of necessary layers for each pixel and the posterior distributions of mean and covariance of each layer by assuming the data to be normally distributed with mean and covariance as random variables.

However, there exists a class of problems that cannot be solved by the traditional foreground-background detection methods. For instance, objects deliberately abandoned in

public places, such as suitcases, packages, etc. do not fall into either of these two categories. They are static; therefore, they should be labeled as background. On the other hand, they should not be ignored as they do not belong to the original scene background. Depending on the learning rate, the pixels corresponding to the temporary static objects can be mistaken as a part of the scene background (in case of a high learning rate), or grouped with the moving regions (low learning rate). A single background is not sufficient to separate the temporarily static pixels from the scene background.

In this paper, we propose a pixel-based method that employs dual foregrounds for multi camera setups. Our motivation is that by changing the background learning rate, we can adjust how soon a static object should be blended into the background. Therefore, temporarily static image regions can be distinguished from the longer term background and moving regions by analyzing multiple foregrounds of different learning rates. This simple idea is wrapped into our adaptive background estimation algorithm, where the slowly adapting background and the fast adapting foreground are aggregated into an evidence image. We impose different learning rates by processing video at different temporal resolutions. The background models have identical initial parameters, thus they require minimal fine tuning in the setup stage. The evidence statistics are used to extract temporarily static image areas at each camera. We fuse the evidence from multiple cameras on a ground plane with associated confidence scores to eliminate the individual camera failures due to the lighting artifacts. In addition, we extend the color backgrounds for the gradient and feature point representations.

Our method does not require object initialization, tracking, or off line training. It accurately segments objects even if they are fully occluded. It has a low computational load and readily lends itself to parallelization if further speed improvements are necessary.

2. Dual Foregrounds

To detect a left behind item (or an illegally parked vehicle, removed article, etc.), we need to know how it alters the temporal and spatial statistics of the video data. We built our method on the fact that an abandoned item is not a part of the original scene, it was brought into the scene not that long ago, and it remained still after it has been left. In other words, it is a temporarily static object which was not there before. This means that by learning the prolonged static scene and the moving foreground regions, we can hypothesize on whether a pixel corresponds to an abandoned item or not.

As opposed to single background approaches, we use two backgrounds to obtain both the prolonged (long-term) background B_L and the temporarily static (short-term) background B_S. Note that, it is possible to improve the temporal granularity by employing more than two backgrounds at different learning rates.

Our background model [12] is most similar to adaptive mixture models [11] but instead of mixture of Gaussian distributions, we define each pixel as layers of 3D multivariate Gaussians. Each layer corresponds to a different appearance of the pixel. We perform our operations in the RGB color space. Using Bayesian update, we are not estimating the mean and variance of the layer, but the probability distributions of mean and variance. We can extract statistical information regarding to these parameters from the distribution functions. We use the expectations of mean and variance for change detection, and variance of the mean for confidence. Bayesian update algorithm maintains the multimodailty of the background model.

Learned background statistics are used to detect the changed regions of the scene. We determine how many layers are necessary for each pixel and use only those layers during foreground segmentation phase. The number of layers required to represent a pixel is not known beforehand so background is initialized with more layers than needed. Usually we select three to five layers. In more dynamic scenes more layers are required. Using the confidence scores we determine how many layers are significant for each pixel. As we observe new samples for each pixel we update the parameters for our background model. At each update, at most one layer is updated with the current observation. This assures the minimum overlap over layers. We order the layers according to confidence score and select the layers having confidence value greater than the layer threshold. We refer to these layers as confident layers. We start the update mechanism from the most confident layer. If the observed sample is inside the 2.5σ of the layer mean, which corresponds to 99% confidence interval of the current model, parameters of the model are updated. Lower confidence models are not updated. Details can be found in [12].

At every frame, we estimate the long- and short-term foregrounds by comparing the current frame I by the background models B_L and B_S. We obtain two binary foreground masks F_L and F_S where $F(x, y) = 1$ indicates the pixel (x, y) is changed. The long-term foreground mask F_L shows the color variations in the scene that were not there before including moving objects temporarily static objects, as well as moving cast shadows and illumination changes that the background models fail to adapt. The short-term foreground mask F_S contains the moving objects, noise, etc. Depending on the foreground mask values, we postulate the following hypotheses as shown in Figure 1:

1. $F_L(x, y) = 1$ and $F_S(x, y) = 1$; (x, y) is a pixel that may correspond to a moving object since $I(x, y)$ does

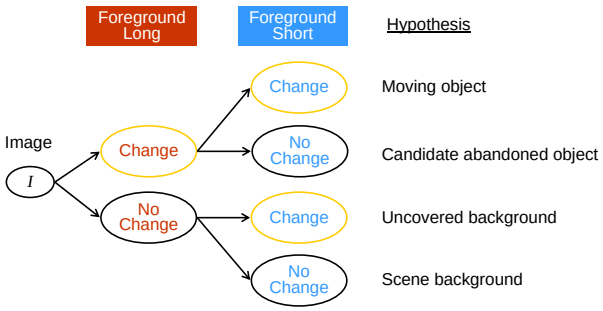


Figure 1: Hypotheses on long and short-term foregrounds.

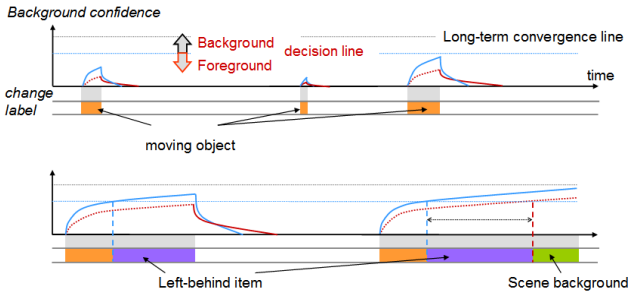


Figure 2: The confidence of the long-term and short-term background models (vertical axis) change differently for ordinary objects (moving or temporarily stationary ones), left-behind items, and scene background.

not fit any backgrounds.

2. $F_L(x,y) = 1$ and $F_S(x,y) = 0$; (x,y) is a pixel that may correspond to a temporarily static object.

3. $F_L(x,y) = 0$ and $F_S(x,y) = 1$; (x,y) is a scene background pixel that was occluded before.

4. $F_L(x,y) = 0$ and $F_S(x,y) = 0$; (x,y) is a scene background pixel since its value $I(x,y)$ fits both backgrounds B_L and B_S.

The short-term background is updated at a higher learning rate than the long-term background. Thus, the short-term background adapts to the underlying distribution faster and the changes in the scene are blended more rapidly. In contrast, the long-term background is more resistant against the changes.

Our update mechanism prevent from momentary changes to contaminate these backgrounds, and such pixels are labeled as foregrounds; $F_S(x,y) = 1$ and $F_L(x,y) = 1$. In case a scene background pixel changes temporarily then sets back to its original value, the long-term foreground mask will be zero; $F_L(x,y) = 0$. The short-term background is pliant and adapts itself during this time, which causes $F_S(x,y) = 1$. We assume it takes more time to adapt the long-term background to the newly observed color than the change period. A changed pixel will be blended into the short-term background i.e. $F_S(x,y) = 0$ if it keeps its new color long enough. If this duration is not prolonged enough to blend it the long-term foreground mask will be one; $F_L(x,y) = 1$. This is the common case for the abandoned items. If no change is observed in neither of the backgrounds $F_L(x,y) = 0$ and $F_S(x,y) = 0$, the pixel is considered as a part of the static scene background as the pixel has the same value for much longer periods of time.

The dual foreground mechanism is illustrated in Figure 2. In this simplified drawing, the horizontal axis corresponds to time and the vertical axis to the confidence of the background model. *Action* indicates that the pixel color has significantly changed. *Label* represents the result of the above hypotheses. For pixels with relatively short duration of change, the confidences of the long- or short-term models do not increase enough to make them valid backgrounds. Thus, such pixels are labeled as moving object. Whenever the short term model blends the pixel in the background but the long term model still marks it as foreground, the pixel is considered to belong to the left-behind item. Finally, if the pixel change takes even longer the pixel is labeled as a scene background. Sample foregrounds that show these cases are given in Figure 3.

We aggregate the frame-wise detection results into an evidence image $E(x,y)$ by updating the pixel-wise values at each frame as

$$E(x,y) = \begin{cases} E(x,y)+1 & F_L(x,y) = 1 \wedge F_S(x,y) = 0 \\ E(x,y)-k & F_L(x,y) \neq 1 \vee F_S(x,y) \neq 0 \\ max_e & E(x,y) > max_e \\ 0 & E(x,y) < 0 \end{cases}$$

where max_e and k are positive numbers. The evidence image enables removing noise in the detection process. It also controls the minimum time required to assign a static pixel as an abandoned item. For each pixel, the evidence image collects the motion statistics. Whenever it elevates up to a preset level $E(x,y) > max_e$, we mark the pixel as an abandoned item pixel and raise an alarm flag. The evidence threshold max_e is defined in term of the number of frames and it can be chosen depending on the desired responsiveness and noise characteristics of the system. In case the foreground detection process produces noisy results, higher values of max_e should be preferred. High values of max_e lower the false alarm rate. On the other hand, higher the preset level gets, longer the minimum duration a pixel takes to be classified as a part of an abandoned item. A typical range of the evidence threshold max_e is 300 frames.

The decay constant k determines how fast the evidence should decrease. In other words, it decides what should hap-

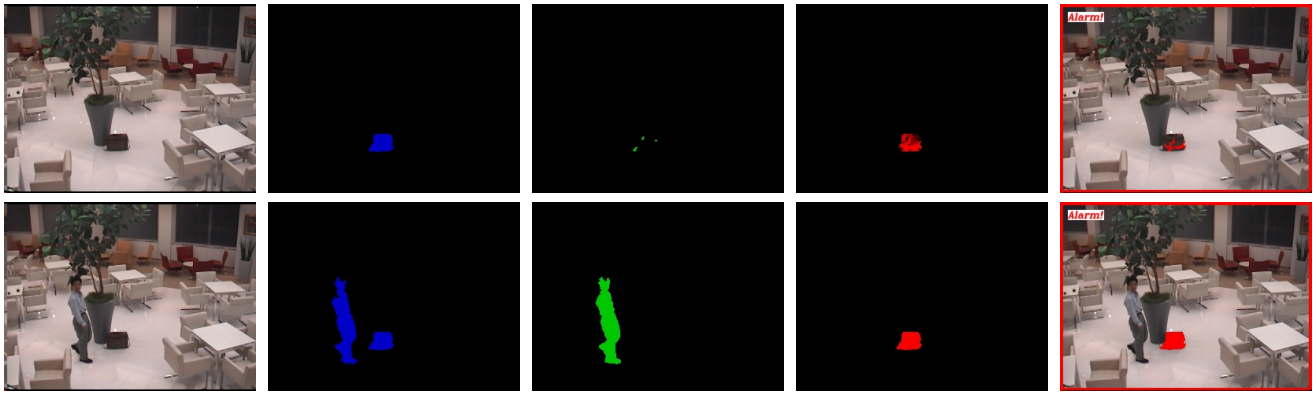


Figure 3: First row: $t = 350$. Second row: $t = 630$. The long-term foreground F_L captures moving objects and temporarily static regions. The short-term foreground F_S captures only moving objects. The evidence E gets greater as the object stays longer.

pen in case a pixel that is marked as an abandoned item is blended into the scene background or gets its original value before the marking. To set the alarm flag off immediately after the removal of object, the value of decay should be large, e.g., $k = max_e$. This means there is only a single parameter to set for the likelihood image. In our experiments we observed the larger values of decay constant generate satisfying results.

3. Multiple Camera Fusion

We fuse the individual detection results obtained for each camera onto a ground plane. We compute the homography matrices using multiple pairs of corresponding points, which are selected manually. In case of a moving camera setup, these transformation matrices can be obtained by a feature detector and RANSAC. All the images are rectified onto the ground plane coordinate system, where the center of the image represent $(0, 0)$ in the world coordinate system and one pixel length in the image represents approximately $2cm$. Figure 4 shows examples of the pixel-level detection results in single camera views and the rectified images. The red pixels represent possible left-behind item regions.

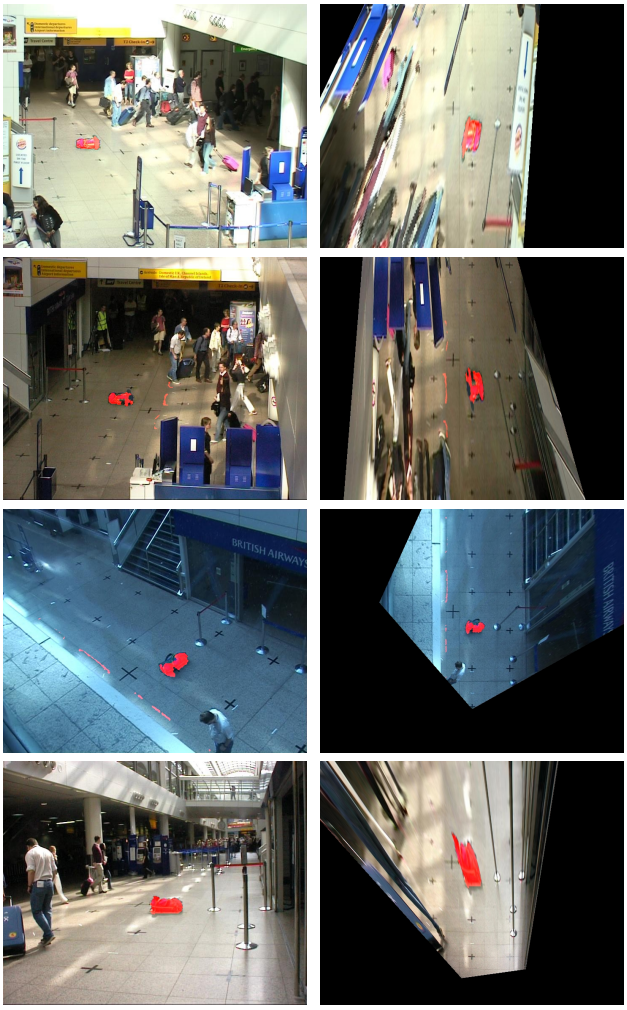
In addition to the color backgrounds, feature points such has corner or high gradient magnitude points can be computed and the backgrounds are fit on the feature points. Figure 5 gives the detection results obtained for edge features using Canny operator and Harris corner point features. We observed similar detection performance in both color and feature point backgrounds, except a few very short termed (couple of frames) false alarms were apparent in the feature point version. On the other hand, the feature point background runs at least $10\times$ faster than the color backgrounds.

If no other confidence score that indicates the saliency of the individual camera estimations is available, the detection results, which are binary masks, can be simply warped and added to find a ground plane evidence map $EG_t(x, y) = \sum_i E_t^{i,w}(x, y)$ where $E_t^{i,w}(x, y)$ is the warped i-th camera image at frame t as shown in Figure 6. For S08 dataset, EG has the maximum value of four, which means that pixel is detected as a possible left-behind luggage pixel in all the four views. We impose that a pixel should be detected in at least 3 camera views simultaneously, which means the ground plane evidence map EG is thresholded at $\tau = 3$. The ground plane evidence EG can fluctuate pixel-wise due to this hard thresholding. To improve such inconsistencies a ground plane evidence history map EGH can be constructed to provide temporal smoothing on the observed results as

$$EGH_t = \begin{cases} max(0, EGH_{t-1} - \alpha) & EG_t < \tau \\ 1 & EG_t \geq \tau \end{cases} \quad (1)$$

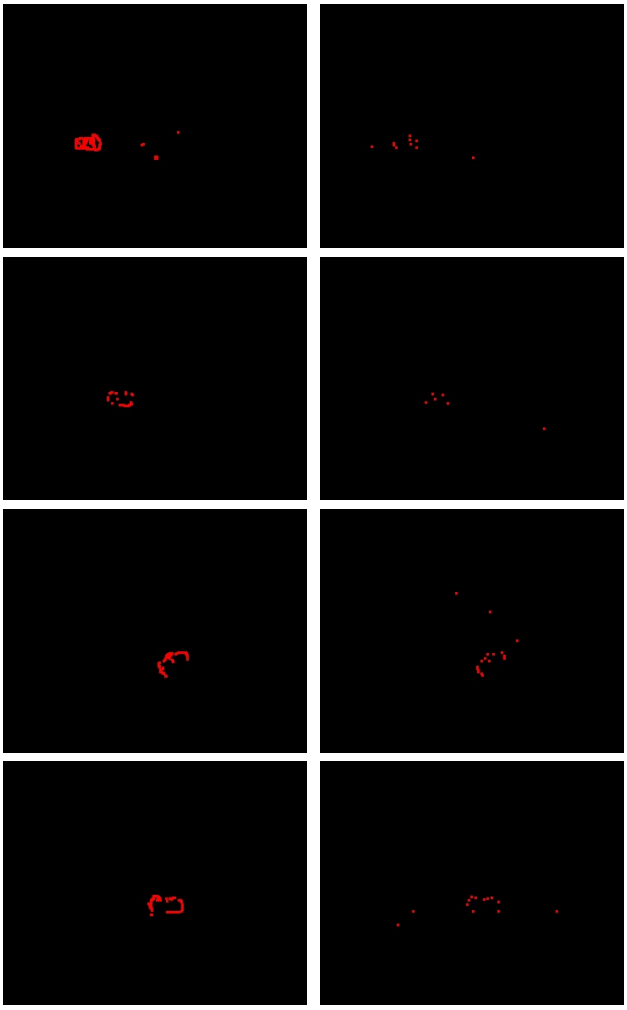
Above update formulation combines the ground plane detections over a temporal window of $1/\alpha$ frames by decaying the results of the previous frames by α in case the detection is below the imposed threshold τ. In other words, EGH represents a cumulative map of the brighter regions of recent frames together with gradually fading older frame detection results. EGH enables to make a soft decision even though a hard threshold in applied.

Figure 6 illustrates the EGH map with $\alpha = 1/25$, i.e. the temporal window size is 25. At frame $t = 1860$ the left luggage is detected accurately, however, at frame $t = 1890$ the luggage is partially occluded and most pixels of EG_{1890} is below threshold $\tau = 3$. Since the luggage is detected accurately in the previous frames, at frame $t = 1890$ it is detected with decayed score of EGH_{1890}. When the luggage is removed at frame $t = 1930$, its detection



Original view Ground plane view

Figure 4: Detection results in each single view and the corresponding warped images on the ground plane.



Canny Harris

Figure 5: Canny (left) and Harris corner (right) feature based detections for $t = 1850$.

score is significantly decayed.

3.1 Confidence Maps

Illumination artifacts, especially the ones that destroys the appearance patterns, might contaminate the detection results of individual camera observations. We tested several strategies to enhance the performance under such conditions.

Spatial contrast: Considering left-behind items likely to have a different colors, contrast, from their surrounding backgrounds (as they are visible in F_L), the pixels within an inner region of a detected pixel should have distinctive color distribution from an outer region that is supposed to outside the left behind item. For each detected pixel p, it is possible to compute an inner color histogram within a $h \times h$ and an outer color histogram within the surrounding $2h \times 2h$ layer as illustrated in Figure 7-left. Then, a confidence score of each pixel, i.e. a spatial contrast value, can be determined as the distance between the inner and outer region histograms, e.g. using quadratic or Bhattacharya norms. The region dimension h can be set by an average size on the ground plane and projecting onto the image planes. Figure 8 shows samples computed for $h = 11$. This approach helps removing false detections in smoother image regions. For each camera view, instead of adding the detection results directly, the weighting scheme can be applied (Figure 7).

Temporal intensity change: Another way of removing false detections due to severe illumination changes, which may not be compensated successfully by the background update, is to weight each camera results inversely by an estimated temporal intensity change score. For instance,

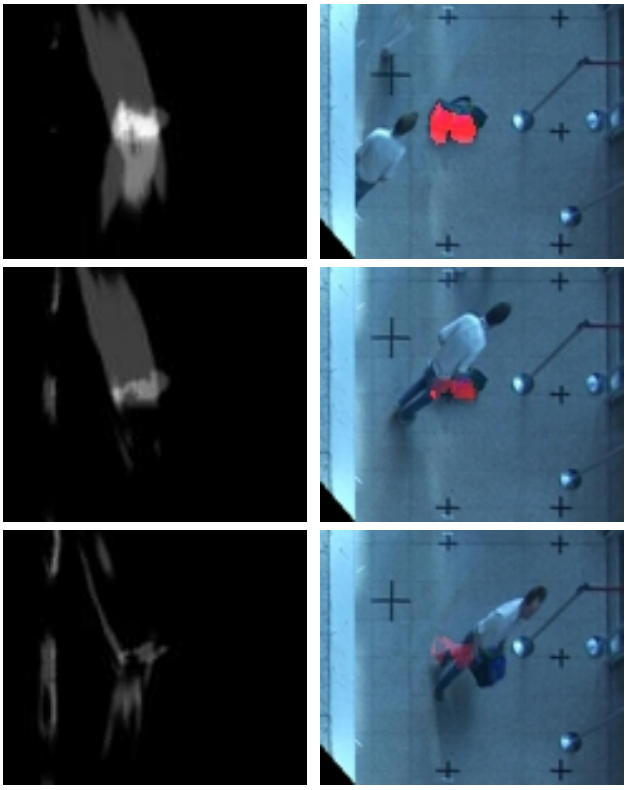


Figure 6: Left: EG_t, right: EGH_t (from the third camera view as ground plane). Rows show the detection results at $t = 1860, 1890, 1930$ of S08 dataset.

as shown in Figure 9-left, a probe window that does not contain any crowd motion area can be selected in the setup time. For the pixels in this probe window, average intensity change ($\triangle Y/\triangle t$) is computed between the current and previous frames, preferably at a low temporal resolution to allow detection of subtle changes. Larger change scores indicate the smaller the confidence weights (β_i in Figure 7) for the corresponding cameras. This approach can accurately identify global illumination changes as it was shown in Figure 9-right and weight each camera's feedback accordingly.

Figure 10 is sample results from S08 dataset from PETS-2007. The combined results are given in the last column for EGH and thresholded EGH on the ground plane. In our experiments, we detected the left-behind item and got no false alarms due to the illumination artifacts that caused artificial patterns on the background after the object was removed from the scene. As visible, none of the moving objects, moving shadows, people that are stationary in shorter durations was falsely detected. Another advantage of this method is that the alarm is immediately set of as soon as the abandoned item is removed from its previous position.

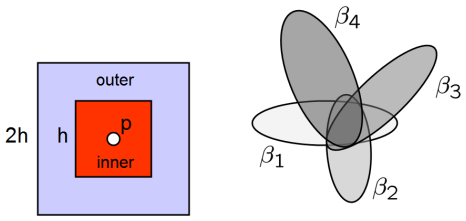


Figure 7: Left: inner and outer regions around a pixel where spatial contrast is computed using the regions histograms. Right: weighted sum is computed by compensating for the amount of the global illumination changes at each camera where larger illumination changes result in smaller camera confidence weights.

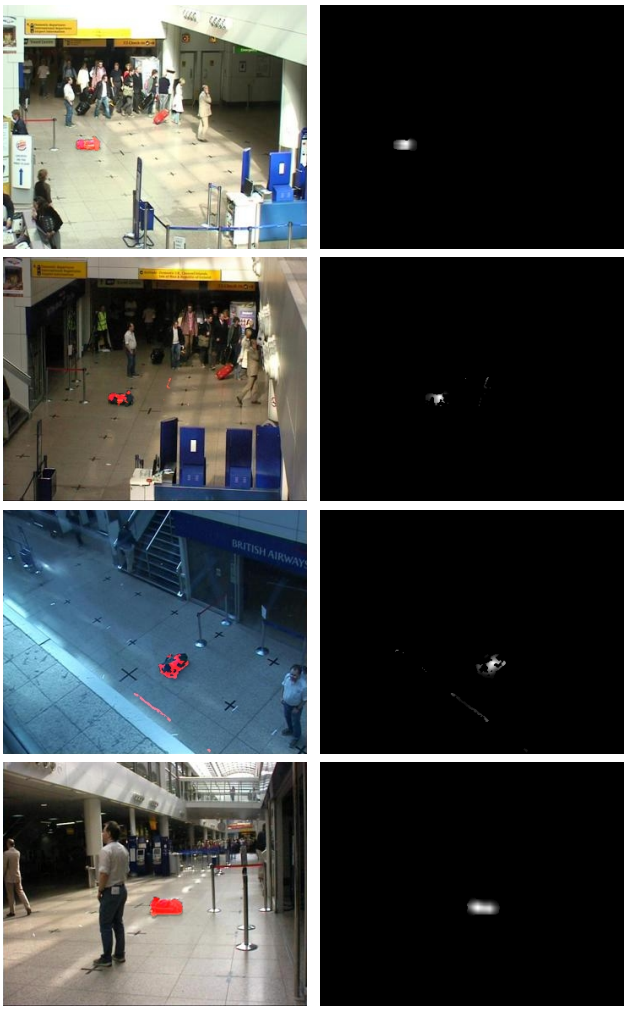


Figure 8: Left: original detections. Right: weighting by spatial contrast cue to filter inaccurate temporally static region detections due to the illumination variations at high gradient background pixels.

Figure 9: Left: probe window. Right: intensity changes.

One shortcoming is that it cannot discriminate the different types of objects, e.g. a person who is stationary for a long time can be detected as a left behind item. Since no tracking is integrated, trajectory based semantics, e.g. who left the item or how long the item left before the person moves away can not be extracted. Still, our method can be used as a preprocessing stage to improve the tracking based video analytics.

The computational load is low; the processing of multiple sequences is done in real-time as it is sufficient to analyze the streaming videos in much lower frame rates. Since we only employ pixel-wise operations and make pixel-wise decisions, we can take advantage of the parallel processing architectures. By assigning each image pixel to a processor on the GPU using CUDA programming, since each processor can execute in parallel, the speed improves more than $14\times$ in comparison to the corresponding CPU implementation. For instance, full background update for 360×288 images takes 74.32msec on CPU (P4 DualCore 3Ghz), however on CUDA it only needs 6.38msec. We observed that the left-behind item detection can be comfortably applied in quarter spatial resolution by processing the short-term background at 5 fps while updating the long-term at every 5 seconds (0.2 fps).

4. Conclusions

A computationally efficient and robust method to detect temporally static regions in multi-camera setups is presented. This method does not depend on object initialization and tracking and accurately outlines the boundary of items even if they are fully occluded later. It uses two backgrounds that are learned by processing the input videos at different frame rates and aggregates evidence both at each camera and on ground plane for multiple cameras. It employs pixel and camera based confidences to weight individual camera estimations accordingly. Since it executes pixel-wise operations it can be implemented on parallel processors.

References

[1] J. D. Courtney, "Automatic video indexing via object motion analysis," *PR*, vol. 30, no. 4, pp. 607–625, 1997.

[2] S. Velastin and A. Davies, "Intelligent CCTV surveillance: Advances and limitations," 2005.

[3] A. E. Cetin, M. B. Akhan, B. U. Toreyin, and A. Aksay, "Characterization of motion of moving objects in video," *United States Patent 20040223652*, 2004.

[4] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* New York, NY, vol. 1, 2006, pp. 260–267.

[5] E. Auvinet, E. Grossmann, C. Rougier, M. Dahmane, and J. Meunier, "Left-luggage detection using homographies and simple heuristics," in *PETS*, 2006, pp. 51–58.

[6] J. M. del Rincon, J. E. Herrero-Jaraba, J. R. Gomez, and C. Orrite-Urunuela, "Automatic left luggage detection and tracking using multi-camera UKF," in *PETS*, 2006, pp. 59–66.

[7] P. T. N. Krahnstoever, T. Sebastian, A. Perera, and R. Collins, "Multi-view detection and tracking of travelers and luggage in mass transit environments," in *PETS*, 2006, pp. 67–74.

[8] F. Lv, X. Song, B. Wu, V. K. Singh, and R. Nevatia, "Left luggage detection using bayesian inference," in *PETS*, 2006, pp. 83–90.

[9] K. Smith, P. Quelhas, and D. Gatica-Perez, "Detecting abandoned luggage items in a public space," in *PETS*, 2006, pp. 75–82.

[10] S. Guler and M. K. Farrow, "Abandoned object detection in crowded places," in *PETS*, 2006, pp. 99–106.

[11] C. Stauffer and E. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Fort Collins, CO, vol. II, 1999, pp. 246–252.

[12] F. Porikli and O. Tuzel, "Bayesian background modeling for foreground detection," in *Proc. of ACM Visual Surveillance and Sensor Network*, 2005.

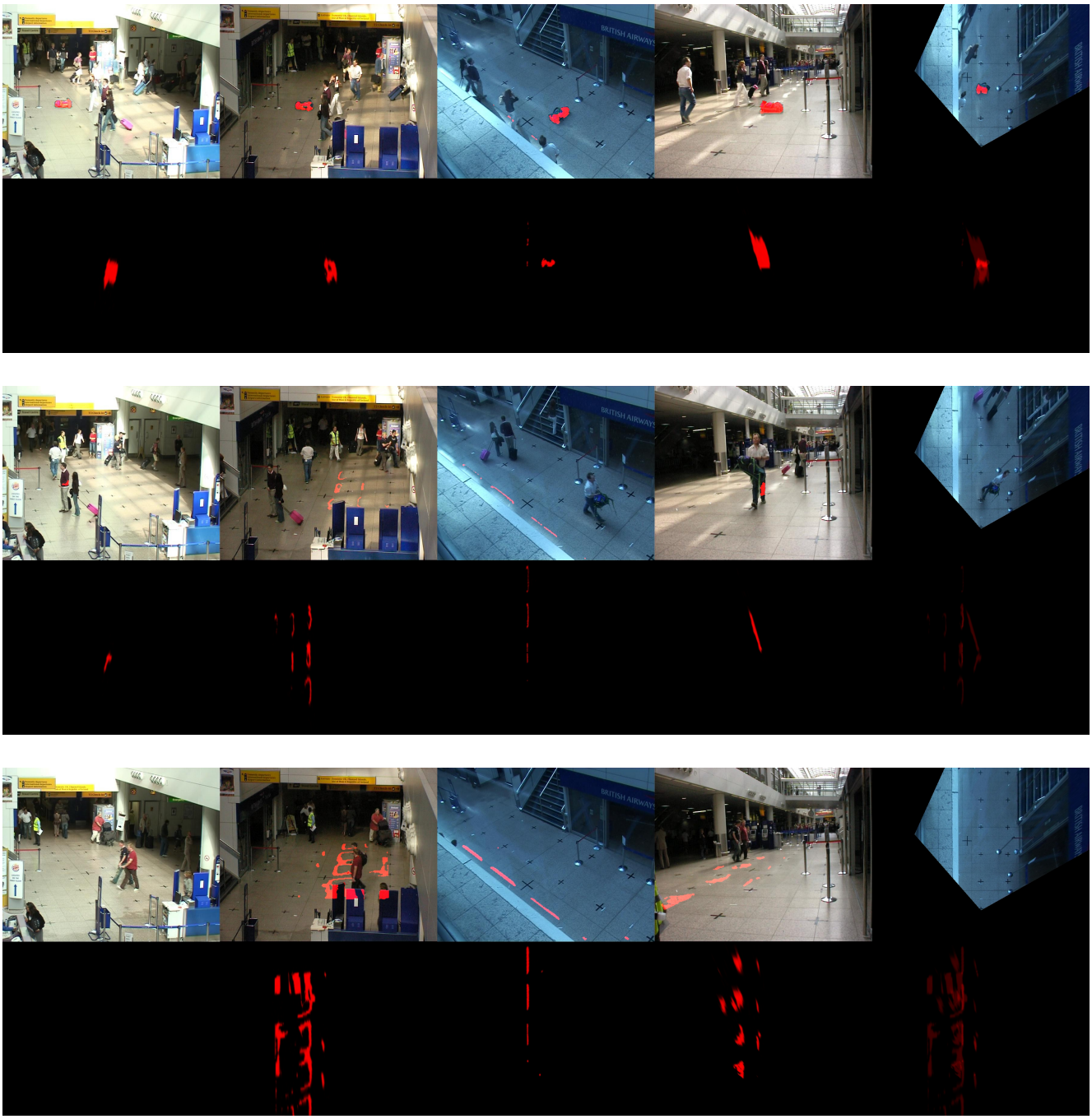


Figure 10: Pixel-level detections in the four camera views and the fused results on the ground plane (last column). Results for $t = 1850, 1950, 2900$. As visible, fusion significantly helps removing noise in individual camera views as there is no false detections in the fused results where full red color indicates a detection (not shades of red). Alarm sets off right after the object is removed.