EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Patch-based local histograms and contour estimation for static foreground classification

Alex Pereira[1*], Osamu Saotome[1] and Daniel Sampaio[2]

Abstract

This paper presents an approach to classify static foreground blobs in surveillance scenarios. Possible application is the detection of abandoned and removed objects. In order to classify the blobs, we developed two novel features based on the assumption that the neighborhood of a removed object is fairly continuous. In other words, there is a continuity, in the input frame, ranging from inside the corresponding blob contour to its surrounding region. Conversely, it is usual to find a discontinuity, i.e., edges, surrounding an abandoned object. We combined the two features to provide a reliable classification. In the first feature, we use several local histograms as a measure of similarity instead of previous attempts that used a single one. In the second, we developed an innovative method to quantify the ratio of the blob contour that corresponds to actual edges in the input image. A representative set of experiments shows that the proposed approach can outperform other equivalent techniques published recently.

Keywords: Abandoned and removed object detection; Video surveillance; Video segmentation

1 Introduction

Video surveillance techniques for abandoned and removed object detection have received great attention in the last few years. Detecting suspicious objects is a central issue in the protection of public areas, such as airports, shopping malls, parks, and other mass-gathering areas.

In such applications, a sequence of computer vision methods is applied. Some approaches identify foreground blobs by applying background subtraction methods and then use an object tracker to determine whether the blob is static or not.

Other approaches avoid object tracking methods due to its flaws under crowded scenes [1]. Some alternatives have been proposed. Bayona performed a survey on stationary foreground detection [2] and concluded that approaches based on sub-sampling schemes or accumulation of foreground masks assure the best results. One year later, Bayona proposed one static foreground detection technique based on a sub-sampling scheme that outperformed other efforts mentioned in his survey. A succession of improvements has been reported in [3] and [4]. Although the stationary foreground detection issue is far from exhausted, the present research work is not concerned with the approach applied to identify stationary foreground. Instead, the focus is on the classification of static foreground blobs as either an abandoned or removed object.

We use a well-known shared assumption described in [5]: *When a background object is removed from the scene, it is reasonable to assume that the area thus vacated will exhibit a higher degree of agreement with its immediate surroundings than before.*

Fitzsimons [6] provided a brief literature review and categorized the main mechanisms used to distinguish abandoned from removed objects into four groups: edge detection, histograms comparison, image inpainting, and region growing.

The edge detection and the histogram comparison approaches are of special interest to our research. An explanation of the other two categories can be found in [6].

The intuitive reasoning on the edge detection approach in [5], is that *placing an object in front of the background will introduce more edges to the scene around the object's boundaries, provided that the background is not extremely cluttered.*

The Sobel [5,7-12] and Canny [5,13-16] operators have been employed with greater frequency for this purpose

*Correspondence: alp@ita.br
[1] Instituto Tecnológico de Aeronáutica (ITA), Praça Mal. Eduardo Gomes, 50, São José dos Campos, BR, CEP 12.228-900
Full list of author information is available at the end of the article

than the SUSAN edge detector [17]. While edge energy was employed in [17-19].

Flaws on distinguishing abandoned from removed objects by their edges can occur when the hypothesis of not extremely cluttered background is not valid. Grigorescu [20] showed that when textures and the scale of objects are similar, a non-contextual edge detector, such as the traditional Canny operator, generates strong responses to the texture regions. Then, object contours can be difficult to identify in the output of such an operator.

In our approach the best results were achieved by combining the Sobel and SUSAN edges, which is more invariant to scale changes than the Canny operator (as reported in [21]).

Henceforth for a simpler notation, unless otherwise specified, neighborhood of foreground blobs means the corresponding neighbor region in the input frame, not in the foreground mask.

Color histogram comparison [15,22-24] is another intuitive manner to discriminate abandoned and removed objects. Researchers compare the color distributions of the interior and exterior neighborhood of foreground blobs. It makes sense to assume that if the internal and external neighbor regions are similar in color, then no object is likely to be present. The inverse is also likely to be true.

We found that the accuracy of histogram-based features relies on the choice (shape and size) of the regions to compare. Usually, a bounding box delimits the external region. However, as we show next, the color distribution comparison of whole multi-colored objects often generates wrong results.

Although both edges and histogram categories present drawbacks, we show in our results that they are complementary and an appropriate combination can take the best of both.

All these approaches rely on a hidden assumption that the foreground blob correctly outlines the objects' contour when a real object is present. Then, they define internal and external regions and extract data to compare one to the other. If the assumption fails, which often occurs, the outcome is a misleading comparison. Using bounding boxes that are smaller than the actual object and computing background pixels from the object color distribution are two examples of many possible mistakes.

Thus, the features we propose consider some degree of inaccuracy on the foreground blob. We argue that this care is essential to deal with several different video scenarios.

2 Description of the removed and abandoned blob classifier (RABC)

The first step is pre-processing each input image filtering noise and then evaluating the following two features in order to provide a reliable classification: F_h - patch-based local histogram similarity and F_c - *contour sampling*, detailed in Sections 2.2 and 2.3, respectively. The final classification using these features is detailed in Section 2.5.

2.1 Preprocessing

The artificial edges created by image compression, with the quantization of 8×8 macroblocks, are not among the edges we aim to detect. Image noise, such as noise due to sensor quality, is not of interest to our present work either.

A common low-pass filter blurs the edges while removing noise, which is inappropriate for our purpose. Tomasi [25] proposed a bilateral filtering, which smooths images while preserving edges by means of a nonlinear combination of nearby image values.

The bilateral filter uses two parameters. The geometric spread σ_d, where a large σ_d blurs more because it combines values from more distant image locations. The photometric spread σ_r, where pixels with values closer than σ_r to each other are mixed together and values more distant than σ_r are not.

We use $\sigma_r = 50$ and $\sigma_d = 20$. Figure 1 presents a sample of the bilateral filtering applied to the 1,364th frame of the Highway test case of the CDW 2014 dataset [26].

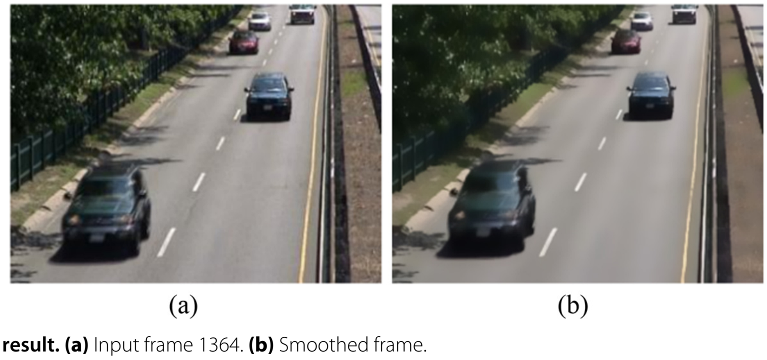

(a) (b)

Figure 1 Bilateral filtering result. (a) Input frame 1364. **(b)** Smoothed frame.

The following presented features benefit from using smoothed input data, mainly the histogram-based feature, which computes the difference between the color distribution of two regions.

2.2 F_h - patch-based local histogram similarity

Figure 2, taken from the AB_Easy test case of the AVSS 2007 dataset [27], illustrates the intuitive reasoning of this feature. The blob from Figure 2b appears because of a bootstrap on the background model. In other words, in the first frame of the video (which was used to initialize the background model), there was a man walking. Some frames after that, the input frame (523rd) brings the uncovered background. Figure 2a presents a piece of the 523rd background frame. Figure 2b presents a segmentation where the foreground blob represents a removed object and Figure 2c shows the blob boundary projected over the corresponding input frame.

In Figure 2c, we note considerable similarity between internal and external regions of the blob. In several instances of these removed objects, the color of the external neighborhood is similar to the color of the neighborhood inside the corresponding blob. We measured this similarity by comparing the histogram of internal and external neighbor regions.

We used the multi-color observation model, by Perez [28], based on hue saturation value (HSV) color histograms. This color histogram is more accurate than a grayscale one. Our technique uses the Kolmogorov-Smirnov test [29] as a metric to evaluate the similarity between the two histograms. Blobs corresponding to objects that differ from their neighbor region are unlikely to be classified as removed ones.

Up to this point, our proposed technique and previous ones are fairly equivalent. However, previous approaches did not tackle situations where the region behind a blob is not as homogeneous as in the example of Figure 2. For such situations, we proposed a novel approach, inspired in [29], to split the image into patches and to analyze whether each patch is homogeneous. This is discussed in Section 2.2.1.

2.2.1 Improving the similarity assessment

Color histograms can distinguish one object from another when their color distributions are distinct. However, color histograms do not differentiate objects with similar distributions but with different color locations. For example, suppose two 2×2 chess boards rotated 90° from each other. A simple color histogram comparison would evaluate that they are the same. As explained in [29], an appropriate approach would be to divide the object into regions (patches) and consider their histograms in order to take a more precise observation model of the object.

Briefly, the overall color distribution of two images might be similar, while the comparison of color distribution taken from lower scale pieces might tell us that the images are different. Lower scale pieces provide more accurate data. Therefore, the issue is how to determine the scale and shape of the pieces. In the following, we explain our method to get local color distributions.

We created rectangular patches by dividing the bounding boxes into $N \times N$ grids. The number of rows and

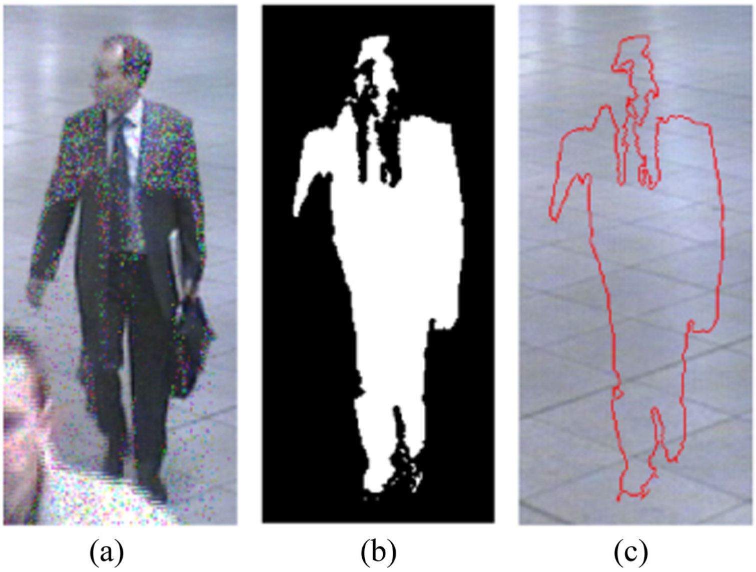


Figure 2 Removed object (person) sample. (a) Background model of frame 523. **(b)** Segmentation mask. **(c)** Input frame 523 with blob boundary.

columns N of the grid is adaptively defined according to the bounding box area A_{bb} and a goal patch area A_{g}, see Expression 1. This is the expression for a canonical example of a squared bounding box and works for rectangular ones as well:

$$\max\left(\sqrt{\frac{A_{\mathrm{bb}}}{A_{\mathrm{g}}}}, 1\right) \tag{1}$$

Perez [28] proposed a color histogram with 110 bins. We use the same number of bins. The number of pixels in a patch must be representative in order to get plausible quality histograms. Then, the minimum goal patch area of 300 pixels showed to be suitable.

We use a bounding box extended by 25% in area (50% in each dimension) compared to the tight bounding box of the blobs. This is necessary to get enough pixels from the external blob neighborhood. Then, from this point on, we consider that bounding box means the extended one.

We perceived that the relative position of the patches to the whole blob can affect the similarity measure. Then, we gather in a single set the patches from grids of size $N - 1$, N and $N + 1$. Some of the patches are disregarded as explained below.

The purpose here is to evaluate the color similarity in the neighborhood of the blob contour. So, only patches that cover the blob contour are used. Each patch comprises two regions, internal and external. Then, we disregard patches in which any of these regions have an area smaller than 15% of the patch area. Very few pixels cannot form representative color distributions.

Figure 3 presents the patches that cover the blob contour. This example was based on the Traffic test case from the CDW 2012 dataset [26], and the foreground mask was taken from [30]. This figure shows that using three grid sizes, we can cover a larger portion of the blob contour. Thus, the comparison accuracy does not depend on a manual selection of patch sizes.

Next, for each patch, we compare the internal and external patch regions with Kolmogorov-Smirnov test [29] as a difference metric. In order to extract the whole similarity, we could take the mean from all the differences. However, this is more appropriately modeled as a voting problem. Each patch gives valuable information about its area. No matter how close its similarity is to 100%, it must not contribute to the similarity of other patches as it would contribute by calculating a simple average. Figure 3 presents such an example. Among 22 squared patches, there are five patches that cover a wrong segmentation area (homogeneous area covering the road) and the voting scheme is able to correctly classify the blob. The following equations show the related calculations.

Consider the Kolmogorov-Smirnov test, represented by the function $\mathrm{KS}(h_i, h_e)$ which produces a real number

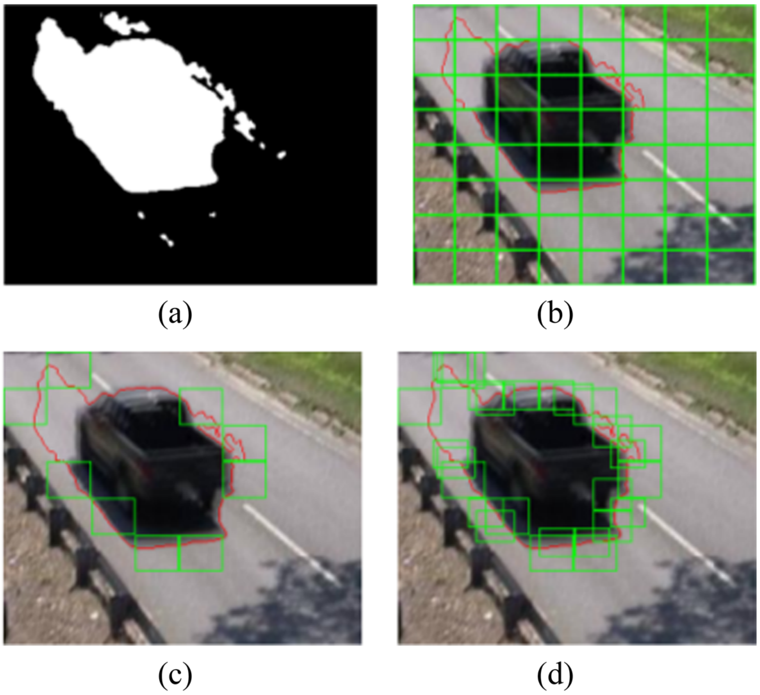


Figure 3 Histogram patches of an actual object. (a) Segmentation mask. **(b)** One complete grid. **(c)** Selected patches from one grid. **(d)** Selected patches from three grids.

in the range [0,1] corresponding to the absolute difference between two histograms, h_i and h_e. Equations 2 and 3 present the procedure to evaluate the feature F_h. The symbol τ_h is the similarity threshold, and P_a *is the number of non-disregarded patches*:

$$D(j) = \begin{cases} 1, & \text{if } \mathrm{KS}(h_i, h_e) > \tau_h \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$F_h = \frac{\sum_{j=1}^{P_a} D(j)}{P_a} \tag{3}$$

The feature value is the ratio of patches that have similar internal and external regions.

2.3 F_c - contour sampling

We developed a method to determine whether a blob region is surrounded by edges or not. The method detects the edges in the neighborhood of the blob border and evaluates the portion of the contour that is surrounded by edges. We consider that a closed/almost closed contour corresponds to an abandoned object.

Consider the sample in Figure 4 taken from Cam1 test case of the Hermes Dataset [31]. The sequence starts with a static car parked in the street (Figure 4a). After sometime, it moves and uncovers the background (Figure 4b). This situation produces a ghost because the true background data was always unavailable. At frame 1,396, the segmentation process produces two blobs with the same shape and size of the car (Figure 4c). The blob in the left represents the initial position of the car that should be classified as removed object. The other blob represents the car in frame 1,396.

Figure 5 presents the sequence of operations performed to detect the removed object. Figure 5a presents the piece of the 1,396th input frame where the car was initially parked. Figure 5b presents the corresponding foreground mask. Figure 5c presents the edges detected as explained in Section 2.4. Figure 5d shows the internal and external neighborhood of the blob border, obtained from the difference of the dilated convex hull [32] and the eroded foreground blob, henceforth referred as crown. Figure 5e presents a binarization of the edges that lie inside the crown region.

We developed a monotonic function that quantifies the ratio of the object contour found by the edge detector in the neighborhood of the blob boundary. We call this function as contour sampling.

A geometric operation of intersecting a straight line at several (and possibly equally spaced) points of the contour can fulfill the monotonic requirement. Tracing concentric straight lines, from a point inside the contour, can perform the underlying procedure. Each line is rotated from the previous by an angle of some degrees.

Figure 5f shows the picked edges, the source point in green, the straight lines, and blue points representing the intersection. In this example, 60% of the lines intersected the edges.

In case of blobs with a complex shape, for example a U-shaped blob, a single source point is not enough to sample the whole contour because, for simplicity, we take only the first intersection point.

Then, we use several source points spaced throughout the blob region. For this, we take N_S points from the Sobol sequence [33]. This sequence is a solution to the problem of filling an area uniformly with quasi-random points.

N_S is calculated with Expression 1, setting A_g to 25 pixels. Thus, a quasi-random point is likely to be at each 5×5 piece of the bounding box.

Equation 4 is used to calculate a ratio considering the source points that lie in the black area inside the crown contour. In this equation, I_s stands for the number of intersections derived from the source point s. L represents the number of lines of each source point. Finally, the ratio is reversed to represent the missing portion of the contour:

$$F_c = 1 - \frac{\sum I_s}{N_S \times L} \tag{4}$$

As the number of traced lines increases, the value F_c approaches the actual percentage of missing contour out of the 360°. We use $L = 30$ lines for each source point, which yields a precise measurement.

Figure 6 presents an analysis of the piece of the input frame 1,396 where the car blob appears. In this example,

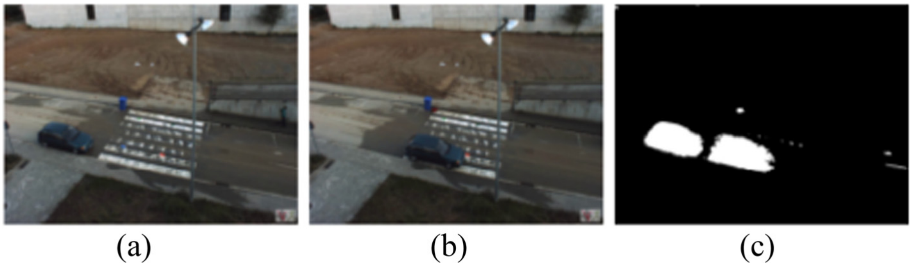


Figure 4 Two blobs: the actual car and the ghost. (a) Input frame 1. **(b)** Input frame 1,396. **(c)** Foreground mask.

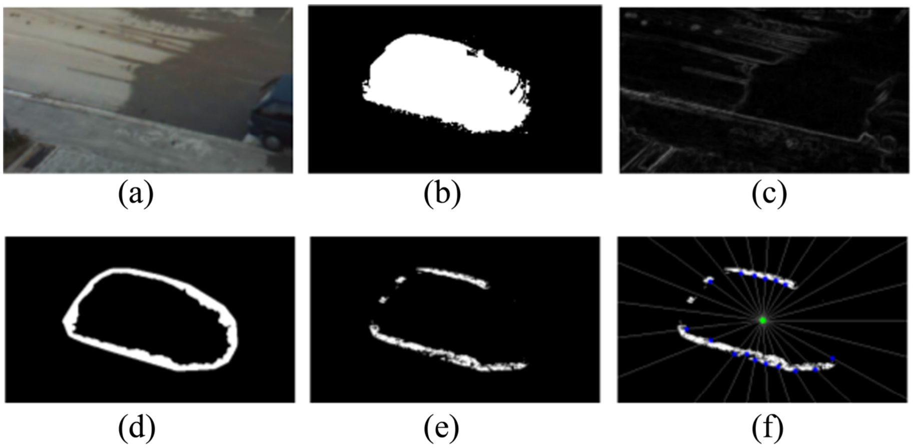


Figure 5 Removed object sample. (a) Piece of frame 1,396. **(b)** Foreground mask. **(c)** Edges mask. **(d)** Dilated hull/eroded foreground. **(e)** Picked edges. **(f)** Removed object contour.

86% of the lines intersected the edges. In Figure 6f, we used only two source points to simplify the presentation.

We assume that all blobs have complex shapes and always use multiple source points. This feature can identify the removed object because it is extremely unusual to find edges around the whole blob that corresponds to a removed object.

2.4 Finding the edges

We extracted the edges of each RGB channel with the SUSAN detector and the edges of the luminance (grayscale) channel with the Sobel operator and combined their results into one edge mask.

We chose the SUSAN detector because it is more invariant to scale changes than other non-contextual edge detectors [21].

Using only the luminance Y (ITU-R BT.601), as in the original experiments of SUSAN [21], is not appropriate because there are many edge samples that do not appear on the luminance channel, but only on the chrominance channels. For example, two neighboring pixels with the same luminance, but opposite extreme values of chrominance show no edges on the luminance channel.

The SUSAN detector relies on a threshold t that determines the minimum contrast of edges that will be picked up. We use a fixed threshold $t = 15$, which sometimes yields missing some edge pixels.

Using Sobel with a dynamic binarization threshold complements the SUSAN edge mask. The Sobel threshold τ_c is defined in Equation 5:

$$\tau_{sb} = \mathrm{MAX}(10, (\mathrm{mean} + 0.25 \times \mathrm{std\_dev})) \tag{5}$$

In Equation 5, mean stands for the mean of the Sobel gradient and std_dev the corresponding standard deviation. The support at 10 is needed to not pick almost dark Sobel pixels from gradient of homogeneous images.

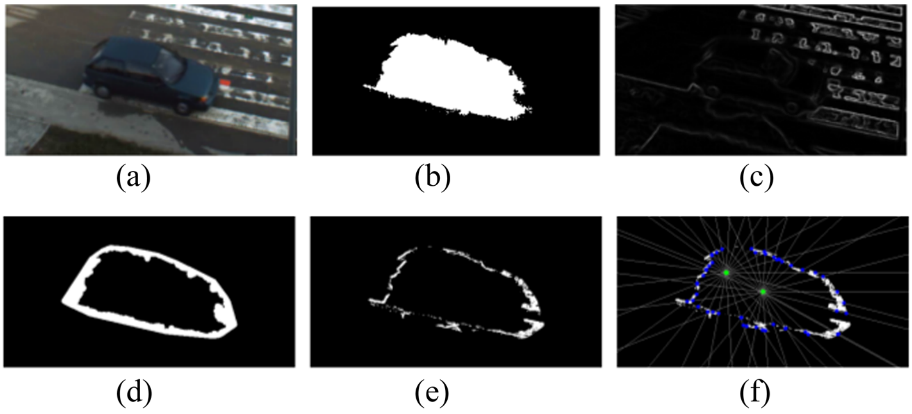


Figure 6 Actual object sample. (a) Piece of frame. **(b)** Foreground mask. **(c)** Edges mask. **(d)** Dilated hull/eroded foreground. **(e)** Picked edges. **(f)** Actual object contour.

The combination of the edges masks E is performed with a logic OR as shown in Equation 6:

$$E(R,G,B,Y) = \mathrm{SUSAN}(R,t) \ \cup \ \mathrm{SUSAN}(G,t) \ \cup \ \mathrm{SUSAN}(B,t) \ \cup \ \mathrm{Sobel}(Y,\tau_{\mathrm{sb}}) \quad (6)$$

2.5 Combining the two features

The target set (codomain) of both features is [0,1]. First, we evaluate each feature at the input frame ($F_h(\mathrm{In})$ and $F_c(\mathrm{In})$) and at the background model ($F_h(\mathrm{Bg})$ and $F_c(\mathrm{Bg})$). A high value of input frame features indicates that the blob is likely to correspond to a removed object. A low value indicates an abandoned one. The inverse is also true for the background features.

We subtract the input and background features; see Equations 7 and 8. The resulting sign is used as a binary rating, and the absolute value represents the corresponding confidence. This approach avoids the infeasible task of finding a single threshold to determine whether the feature values correspond to one or another classification:

$$\mathrm{Sub}_h = F_h(\mathrm{Bg}) - \mathrm{F_h}(\mathrm{In}) \quad (7)$$

$$\mathrm{Sub}_c = F_c(\mathrm{Bg}) - \mathrm{F_c}(\mathrm{In}) \quad (8)$$

A negative value of the subtraction (Sub_h or Sub_c) indicates that it is likely to exist an object in the background model and do not in the input frame, i.e., the background model does not correspond to the reality and the referred blob is a removed one. While a positive value indicates that the object is likely to be in the input frame and do not in the background model, i.e., an abandoned object.

Equation 9 models the aforementioned reasoning. Here, the underlying idea is to pick the classification of the most confident feature. If both subtractions agree in sign, the chosen Class is the corresponding class of that sign (removed for negative values). If the subtractions disagree in sign, the most confident is chosen:

$$\mathrm{Class} = \begin{cases} \text{removed}, & \text{if } \mathrm{Sub}_h + \mathrm{Sub}_c < 0 \\ \text{abandoned}, & \text{otherwise} \end{cases} \quad (9)$$

3 Experimental results

One advantage of the proposed technique is that it is quite autonomous. It relies on two parameters τ_h and t, one for each feature. The threshold τ_h is set to 0.99. In our experiments, lower values of τ_h produced undesirable false positives. Smith in [21] suggests a value between 10 and 20 to SUSAN threshold t. We set it to 15.

The classifier uses three input data: the input frames, a foreground mask, and the corresponding background model frame.

In the first experiment, we used the ASOD [34] dataset comprised of input frames, a background frame, and the corresponding ground truth (manually annotated and automatically generated inaccurate masks) of static foreground from PETS2006 [35], PETS2007 [36], AVSS2007 [27], CVSG [37], VISOR [38], CANDELA [39], and WCAM [40]. We call the manually annotated ground truth as the annotated subset, and the automatically generated masks as the real subset. The amount of blobs in both subsets is shown in Table 1.

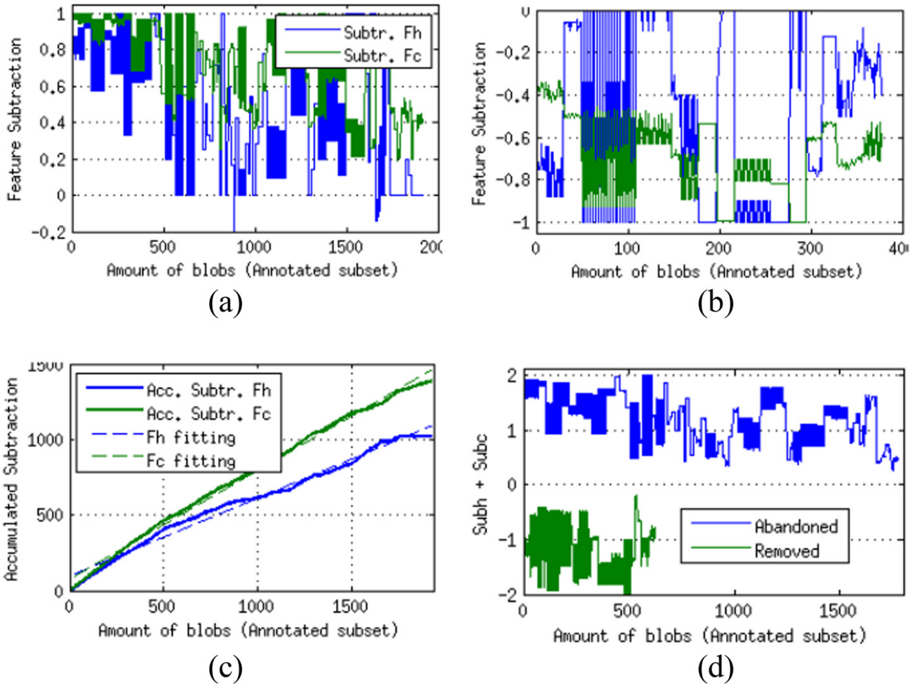
We achieved 100% of accuracy classifying the blobs from the annotated subset (second and third column of Table 1) as either abandoned or removed. Fitzsimons [6] also achieved 100% of accuracy in the same subset. There are some reasons that we achieved a flawless result. The dataset provided canonical background frame and an annotated foreground mask. The background is a frame taken from the sequence where the only change is the presence or the absence of the object under analysis. The manually annotated foreground blobs tightly fit the border of the objects. This is the best scenario to evaluate the features. Although simple, this experiment is useful for the early validations.

The plots from Figure 7 give an overall view of this classification problem on the annotated subset. Figure 7a,b presents the Sub_h and Sub_c measures, for the abandoned and removed blobs, respectively. Note that the stepped aspect of the plots shows the beginning and ending of each scenario evaluation. The features are fairly complementary. In Figure 7a, their value alternately move away from 1, while in Figure 7b, they alternately move away from -1.

Figure 7c presents the accumulated value of Sub_h and Sub_c, and their corresponding best fitted lines (least square sense). An ideal feature would approach the line $x = y$, since in the abandoned scenarios, the subtractions Sub_h and Sub_c should always be 1. The slope of these lines are 0.61 and 0.72, for Sub_h and Sub_c, respectively. The

Table 1 ASOD dataset description

Category	Annotated		Real	
	Abandoned	Removed	Abandoned	Removed
C1	771	442	751	806
C2	574	264	497	353
C3	575	174	739	588
Total	1,920	880	1,987	1,747



(a) (b)

(c) (d)

Figure 7 Plots of the annotated subset. (a) Feature subtraction of abandoned blobs. **(b)** Feature subtraction of removed blob. **(c)** Accumulated subtraction of abandoned blobs. **(d)** Plot of $\mathrm{Sub}_h + \mathrm{Sub}_c$.

slope is a suitable way to compare the features, since it represents the trend of the feature plot. The conclusion here is that the feature F_c is more accurate than the feature F_h.

Finally, Figure 7d shows that the feature F_c can correctly classify the whole annotated subset. By Equation 9, any sum value ($\mathrm{Sub}_h + \mathrm{Sub}_c$) above zero is abandoned and below zero is removed. The margin is approximately the range [-0.2,0.2].

The next experiment refers to the real subset. It is more realistic as the masks are fairly inaccurate. We disregarded blobs with less than 50 pixels. Further, we removed from the experiment the test case called *AVSSS07 indoor abandoned object easy 4cif* (comprised in the second category) because it presents misclassified blobs. The classification accuracy on the real subset is reported in Table 2. In this table, TP stands for the number of true classified abandoned object, FP stands for the misclassified abandoned objects, TN stands for true classified removed objects, and FN misclassified removed objects. The sixth column presents the recall ($\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$), the seventh column presents the accuracy ($(\mathrm{TP}+\mathrm{TN})/(\mathrm{FP}+\mathrm{FN})$) of the proposed technique, and the last column presents the best accuracy results achieved by the creators of this dataset [14].

Our result is 3.7% more accurate than the results from [14]. We argue that this improvement is mainly due to: 1) the diversity of patch shapes that makes the histogram feature take into consideration (most of the times) suitable regions, 2) the contour feature searching for edges in the internal neighborhood of a blob and in the external neighborhood of the blob convex hull, 3) combining the SUSAN with Sobel edges in the contour feature, and 4) replacing fixed feature thresholds for dynamic ones.

The plots in Figure 8 give an overall view of this classification problem on the real subset. Figure 8a,b presents

Table 2 Results on the real subset

Category	Abandoned		Removed		RABC	RABC	[14]
	TP	FP	TN	FN	Recall	Accuracy	Accuracy
C1	751	0	806	0	100%	100%	96.7%
C2	495	2	352	1	99.8%	99.6%	94.3%
C3	708	31	588	0	100%	97.7%	95.1%
Total	1,954	33	1,746	1	99.9%	99.1%	95.4%

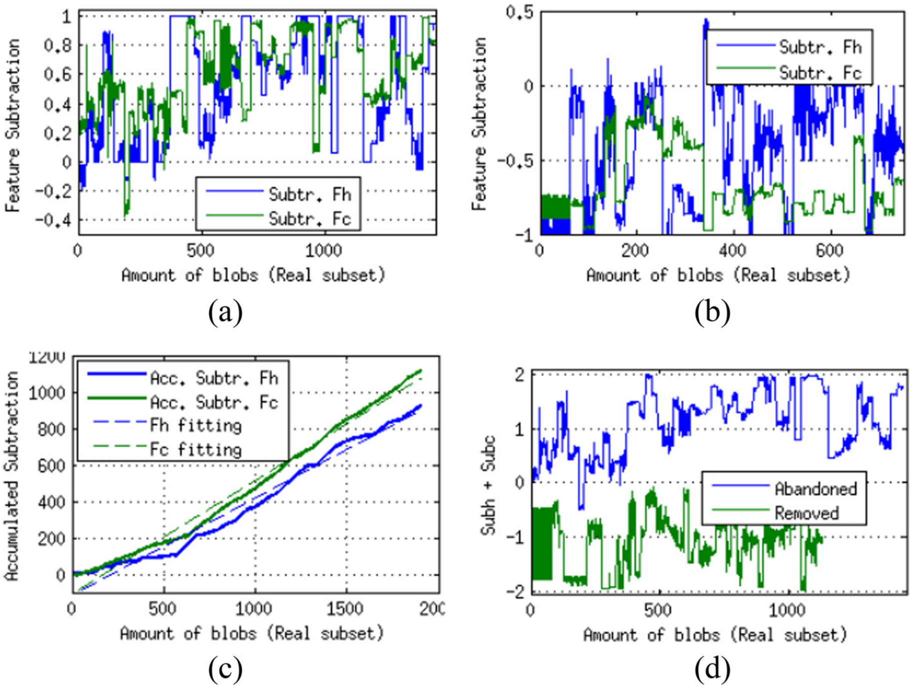


Figure 8 Plots of the real subset. (a) Feature subtraction of abandoned blobs. **(b)** Feature subtraction of removed blob. **(c)** Accumulated subtraction of abandoned blobs. **(d)** Plot of $\mathrm{Sub}_h + \mathrm{Sub}_c$.

the Sub_h and Sub_c measures, respectively, for the abandoned and removed blobs. These plots present a noisier appearance compared to the plots of Figure 7a,b. This appearance reflects the inaccuracy of the blobs from the real subset.

Figure 8c presents the accumulated value of Sub_h and Sub_c and their corresponding best fitted lines (least square sense). Here, we see that the classification problem is harder than the annotated one because the slope of the fitted lines is lower, 0.53 and 0.62, respectively, for Sub_h and Sub_c. The feature F_c is again more accurate than the feature F_h.

Finally, Figure 8d shows that neither the combination of the features could correctly classify the whole real subset. The mistakes were just 0.9% of the total, and the corresponding blobs barely resemble the annotated ones.

In the next experiment, we used the PETS2006 videos of the camera 3 from scenarios 1 to 7. A single event in each of these videos has been used for the accuracy evaluation on previous research [41-44]. All the seven events are abandoned bags.

In this experiment, we used the foreground mask produced with the SuBSENSE [45] segmenter. SuBSENSE does not maintain a single background model frame. Instead, it manages a set of samples for each pixel. Then, for each pixel, we extracted a background frame by choosing the sample that best fits each corresponding pixel from input frame and used it as a running average background model. This procedure was repeated for each input frame.

We correctly classified the blobs of these seven events as abandoned objects. Table 3 shows that we matched the performance of [41,43,44] and outperformed [42].

Gaetano [46] reported the detection of the blobs that appeared after the removal of the purple bins. We also classified these blobs as removed object ones.

We performed the experiments on a PC with an Intel(R) Core(TM) i5-3210M CPU @ 2.50 GHz. The performance on the PETS2006 dataset, with frames measuring 720×576, was 11 frames per second. The bilateral filter took 75% of the time to analyze each frame.

4 Conclusions

The main goal of the present research work is to develop a technique to classify static foreground blobs as abandoned or removed objects. The proposed technique, named as removed and abandoned blob classifier (RABC), is based

Table 3 Comparison on the PETS2006 dataset

Dataset	[41]	[42]	[44]	[43]	Proposed
PETS2006	6/6	6/7	6/6	7/7	7/7

on a widely used assumption that a removed region is similar to its neighborhood, while abandoned object regions usually have discontinuity, i.e., edges, defining their borders.

The RABC technique combines two features, derived from the aforementioned properties: 1) patch-based local histogram similarity and 2) contour sampling.

Both features were designed considering that some degree of inaccuracy is present in the input data. We argue that this care is essential for the classifier to deal with several different video scenarios. For example, combinations of edge operators, dynamic thresholds and patch sizes, and extended bounding boxes were designed based on this care.

The feature values are ratios in the range $[0, 1]$. Thus, the feature values can be understood as confidence values. The final classification compares the feature values extracted from the background with those extracted from the input frame. If the feature outcomes are the same (whether abandoned or removed), the final result is the agreed outcome. Otherwise, the most confident outcome between them is chosen. This procedure avoids the unfeasible task of defining suitable thresholds while achieving high accuracy.

The results showed that our proposed technique outperformed recent state-of-the-art techniques with the same purpose.

There is potential research that could build-on our work and our findings. One potential future work would be replacing the squared patches with superpixels in the patch-based local histogram feature. Superpixels describe image regions more precisely. Such change needs a metric like earth mover distance (EMD) metric to compare histograms. EMD has the capability of comparing two distinct sets of image pieces.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The authors would like to thank Mr. Quanfu Fan, from the Thomas J. Watson Research Center, for sharing his ground truth data and results of his research [44]. The authors would like to thank Mr. Jack Fitzsimons, from the University of Oxford, for confirming the actual number of annotated blobs on the ASOD dataset. This work has been supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

Author details

[1]Instituto Tecnológico de Aeronáutica (ITA), Praça Mal. Eduardo Gomes, 50, São José dos Campos, BR, CEP 12.228-900. [2]Universidade Estadual Paulista, Av. Ariberto Pereira da Cunha, 333, Guaratinguetá, BR, CEP 12.516-410.

Received: 19 September 2014 Accepted: 29 January 2015
Published online: 25 February 2015

References

1. F Porikli, Y Ivanov, T Haga, Robust abandoned object detection using dual foregrounds. EURASIP J. Adv. Signal Process. **2008**, 30 (2008)
2. Á Bayona, JC SanMiguel, JM Martínez, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS) (Genoa, 2009)*. Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques, pp. 25–30
3. D Ortego, JC SanMiguel, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS) (Krakòw, 2013)*. Stationary foreground detection for video-surveillance based on foreground and motion history images, pp. 75–80
4. D Ortego, JC SanMiguel, in *Proceedings of IEEE International Conference on Image Processing (ICIP)*. Multi-feature stationary foreground detection for crowded video-surveillance (Paris, 2014), pp. 2403–2407
5. A Singh, A Agrawal, in *Proceedings of IEEE India Conference (INDICON)*. An interactive framework for abandoned and removed object detection in video (Bombay, 2013), pp. 1–6. doi:10.1109/INDCON.2013.6725905
6. JK Fitzsimons, TT Lu, in *Proceedings of SPIE Conference on Applications of Digital Image Processing XXXVII, vol. 9217*. Markov random fields for static foreground classification in surveillance systems (San Diego, 2014), pp. 92171O–92171O-10
7. J Kim, A Ramirez Rivera, B Ryu, K Ahn, O Chae, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Unattended object detection based on edge-segment distributions (Seoul, 2014), pp. 283–288. doi:10.1109/AVSS.2014.6918682
8. T Kryjak, M Komorkiewicz, M Gorgon, Real-time foreground object detection combining the PBAS background modeling algorithm and feedback from scene analysis module. Int. J. Electron. Telecommunications. **60**(1), 53–64 (2014)
9. HH Liao, JY Chang, LG Chen, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. A localized approach to abandoned luggage detection with foreground-mask sampling (Santa Fe, 2008), pp. 132–139
10. W Hassan, P Birch, R Young, C Chatwin, in *Proceedings of SPIE Video Surveillance and Transportation Imaging Applications, vol. 8663*. An improved background segmentation method for ghost removals, (2013), pp. 86630W–86630W-6
11. I Huerta, A Amato, X Roca, J Gonzàlez, Exploiting multiple cues in motion segmentation based on background subtraction. Neurocomputing. **100**, 183–196 (2013)
12. M Magno, F Tombari, D Brunelli, L Di Stefano, L Benini, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Multimodal abandoned/removed object detection for low power video surveillance systems (Genoa, 2009), pp. 188–193
13. YJ Chai, SW Khor, YH Tay, in *Proceedings of SPIE International Conference on Digital Image Processing (ICDIP), vol. 8878*. Object occlusion and object removal detection (Beijing, 2013), pp. 88782F–88782F-5
14. J SanMiguel, L Caro, J Martinez, Pixel-based colour contrast for abandoned and stolen object discrimination in video surveillance. Electron. Lett. **48**(2), 86–87 (2012)
15. L Caro Campos, JC SanMiguel, JM Martínez, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Discrimination of abandoned and stolen object based on active contours (Klagenfurt, 2011), pp. 101–106
16. RH Evangelio, T Sikora, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Complementary background models for the detection of static and moving objects in crowded environments (Klagenfurt, 2011), pp. 71–76. doi:10.1109/AVSS.2011.6027297
17. P Spagnolo, A Caroppo, M Leo, T Martiriggiano, T D'Orazio, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. An abandoned/removed objects detection algorithm and its evaluation on pets datasets (Sydney, 2006), pp. 17–17
18. YL Tian, M Lu, A Hampapur, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1*. Robust and efficient foreground analysis for real-time video surveillance (San Diego, 2005), pp. 1182–1187
19. J Connell, AW Senior, A Hampapur, YL Tian, L Brown, S Pankanti, in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*. Detection and tracking in the IBM peoplevision system (Taipei, 2004), pp. 1403–1406
20. C Grigorescu, N Petkov, MA Westenberg, Contour and boundary detection improved by surround suppression of texture edges. Image Vision Comput. **22**(8), 609–622 (2004)
21. SM Smith, JM Brady, Susan-a new approach to low level image processing. Int. J. Comput. Vision. **23**(1), 45–78 (1997)
22. LF Tu, SD Zhong, Q Peng, Moving object detection method based on complementary multi resolution background models. J. Cent. S. University. **21**, 2306–2314 (2014)

23. B Hu, Y Li, Z Chen, G Xiong, F Zhu, in *Proceedings of IEEE International Conference on Intelligent Transportation Systems (ITSC)*. Research on abandoned and removed objects detection based on embedded system (Qingdao, 2014), pp. 2968–2971
24. S Ferrando, G Gera, C Regazzoni, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Classification of unattended and stolen objects in video-surveillance system (Sydney, 2006), pp. 21–21
25. C Tomasi, R Manduchi, in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. Bilateral filtering for gray and color images (Bombay, 1998), pp. 839–846
26. N Goyette, P Jodoin, F Porikli, J Konrad, P Ishwar, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Changedetection. net: a new change detection benchmark dataset (Providence, 2012), pp. 1–8
27. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), London (2007). http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html, September 2014
28. P Pérez, C Hue, J Vermaak, M Gangnet, in *Proceedings of Conference on Computer Vision (ECCV)*. Color-based probabilistic tracking (Springer Copenhagen, 2002), pp. 661–675
29. A Adam, E Rivlin, I Shimshoni, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Robust fragments-based tracking using the integral histogram (New York, 2006), pp. 798–805
30. R Wang, F Bunyak, G Seetharaman, K Palaniappan, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Static and moving object detection using flux tensor with split gaussian models (Columbus, 2014), pp. 414–418
31. J Gonzàlez, FX Roca, JJ Villanueva, in *Proceedings of Conference on Computational Vision and Medical Image Processing (VipIMAGE)*. Hermes: a research project on human sequence evaluation (Porto, 2007)
32. M De Berg, M Van Kreveld, M Overmars, O Cheong, *Computational Geometry: Algorithms and Applications*, 3rd ed. (Springer, Santa Clara, CA, USA, 2008)
33. P Bratley, BL Fox, Algorithm 659: Implementing Sobol's quasirandom sequence generator. ACM Trans. Math. Software (TOMS). **14**(1), 88–100 (1988)
34. JC San Miguel, JM Martínez, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Robust unattended and stolen object detection by fusing simple algorithms (Santa Fe, 2008), pp. 18–25
35. PETS 2006 Benchmark Data. http://www.cvg.reading.ac.uk/PETS2006/data.html, January 2015
36. PETS 2007 Benchmark Data. http://www.cvg.reading.ac.uk/PETS2007/data.html, January 2015
37. Chroma-based Video Segmentation Ground-truth (CVSG). http://www-vpu.eps.uam.es/CVSG, January 2015
38. R Vezzani, R Cucchiara, Video surveillance online repository (visor): an integrated framework. Multimedia Tools Appl. **50**(2), 359–380 (2010)
39. Candela - Surveillance. http://www.multitel.be/cantata, January 2015
40. WCAM - Surveillance. http://www.vpu.eps.uam.es/DS/ASODds/index.html, January 2015
41. A Lopez-Mendez, F Monay, JM Odobez, in *Proceedings of International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*. Exploiting scene cues for dropped object detection (Lisbon, 2014)
42. G Szwoch, Extraction of stable foreground image regions for unattended luggage detection. Multimedia Tools Appl., 1–26 (2014). doi:10.1007/s11042-014-2324-4
43. YL Tian, RS Feris, H Liu, A Hampapur, MT Sun, Robust detection of abandoned and removed objects in complex surveillance videos. Systems, Man, Cybernetics, Part C: App. Rev. IEEE Trans. **41**(5), 565–576 (2011). doi:10.1109/TSMCC.2010.2065803
44. Q Fan, P Gabbur, S Pankanti, in *Computer Vision (ICCV), 2013 IEEE International Conference On*. Relative attributes for large-scale abandoned object detection, (2013), pp. 2736–2743. doi:10.1109/ICCV.2013.340
45. PL St-Charles, GA Bilodeau, R Bergevin, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), Columbus, OH, USA, 2014*. Flexible background subtraction with self-balanced local sensitivity, (2014)
46. G Di Caterina, Video analytics algorithms and distributed solutions for smart video surveillance. PhD thesis, University of Strathclyde (2013)