ROBUST ABANDONED OBJECT DETECTION USING REGION-LEVEL ANALYSIS

Jiyan Pan, Quanfu Fan, Sharath Pankanti

IBM T.J. Watson Research Center, Hawthorne, NY, U.S.A.

ABSTRACT

We propose a robust abandoned object detection algorithm for real-time video surveillance. Different from conventional approaches that mostly rely on pixel-level processing, we perform region-level analysis in both background maintenance and static foreground object detection. In background maintenance, region-level information is fed back to adaptively control the learning rate. In static foreground object detection, region-level analysis double-checks the validity of candidate abandoned blobs. Attributed to such analysis, our algorithm is robust against illumination change, "ghosts" left by removed objects, distractions from partially static objects, and occlusions. Experiments on nearly 130,000 frames of i-LIDS dataset show the superior performance of our approach.

Index Terms— Video surveillance, background estimation, abandoned object detection

1. INTRODUCTION

Abandoned object detection is one of the most important tasks in automated video surveillance systems. A lot of research has been devoted to developing abandoned object detection algorithms. Among them, the most popular approaches are based on background subtraction, due to their superior robustness in complex real-world scenarios [1, 2, 3, 4]. In such approaches, two major components are background maintenance and static foreground object detection. For background maintenance, many algorithms employ a mixture of Gaussians [5] to model the background and foreground for each individual pixel [6, 2, 3, 7]. However, information carried by individual pixels is highly limited. As a result, Gaussian Mixture Model (GMM) is frequently observed to generate rather noisy background estimation. In addition, pixel-level analysis is rather difficult to handle rapid illumination change as well as "ghosts" left by removed objects.

The algorithms for detecting static foreground objects could be classified into four categories. The first category performs tracking on foreground blobs and detects static ones by analyzing tracks [1, 8, 9]. When scene is crowded, however, tracking-based approaches are not effective. The second category utilizes mode switching in GMM to identify static foreground pixels [3, 7]. Nevertheless, as GMM only looks at individual pixels, it often generates highly fragmented static foreground masks. The third category accumulates foreground mask for each pixel [4, 10]. However, they could not handle internal motion of a non-static object. The fourth category detects static foreground pixels by comparing a long-term foreground map with a short-term one [6, 2, 11]. Yet they are not able to prevent false alarms generated by partially static objects.

In fact, the information from the analysis on a scale beyond pixels and patches could be the key to breaking the bottlenecks of most existing methods. In this paper, we propose to perform region-level analysis in both background maintenance and static foreground object detection. More specifically, we adaptively update the background estimate by defining a "foregroudness" score for each pixel according to the global properties of the blob the pixel belongs to. When detecting static foreground objects, we define an "abandoness" score for each pixel, and each candidate abandoned blob is subject to further region-level analysis to eliminate potential false alarms. By introducing region-level analysis in both the two components, our method is robust against illumination change, "ghost" effects, distractions from partially static objects, and occlusions.

The remainder of this paper is organized as follows. In Section 2, we describe adaptive background maintenance using foregroundness scores. Static foreground object detection is detailed in Section 3. We present experimental results in Section 4, and Section 5 concludes this paper.

2. BACKGROUND MAINTENANCE

The overall structure of our background maintenance component is illustrated in Figure 1. In what follows, we describe in detail how to perform hybrid differencing between frame and background images, and how to evaluate region-level foregroundness scores to guide adaptive background learning.

2.1. Hybrid differencing

In order to generate a preliminary foreground map, we need to eliminate those pixels that could be regarded as in the background for a high certainty. For each pixel, we first compute its color difference D_C between the frame and the current background. Color difference is simply the maximum absolute difference of the pixel value over three color channels. To make our approach robust against local illumination change, we further compute the structural difference of the local patch around the pixel. Here, an issue often overlooked by prior arts is that part of the original structure might disappear due to the change in lighting conditions, as is illustrated in Figure 2. In this case, conventional measures like normalized

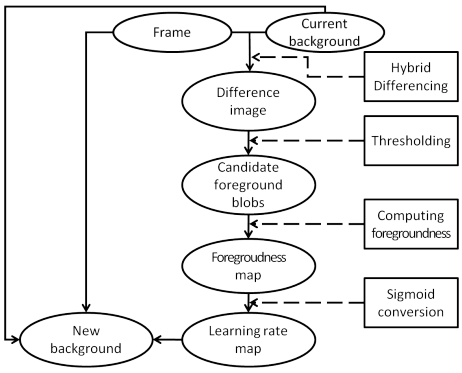

Fig. 1. The overall structure of our background maintenance component.

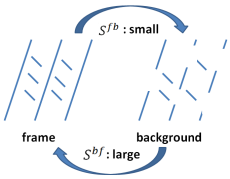

Fig. 2. The two patches have similar structures, yet conventional measures would yield a lower similarity score due to the missing of some structures in the background patch.

correlation would give a undesirably low similarity score. To solve this problem, we first match the structures of the frame patch to the background patch. As some of the structures of the frame patch cannot find a good match in the background patch, the similarity score is low. Then we perform the matching the other way round. This time, most of the structures of the background patch could find a good match, resulting in a high similarity score. The final structural similarity score is the larger of the two.

In implementation, structural matching from one patch (say frame patch) to the other one (say background patch) is conducted as follows. Firstly, for each pixel in the frame patch, its gradient orientation is compared with a neighborhood of pixels in the background patch, and the largest agreement score over the neighborhood is taken as the matching score for the pixel:

$$s_i^{fb} = \max_{j \in \mathbf{N}(i)} \{(1 - \frac{|\theta_i^f - \theta_j^b|_A}{\pi/2}) \cdot \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})\}, \quad (1)$$

where s_i^{fb} is the matching score for pixel i from frame to background, θ_i^f (or θ_j^b) is the gradient orientation of pixel i (or j) in the frame (or background) patch, $|\theta_i^f - \theta_j^b|_A$ is the acute angle formed by θ_i^f and θ_j^b, $\mathbf{x}_i$ and $\mathbf{x}_j$ are the coordinates of pixels i and j, respectively, and $\mathbf{N}(i)$ indicates the neighborhood of pixel i. The purpose of the neighborhood is to add robustness against small spatial displacement. The frame-to-background structural similarity score S^{fb} is the gradient-magnitude-weighted average of the matching scores of individual pixels: $S^{fb} = (\sum_{i \in \mathbf{R}} g_i^f \cdot s_i^{fb})/(\sum_{i \in \mathbf{R}} g_i^f)$, where g_i^f is the gradient magnitude of pixel i in the frame patch, and $\mathbf{R}$ denotes the patch region. The background-to-frame structural similarity S^{bf} is computed likewise.

The final structural similarity score between the frame and image patches is $S = \max\{S^{fb}, S^{bf}\}$, and the structural difference $D_S = 1 - S$.

After we obtain the color difference D_C and the structural difference D_S, the hybrid difference $D = D_C \cdot D_S$. In other words, the hybrid difference is small when any one of the two differences is low.

Fig. 3. The candidate foreground blob corresponding to a "ghost" is highlighted by the red ellipse. Although the edge energy along the blob contour in the frame image is high, the support (which takes into account both gradient magnitude and orientation) is much lower than in the background image.

2.2. Foregroundness and adaptive learning rate

The difference image resulted from hybrid differencing is thresholded, upon which connected component analysis is performed to generate candidate foreground blobs. As so far only local information has been used, many candidate foreground blobs are actually false detections due to "ghost" effects and/or illumination change not handled by hybrid differencing. Therefore, we need to further verify the "foregroundness" of each candidate foreground blob.

Many approaches have been proposed to identify "ghosts" left my removed objects. For example, dual foreground maps are employed in [6]. Other researchers compare color information within and outside the candidate foreground blob [4, 3]. Another approach compares the edge energy along the blob contour within the frame and background images [9, 7]. However, as most of the existing methods do not take into account edge directions, they are not effective when background is cluttered. In our approach, we compute the support that the blob contour gains from the gradient of the frame and background images. If the blob corresponds to a "ghost", then the blob contour would obtain a greater gradient support from the background image than from the frame image, as is illustrated in Figure 3. Here, support is large only when gradient magnitude along the contour is large *and* gradient orientation agrees with contour orientation. Formally, the support from the frame image for each pixel along the blob contour is

$$C_i^f = \max_{j \in \mathbf{N}(i)} \{\langle \mathbf{c}_i, \mathbf{g}_j^f \rangle \cdot \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})\}, \quad (2)$$

where C_i^f is the support of pixel i from the frame image, $\mathbf{c}_i$ is the contour normal at pixel i, $\mathbf{g}_j^f$ is the gradient vector of pixel j in the frame image, and $\langle \cdot, \cdot \rangle$ denotes inner product. The total contour support from the frame image is the average support along the entire contour: $C^f = (\sum_{i \in \mathbf{B}} C_i^f)/|\mathbf{B}|$, where $\mathbf{B}$ represents the blob contour. The contour support C^b

Fig. 4. The background estimated using our approach is less noisy and preserves more true details than using GMM. Note the tile boundaries on the floor and the bench area. The black regions on the floor are estimated shadows.

from the background image is computed likewise. To compare C^f and C^b, we use the normalized difference defined as $d = (C^f - C^b)/(C^f + C^b)$. The normalized difference is further turned into a foregroudness score F_C of the blob as follows: $F_C = 1/(1 + \exp(-w_C d))$, where a more positive difference leads to a higher foregroudness score, and w_C controls how sensitive the latter is with respect to the former.

In addition to "ghosts", we also need to check if a candidate foreground blob is caused by illumination change. Although the hybrid differencing could handle some illumination change, yet its analysis scale is still rather local. Therefore, we need to compute the structural similarity score on the entire blob. The computation of the structural similarity score is exactly the same as is described in Section 2.1, except that $\mathbf{R}$ is the entire blob now. Denoting the structural similarity score of the blob as S_B, we could derive another foregroundness score F_B as $F_B = 1/(1 + \exp(w_S(S_B - t_S)))$, where a higher S_B results in a lower F_B, and w_S and t_S adjusts the sensitivity and offset of the conversion, respectively.

The final foregroundness score F is taken as the minimum of F_C and F_B, because the drop of any one of them indicates the candidate foreground blob might be false. The foregroundness scores of all the other pixels that do not belong to any candidate foreground blobs are zero.

Having obtained the foregroundness scores, we use them to guide the maintenance of background by adaptively adjusting the learning rate of each pixel according to its foregroundness score. We compute the learning rate using a sigmoid conversion: $\lambda_i = \lambda_0/(1 + \exp(w_F(F - 0.5)))$, where λ_i is the learning rate for pixel i, λ_0 controls the overall learning speed. Here, a higher foregroudness score would result in a lower learning rate, and w_F adjusts the sensitivity. The background estimate b_i at pixel i is then updated as $b_i := (1-\lambda_i)b_i + \lambda_i f_i$, where f_i is the pixel value in the frame image.

The benefit of using region-level analysis in background maintenance is shown in Figure 4 where the estimated background using our approach is far less noisy than using the conventional GMM algorithm [3]. Also, true details in the background are better preserved in our approach.

3. STATIC FOREGROUND OBJECT DETECTION

Given a background estimate, we detect static foreground objects as is illustrated in Figure 5. At each frame, we compute, for each pixel, an instant "abandoness" score a_i, defined as $a_i = F_i \cdot \delta[\Delta f_i < t]$, where F_i is the region-level foregroudness score at pixel i, Δf_i is the difference in pixel value between the current and previous frames, and t is a threshold. In other words, the instant abandoness score is higher when the pixel has a higher foregroudness score *and* its pixel value remains almost unchanged between adjacent frames. The instant abandoness score is accumulated over time to yield the (accumulative) abandoness score A_i using the updating equation: $A_i := (1-\eta)A_i + \eta a_i$, where η is the updating strength which determines how long an object has to stay static before it is regarded as being abandoned. The accumulative nature of the abandoness score makes the algorithm robust against occlusions. After we obtain the abandoness score for every pixel, the abandoness map is thresholded and connected component analysis is performed to generate candidate abandoned blobs. (In what follows, we use "static foreground object" and "abandoned object" interchangeably.)

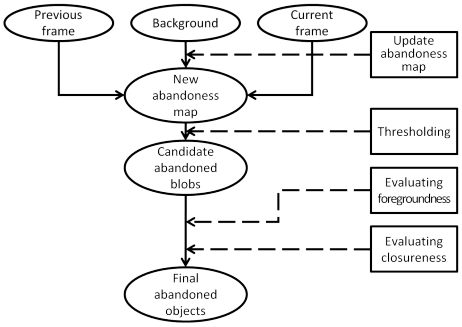

Fig. 5. The overall structure of our static foreground object detection component.

Fig. 6. Part of the body of the person sitting on the bench has been static for an extended period and therefore the abandoness scores in that region have exceeded the threshold and formed a candidate abandoned blob, as is indicated by the cyan region. However, as the person is wavering her upper body, the abandoness scores decay gradually on the upper contour of the cyan region. By contrast, the red candidate abandoned blob occupied by the suitcase has a sharp decrease in abandoness score along the entire blob contour.

As we use a soft foregroundness score which leverages region-level information, our approach is more robust than hard foreground mask accumulation [10]. As a double check, we evaluate the foregroudness score on each candidate abandoned blob as is described in Section 2.2, and it is rejected from being an abandoned object if its foregroundness score is lower than 0.5.

However, it is possible that a non-abandoned foreground blob (like a person) is *partially* static for an extended period. In this case, the static part would become a candidate abandoned blob which has a rather high foregroudness score. An

Sequence Name	Frames	Groundtruth Events	True Detections	False Detections	Static Persons
AVSS_AB_Easy	5,222	1	1	0	0
AVSS_AB_Medium	4,583	1	1	0	0
AVSS_AB_Hard	5,060	1	1	0	0
AVSS_PV_Easy	5,040	2	2	0	0
AVSS_PV_Medium	3,497	1	1	0	0
AVSS_PV_Hard	4,110	1	1	0	0
AVSS_PV_Night	5,848	1	1	0	0
ABTEA101a	101,748	18	15	3	7

Fig. 7. The performance of our algorithm on 8 i-LIDS video sequences.

example is illustrated in Figure 6. To prevent such false detections, we further evaluate the "closureness" score of each remaining candidate abandoned blobs. The closureness score measures the decrease rate of the abandoness score at locations near the blob contour. More specifically, for each pixel i on the contour, we compute a gradient score

$$h_i = \max_{j \in \mathbf{N}(i)} \{g_j^A \cdot \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)\}, \qquad (3)$$

where g_j^A is the gradient magnitude of the abandoness score map at pixel j. The closureness score h of the blob is the mean of the M pixels with the least gradient scores: $h = (\sum_{i=1}^{M} \tilde{h}_i)/M$, where $\tilde{h}$ are sorted gradient scores. If h is too low, the candidate abandoned blob is most likely a static part of a wavering object, and is therefore discarded.

After the elimination based on foregroudness and closureness, the remaining candidate abandoned blobs are regarded as abandoned objects, and alarms are triggered.

4. EXPERIMENTAL RESULTS

We evaluate our algorithm on i-LIDS dataset [12]. In addition to the clips for AVSS 2007, we also tested on a challenging sequence "ABTEA101a". The total number of frames under evaluation is nearly 130,000. All the parameters are fixed throughout our experiment.

The results are shown in Figure 7. Note that for clip ABTEA101a, 10 non-abandoned-object alarms are generated. Yet 7 of them are actually for static persons which, to our algorithm, are indistinguishable from static baggages, as we do not apply an appearance-based classifier on top of detected blobs for the sake of genericity.

It is observed that region-level analysis plays a major role in identifying false foreground and false abandoned blobs. An example of how the region-level analysis provides robustness for our algorithm are shown in Figure 8. Due to the limit in space, we are not able to include an example showing the robustness of our approach to occlusion, yet we note that occlusion never causes a problem throughout our experiment.

5. CONCLUSION

In this paper, we propose to detect abandoned objects using region-level analysis in both background maintenance and static foreground object detection. Attributed to much otherwise unavailable information provided by such analysis, our method achieves superior detection accuracy of abandoned objects on extensive real-world surveillance videos.

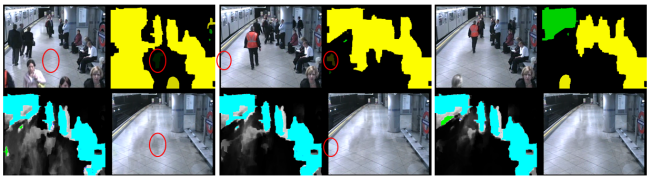

Fig. 8. Example of the robustness of our approach. The four panels (clockwise) in each image are the original frame, the foregroundness map, the current background estimate, and the abandoness map overlaid with candidate abandoned blobs. In the foregroundness map, a darker shade indicates a lower foregroundness score, and the color illustrates if F_C (green) or F_B (yellow) is lower. In the abandoness map, cyan and green colors indicate the blob is rejected because of wavering parts and low foregroundness, respectively. Note how the "ghost" of shadow in the left image and the illumination change in the middle image (highlighted by red ellipses) are identified in those images and completely removed in the right image. Also note all those standing and sitting persons, although being there for a long while, are not mistakenly regarded as abandoned objects.

6. REFERENCES

[1] H.H. Liao, J.Y. Chang, and L.G. Chen, "A localized approach to abandoned luggage detection with foreground-mask sampling," in *IEEE Int'l Conference on Advanced Video and Signal Based Surveillance*, 2008, pp. 132–139.

[2] X. Li, C. Zhang, and D. Zhang, "Abandoned objects detection using double illumination invariant foreground masks," in *IEEE Int'l Conference on Pattern Recognition*, 2010.

[3] Y. Tian et al., "Robust detection of abandoned and removed objects in complex surveillance videos," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. PP(99), pp. 1–12, 2010.

[4] J. Wen et al., "Generative model for abandoned object detection," in *IEEE Int'l Conference on Image Processing*, 2009, pp. 853–856.

[5] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR*, 1999, pp. II–2246–2252.

[6] F. Porikli, Y. Ivanov, and T. Haga, "Robust abandoned object detection using dual foregrounds," in *EURASIP Journal on Advances in Signal Processing*, 2008.

[7] Y. Tian, M. Lu, and A. Hampapur, "Robust and efficient foreground analysis for real-time video surveillance," in *CVPR*, 2005, vol. 1, pp. 1182–1187.

[8] S. Ferrando et al., "A new method for real time abandoned object detection and owner tracking," in *IEEE Int'l Conference on Image Processing*, 2006, pp. 3329–3332.

[9] J.C. San Miguel and J.M. Martinez, "Robust unattended and stolen object detection by fusing simple algorithms," in *IEEE Int'l Conference on Advanced Video and Signal Based Surveillance*, 2008, pp. 18–25.

[10] A. Bayona et al., "Stationary foreground detection using background subtraction and temporal difference in video surveillance," in *IEEE Int'l Conference on Image Processing*, 2010.

[11] W. Wang and Z. Liu, "A new approach for real-time detection of abandoned and stolen objects," in *IEEE Int'l Conference on Electrical and Control Engineering*, 2010, pp. 128–131.

[12] i-LIDS Dataset for AVSS 2007, ftp://motinas.elec.qmul.ac.uk/pub/iLids.