Edge Noise Removal in Multimodal Background Modeling Techniques

J. W. Choi, S. Apewokin, B. E. Valentine, D. S. Wills, L. M. Wills
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332-0250

ABSTRACT

Traditional video scene analysis depends on accurate background modeling techniques to segment objects of interest. Multimodal background models such as Mixture of Gaussian (MOG) and Multimodal Mean (MM) are capable of handling dynamic scene elements and incorporating new objects into the background. Due to the adaptive nature of these techniques, new pixels have to be observed consistently over time before they can be incorporated into the background. However, pixels in the boundary between two colors tend to fluctuate more, creating false positive foreground pixels that result in less accurate foreground segmentation. To correct this, a simple and computationally efficient edge detection-based algorithm is proposed. On average, approximately 70 percent of these false positives can be eliminated with little computational overhead.

Keywords: Edge Detection, Noise Removal, Background Modeling

1. INTRODUCTION

Automated video surveillance applications require the use of accurate background modeling techniques to extract salient foreground. Several techniques exist to accomplish this, varying from simple unimodal techniques, such as temporal median filtering, to the more complex multimodal Mixture of Gaussians.[1, 2] Adaptive multimodal techniques are essential to modeling busy, cluttered, or outdoor scenes, due to conditions such as repetitive object motion, shadows, or reflectance.[3]

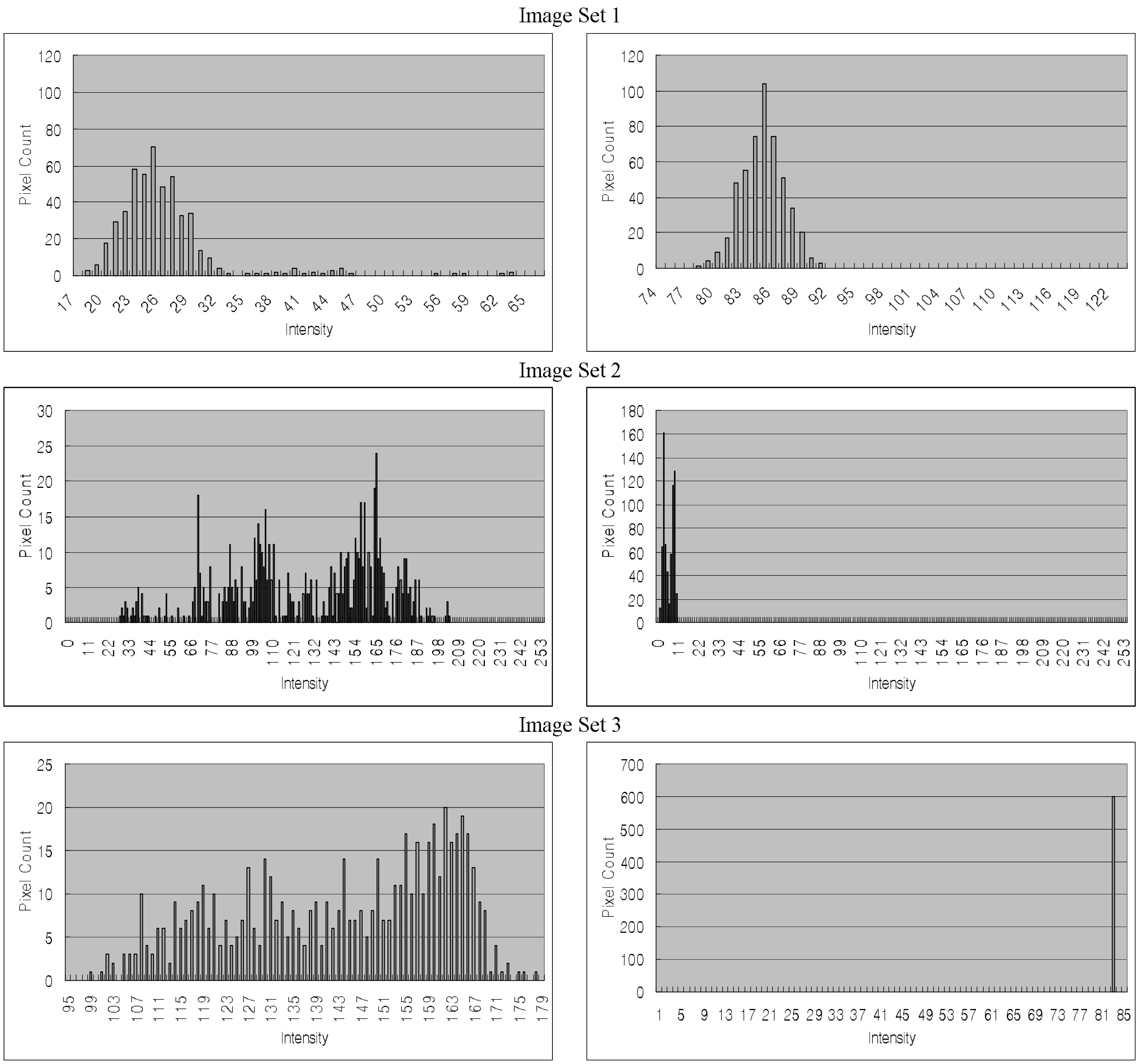
A key benefit of multimodal background models is their ability to store multiple representations of the background on a pixel-level basis. Therefore new pixels that appear in the image have to be observed consistently over time before they are considered background. For typical background pixels that lie in the interior regions of the background object, such as a large building, this is not a problem, as they do not move and their pixel value variance is minimal. However, pixels that lie on the boundary between two objects of different colors have greater variance. Examples of such pixels are those lying in the boundary between a building and the sky, or those lying in the boundary between differently colored parts of the same building, as shown in Figure 1. The intensity values of these "boundary pixels" tend to vary more than the pixels that lie in a uniformly colored part of the same object.

(a) (b)

Figure 1. An example of boundary pixels: (a) original image, (b) foreground segmented binary image.

Profiling the intensity values of 3x3 blocks of edge pixels and their neighboring blocks of non-edge pixels over time reveals that the pixel values over edges have a standard deviation that ranges from two to sixteen times more than those of internal stationary background pixels. The profiled images have been chosen at random from three different image sequences, including the Performance Evaluation of Tracking and Surveillance (PETS) 2000 image set. The distributions of the median intensity values of edge pixels and non-edge pixels taken from the profiled image sequences are shown in Figure 2.

If pixel values at edges were stable, a distribution of pixel values over time at such an edge should show two narrow Gaussian-like curves, each representing a particular color at each side of the edge. However, as it can bee seen in Figure 2, the pixel values for non-edge pixels are distributed over a much wider range of intensities as compared to those of non-edge pixels.



(a) (b)

Figure 2. Distribution of Pixel Intensity: (a) edges, (b) non-edges.

These boundary pixels tend to be both unconnected pixels (individual white dots) and clustered pixels (thick white bars), as shown in Figure 1. Therefore, using common impulse noise removal algorithms, such as spatial median filtering will only remove the unconnected pixels, leaving the clustered pixels unchanged.[4] Using connected component analysis may remove both, but they will also remove parts of the foreground, causing unwanted false negatives.[5] Clustered boundary pixels may cause critical errors because they could be falsely identified as meaningful foreground objects.

This paper introduces a novel algorithm which utilizes existing edge detection methods to identify these boundary pixels and eliminate them. Edge detection is performed on both the current image and its representative background. Edge information from both the image and the background are used to identify boundary pixels and eliminate them without causing pixels in the real foreground to be removed.

2. RELATED WORK

A variety of background modeling techniques exists, including unimodal techniques such as frame differencing, temporal median filters, and linear predictive filters, and multimodal techniques, such as Mixture of Gaussians, and multimodal mean (see [1, 9, 10] for comprehensive surveys). Frame Differencing, one of the simplest background modeling techniques, uses the frame at t-1 as the background model for the frame at time t, and subtracts the two frames to obtain the foreground. However, this type of background modeling suffers from the foreground aperture problem and does not handle multimodal dynamic background features.[1, 12] Temporal median filters, one of the most commonly used background modeling techniques,[4, 11] approximates the background at each pixel location by taking the median value from a set of pixels at that pixel location in a temporal window. Linear Predictive Filters, such as Wallflower,[12] applies a linear predictive filter whose coefficients are obtained from the sample covariances on the pixels in the buffer. When dealing with outdoor and cluttered, busy indoor scenes, unimodal techniques are inadequate since they are less capable of handling repetitive object motion, shadows or reflectance, where multiple colors can be observed at a certain location.[3] In this case, using a multimodal technique will vastly improve the modeled background.

One of the most popular multimodal background modeling techniques is Mixture of Gaussians.[2] It maintains multiple Gaussian density functions for each pixel, and matches the pixel to the background if its value is close to the mean of a Gaussian function. The pixel is part of the foreground if it does not match any of the Gaussian functions.

Another more recently developed multimodal background modeling technique is the Multimodal Mean.[6] This technique models each background pixel as a set of possible average pixel values and each pixel in the frame is compared to each of the background pixel means to determine whether it is close to any one of them. This technique achieves accuracy that is comparable to Mixture of Gaussians, but improves upon execution time and reduces storage requirements.

Noise in the form of false positive dots in the segmented foreground can be treated as impulse noise and removed using a simple median filter.[4] However, removing larger clusters of false positives is more challenging. An algorithm such as connected components analysis is capable of isolating clusters of pixels of certain area.[5] However, they cannot differentiate between clusters of false positive pixels and real foreground pixels and could potentially remove useful foreground information. More information is needed upon which to decide which pixels are parts of the foreground.

Edge detection algorithms are widely used in image processing applications, typically in tasks such as face recognition or object classification, where the structural information of the image is important. Various types of edge detectors exist, each detecting different types of edges. The simplest is based on the derivative of adjacent pixel values; the difference in neighboring pixel values indicates an edge.[8] Examples are the Prewitt mask and Sobel mask. More complicated edge detectors, such as Marr-Hildreth, are based on mathematical models which define performance criteria.[7] Edge detection algorithms efficiently capture structural information of objects in images.

3. BOUNDARY PIXEL DETECTION

The proposed algorithm applies a simple edge detection technique on the background model and intersects the result with the segmented foreground. It is assumed that any part of the segmented foreground that overlaps with the detected edges of the background is unlikely to be true foreground, and the pixels that lie in those regions are therefore removed to give the final segmented foreground.

3.1 Background Edge Detection

Edge detection based on the first derivative[7] is performed on a given background model. Edge detection is done on a pixel by pixel basis where the intensity value of each pixel, computed by taking the average of the three Red, Green, and Blue components, is compared to those of the four North, East, West, and South (NEWS) neighbors. A pixel in the background image is considered to be an *edge pixel* if the difference between the intensity value of the pixel and any one of its neighbors exceeds a certain threshold. A threshold of 20 was used to distinguish between edge and non-edge pixels in the experiments used for this paper. The value of the threshold determines how many edge pixels are detected – a larger threshold yields fewer detected pixel edges. The value was chosen based on experimental results. However, varying the threshold around the value of 20 does not greatly affect the result as a pixel is eliminated if and only if it is an edge pixel in both the background and the segmented foreground.

This creates a binary image which contains all the edges that are present in the background. This also represents a high percentage of the set of pixels that could potentially become boundary pixels that will cause noise to appear in the segmented foreground image.

3.2 Segmented Foreground Edge Detection

Performing edge detection on the current background creates a set of potential boundary pixels. However, the background edge image alone does not contain enough information to identify and eliminate noise in the segmented foreground. A boundary pixel p is considered noise that can be eliminated from the foreground if the following additional constraints are satisfied. First, p is also an element of the segmented foreground. Second, p must also be an edge pixel in the current image frame. This constraint is needed because if an edge in the background is temporarily occluded by a foreground object, those edge pixels may correspond to pixels that should be in the foreground (i.e., they are not edge noise that must be eliminated). Only pixels that are part of edges in both the background and foreground can be eliminated safely from the segmented foreground.

In certain cases, real foreground pixels may be detected as boundary pixels, causing false negatives to occur. This most commonly occurs when the background contains very small objects with complicated patterns, resulting in a high density of edge pixels. When such background objects are occluded by small and similarly patterned foreground objects, those foreground objects may be falsely detected as boundary pixels and will subsequently be removed. However, such occurrences are rare.

3.3 Computational Requirements

A significant amount of computation is spent performing edge detection on the constructed background. However, by definition, the background does not change frequently in relation to the foreground. Changes in the background model are typically brought about by structural changes, where new objects are inserted or removed permanently from the scene. Background changes can also arise from environmental conditions such as significant lighting changes. However, lighting changes do not alter the edges of the background, except with abrupt lighting changes, where objects become invisible due to darkness. Therefore, the only times when edge detection needs to be performed again is usually when new objects are introduced or removed. The frequency with which edge detection is performed on the background is purely application dependent, and in many cases this can be very infrequent, on the scale of once every few hours. In every data sequence on which this algorithm was tested, there were no significant changes in the background during the entire duration of the sequence, which ranged from a minute to over an hour. However, in all of our experiments, edge detection was performed on the background once every 20 seconds to account for possible worse case scenarios.

4. EXPERIMENTAL RESULTS

The algorithm was tested on 3 different image sequences, using both multimodal mean (MM) and Mixture of Gaussians (MOG) background modeling. The first image sequence is from the PETS 2000 dataset and it contains a building and parked cars, with moving cars and people walking by as the main sources of foreground. The second image sequence contains a pathway with people walking horizontally across the frame. The third sequence is similar to the second but contains more multimodal background objects. The frame rates, frame size and the number of frames in each sequence are shown in Figure 3. The results of running multimodal mean and MOG on these sequences and the images resulting from running the boundary pixel detection algorithm on them are shown in Figure 4 and Figure 5, along with the ground truth. A single image from each sequence that was representative of the problem was chosen. The ground truth for each frame was created by hand.

Set #	Source	Frame Rate (FPS)	Frame Size	# of Frames
1	PETS 2000	25	768x576	1452
2	Private	1	640x480	607
3	Private	1	640x480	690

Figure 3. Table of information for the sequences used for experiments.

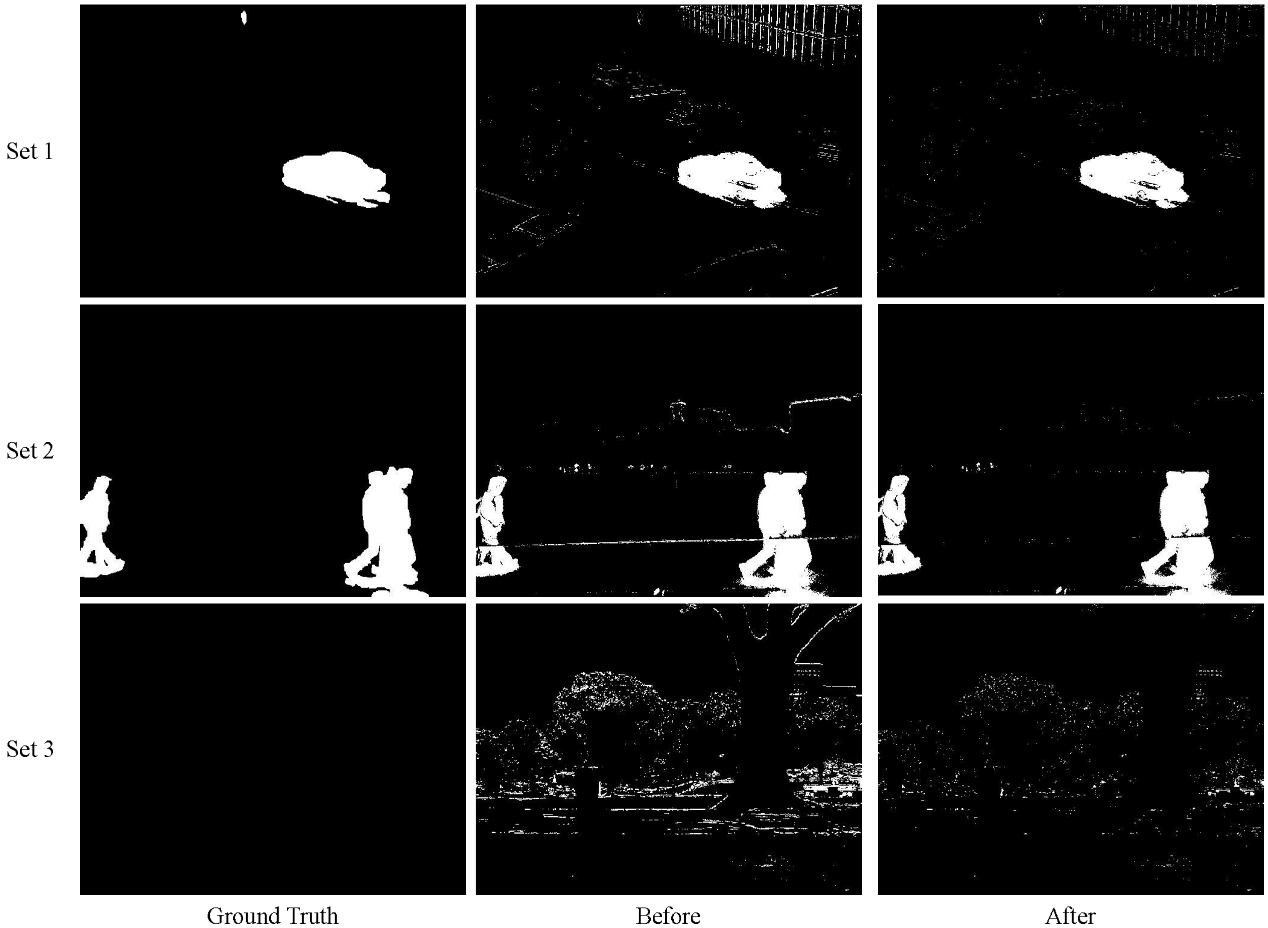


Figure 4. Comparison of Multimodal segmented foreground before and after boundary pixel detection.

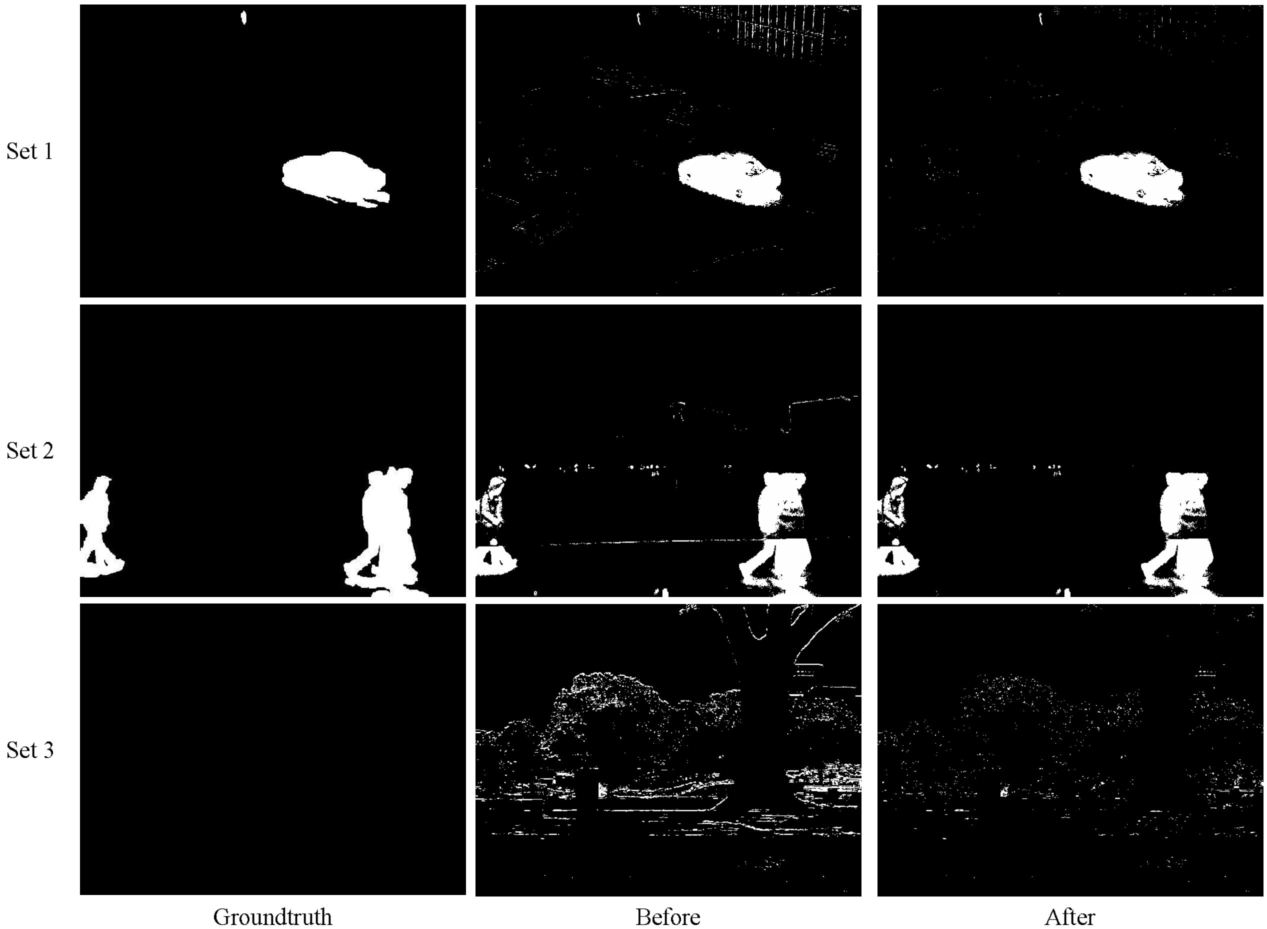


Figure 5. Comparison of Multimodal segmented foreground before and after boundary pixel detection.

As it can be seen in the above images, performing boundary pixel detection produces a much cleaner foreground segmented image without creating too much false negatives. Most of the remaining boundary pixels in the final image are the result of impulse noise and can be removed using a simple median filter.

Figure 6 shows a quantitative comparison of the results in terms of number of false positives and false negatives after the boundary pixel elimination algorithm has been applied. From 49% to 79% of the total false positives in the image can be removed. The elimination rate is higher when there are more false positive boundary pixels in the frame, and lower when there are fewer. There is a minimal increase in false negatives due to real foreground removal, ranging from 0.1% to at most 7%, with a median of 0.4%.

Set #		Multimodal Mean		Mixture of Gaussians	
		False Positive	False Negative	False Positive	False Negative
1	Before	6098	1073	3950	1095
	After	1414	1119	1906	1147
2	Before	2523	2109	1352	6176
	After	1078	2125	689	6187
3	Before	10984	0	11484	0
	After	2442	0	2419	0

Figure 6. A numerical comparison of results for the sample images.

5. CONCLUSIONS

This paper proposed a novel algorithm for detecting and eliminating noise that occurs at color boundaries when using multimodal background modeling techniques such as multimodal mean and MOG by using overlapping edge information from both the current input frame and the modeled background. By composing this algorithm with MM or MOG, 49% to 79% of false positive foreground pixels can be eliminated.

Future research on this algorithm will explore using a decision-based algorithm for determining whether to remove what is considered to be a boundary pixel to reduce false negatives. The algorithm can also be made to dynamically determine how often the background is changed, to vary the frequency at which edge detection is performed on the background.

REFERENCES

[1] Cheung, S. and Kamath, C., "Robust techniques for background subtraction in urban traffic video," Video Communications and Image Processing 5308, 881-892 (2004).

[2] Stauffer, C. and Grimson, W.E.L., "Adaptive background mixture models for real-time tracking," Computer Vision and Pattern Recognition, 246-252 (1999).

[3] Yilmaz, A., Javed, O., Shah, M., "Object Tracking: A Survey", ACM Journal of Computing Surveys 38(4), Article 13 (2006)

[4] Tukey, J.W., [Exploratory Data Analysis], Addison-Wesley Pub. Co., Reading, Mass., 205-236 (1977) .

[5] Simard, P.Y., Malvar, H.S., "An efficient binary image activity detector based on connected components," IEEE International Conference on Acoustic, Speech, and Signal Processing 3, iii - 229-32 (2004).

[6] Apewokin, S., Valentine, B., Wills, S., Wills, L., and Gentile, A., "Multimodal Mean Adaptive Backgrounding for Embedded Real-Time Video Surveillance," Embedded Computer Vision Workshop 2007, 1-6 (2007).

[7] Ziou, D., Tabbone, S., "Edge Detection Techniques - An Overview," International Journal of Pattern Recognition and Image Analysis 8(4), 537-559 (1998).

[8] Parker, J. R. [Algorithms for Image Processing and Computer Vision], Wiley, Hoboken, NJ, 1-62 (1996).

[9] Radke, R.J., Andra, S., Al-Kofahi, O., Roysam, B., "Image change detection algorithms: A systemic survey," IEEE Trans. on Image Processing, 14(3), 294-307 (2005).

[10] Piccardi, M., "Background subtraction techniques: a review," IEEE International Conference on Systems, Man and Cybernetics 4, 3099-3104 (2004).

[11] Cucchiara, R., Piccardi, M., and Prati, A., "Detecting Moving Objects, Ghosts, and Shadows in Video Streams," IEEE Transactions on Pattern Analysis and Machine Intelligence 25(10), 1-6 (2003).

[12] Toyama, K., Krumm, J., Brummitt, B., and Meyers, B., "Wallflower: Principles and Practices of Background Maintenance," in Proc. of ICCV 1, 255-261 (1999).