



**HAL**  
open science

# Automatic Background Generation from a Sequence of Images Based on Robust Mode Estimation

Dro Desire Sidibe, Olivier Strauss, William Puech

► **To cite this version:**

Dro Desire Sidibe, Olivier Strauss, William Puech. Automatic Background Generation from a Sequence of Images Based on Robust Mode Estimation. *Electronic Imaging*, Jan 2009, San Jose, United States. pp.72500P, 10.1117/12.810157 . ujm-00357228

**HAL Id: ujm-00357228**

**<https://ujm.hal.science/ujm-00357228>**

Submitted on 29 Jan 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Automatic Background Generation from a Sequence of Images Based on Robust Mode Estimation

D. Sidibé<sup>a</sup>, O. Strauss<sup>a</sup> and W. Puech<sup>ab</sup>

<sup>a</sup>LIRMM, UMR 5506 CNRS, University of Montpellier II  
161, rue Ada, 34392 MONTPELLIER CEDEX 05, FRANCE

<sup>b</sup>University of NIMES, Place G. Péri, 30021 NIMES CEDEX 1, FRANCE

## ABSTRACT

In this paper, we present a novel method for generating a background model from a sequence of images with moving objects. Our approach is based on non-parametric statistics and robust mode estimation using quasi-continuous histograms (QCH) framework. The proposed method allows the generation of very clear backgrounds without blur effect, and is robust to noise and small variations. Experimental results from real sequences demonstrate the effectiveness of our method.

**Keywords:** Background generation; Background subtraction; Mode detection; Quasi-continuous histograms.

## 1. INTRODUCTION

Detecting and tracking moving objects is a basic and important task in many computer vision and video analysis applications such as video surveillance, object tracking, pattern recognition and activity analysis. In many surveillance and tracking applications, a background image without moving objects is needed to be used as reference information for further analysis. But most of the time, it is not possible to obtain a sequence of images without moving objects, for example in traffic monitoring. Moreover, one of the most adopted approach for moving objects detection is background subtraction. The principle of background subtraction is to compare each new frame of an image sequence to a reference background model of the scene. The subtraction identifies moving or new objects in the scene. So the critical step of background subtraction method is how to build and maintain the background model of a scene.

The problem of automatically recovering the static background of a scene has received lot of attention in the computer vision community. The basic approach considers the background as the average or the median of the previous frames in the sequence.<sup>1</sup> Taking the median value provides better results than the average. However, such methods fail to accurately reflect the real background of the scene and do not correctly adapt to changes in the scene. Other commonly used background modeling methods can be divided into two groups: parametric and non-parametric methods. Methods of the first group assume that the series of intensity values of a pixel follow a known distribution whose parameters have to be estimated. Usually, the underlying probability density function is supposed to be Gaussian or a mixture of Gaussians.<sup>2,3</sup> On the contrary, non-parametric methods do not need to specify the underlying model and estimate its parameters explicitly. So they can adapt to arbitrary unknown data distribution. Elgammal *et al.*<sup>4</sup> propose a non-parametric background model based on kernel density estimation. This method gives good results and can handle situation where the background of the scene is cluttered and not completely static. In more recent works, Kim *et al.*<sup>5</sup> propose to quantize the background values into codebooks which represent a compressed form of background model for a long image sequence. This efficient representation allows a real-time segmentation. Liu *et al.*<sup>6</sup> propose a non-parametric method for background generation using the mean shift algorithm to detect the most reliable background mode (MRBM).

This paper presents a method belonging to the non-parametric category and shares similar ideas with Liu's work.<sup>6</sup> However, the main difference is the mode detection procedure. We employ a robust mode detection method using quasi-continuous histograms (QCH) introduced by Comby and Strauss<sup>7</sup> and by Strauss *et al.*<sup>8</sup>

---

desire.sidibe@lirmm.fr, olivier.strauss@lirmm.fr, william.puech@lirmm.fr

Therefore, we avoid kernel density estimation which is somewhat a time consuming operation. In addition, QCH provide a precise localization of the modes.

The rest of the paper is organized as follows. In Section 2, we describe our background generation method. Section 3 shows experimental results from real sequences, and Section 4 contains a conclusion and discussion of future work.

## 2. PROPOSED METHOD

We consider the case of a scene taken by a stationary camera. If there is no moving objects in this scene and no variations in the illumination, then for each pixel location, the intensity values along the temporal axis should be constant. In practice, different effects, such as moving objects or system noise, cause a pixel's intensity to vary from a constant value.

Fig. 1 shows the intensity variation of a pixel location (the center of the dark circle) over time. As we can see, the intensity values remain stable over time with modest variations due to system noise and significant perturbations caused by moving objects. The small variations occur most of the time, while important deviations occur in short periods of time, only when a moving object passes the location.

As a consequence of this analysis, we can model the observed values of a scene at a pixel location as the sum of an ideal background scene and different effect components:

$$V_{obsv} = B + T_{effects}, \quad (1)$$

where  $V_{obsv}$  represents the observation values at the pixel location,  $B$  the ideal background constant value, and  $T_{effect}$  the effect components such as system noise introduced by sensors, changes introduced by a moving object or changes caused by moving background regions and long term illumination changes.

We want to find an estimate  $\hat{B}$  of the ideal background  $B$  from the observed values  $\{V_{obsv}^t\}_{t=1\dots N}$ , where  $N$  is the length of the sequence. Since the moving objects appear only in a small number of images at any pixel location, an estimate of the background can be obtain as the main mode of the underlying distribution. Here, we use the so-called quasi-continuous histograms (QCH) method to estimate the modes of the distribution.

### 2.1. Quasi-continuous histograms and background generation

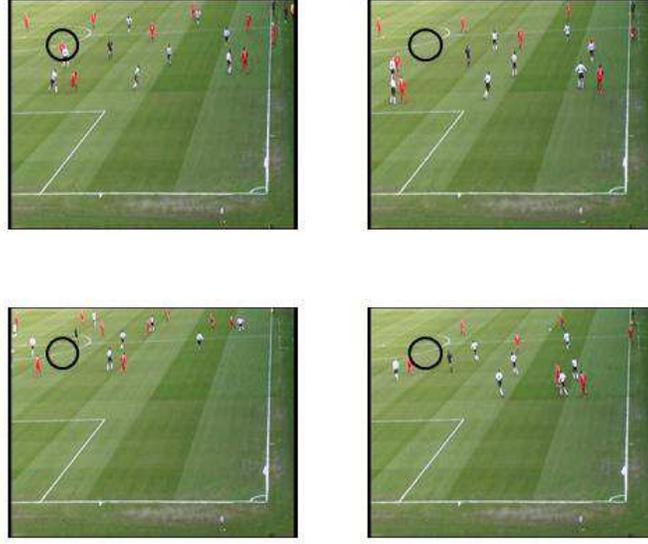
For each pixel location, we detect the background as the main mode of the distribution underlying the set of observation values  $\{V_{obsv}^t\}_{t=1\dots N}$ . To achieve this main mode estimation, we use the quasi-continuous histograms (QCH) framework.<sup>7</sup> This method provides a robust tool for estimating distribution's parameters from noisy data.

Let  $(x_1, \dots, x_N)$  be a set of  $N$  observations. The construction of a classical histogram consists in dividing a given reference interval  $[e_{min}, e_{max}]$  into  $p$  cells (or bins)  $C_k$  and counting the number  $a_k$  of observations that belong to each cell  $C_k$ .  $a_k$  is the accumulator associated with  $C_k$  and the common width  $\Delta$  of the cells is referred to as the granularity of the histogram. The major drawbacks of classical histograms are related to the crisp partition used. To get an accurate estimation of the probability density function of the distribution underlying the set of observations, the bin width  $\Delta$  should be the lower. But, because of the finite number of observations, the lower will be also the value of the accumulator associated with each cell and therefore the least is the convergence of the estimated function toward the true density.

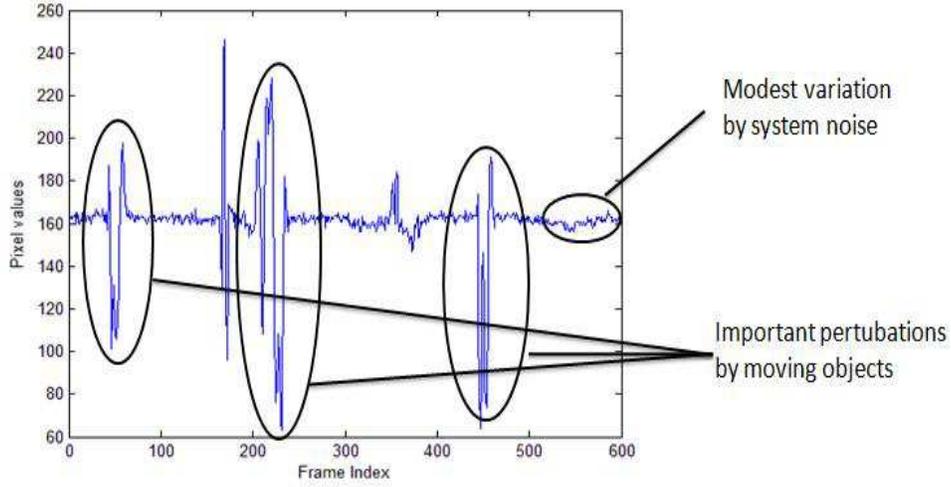
The main goal of QCH is to dissociate the histogram's granularity from the precision of the information computed using the histogram. Thus, QCH are built on a fuzzy partition of the real line instead of a crisp one as in the case of classical histograms. Replacing the classical crisp partition by a fuzzy partition leads to the distribution of the vote of each datum to two contiguous cells of the partition. This vote's distribution limits the influence of the class boundaries. Fig. 2.a. shows an example of a fuzzy partition of the interval  $[e_{min}, e_{max}]$ , and Fig. 2.b shows an example of vote's distribution in a QCH.

The generalization of the classical accumulation process can then be expressed by:

$$Acc_k = \sum_{i=1}^N \mu_{h_k}, \quad (2)$$



(a)



(b)

**Figure 1.** Intensity variation over time: a) Frames #1, #100, #210 and #420 of a sequence of 600 frames, b) Pixel intensity variation at the center of the dark circle.

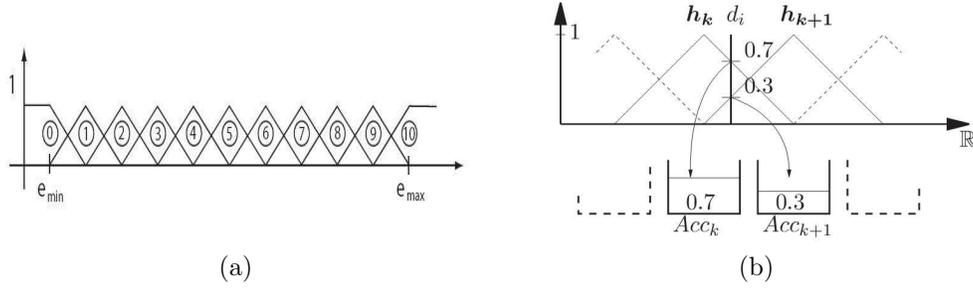
where  $Acc_k$  is the accumulator of cell  $C_k$ , and  $\mu_{h_k}$  is the membership function of  $C_k$  defined as:

$$\mu_{h_k}(x) = \begin{cases} 0 & \text{if } |x - m_k| \geq \Delta/2 \\ 1 - \frac{2|x - m_k|}{\Delta} & \text{otherwise} \end{cases} \quad (3)$$

where  $m_k$  is the central position of  $C_k$  and  $\Delta$  is the width of  $C_k$ .

The probability of an event can be estimated by the ratio between the number of observations in favor of this event and the total number of observations. Then the density of any  $x$  belonging to  $C_k$  is given by:

$$P(C_k; x) = \frac{Nb(C_k; x_i)}{N}, \quad (4)$$



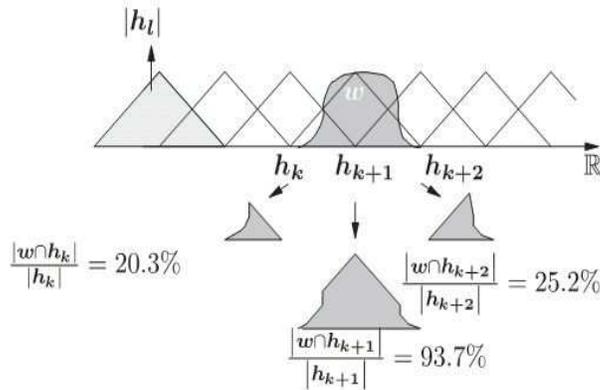
**Figure 2.** Quasi-continuous histograms: a) An example of fuzzy partition of the interval  $[e_{min}, e_{max}]$  into 11 cells, b) An example of vote in a QCH: the observation  $d_i$  votes for the two cells  $h_k$  and  $h_{k+1}$ .

where  $Nb(C_k; x_i)$  is the number of observation data  $x_i$  in  $C_k$ .

This estimation method leads to a discontinuous density. A continuous estimation of the probability density function can be obtained using QCH. This estimation is based on transferring the votes polling each cell of the partition to any kernel intersecting the partition, regardless the sample nature of the histogram.<sup>7</sup> Let consider a subset  $\mathbf{w}$  of  $\Omega$ , with  $\Omega = \bigcup_k C_k$ . Then, the number of observation data  $x_i$  in  $\mathbf{w}$  is given by:

$$Nb(\mathbf{w}; (x_i)) = \sum_{k=0}^p Acc_k \frac{|\mathbf{w} \cap C_k|}{|C_k|}, \quad (5)$$

where  $p$  is the total number of cells in the partition and  $|\mathbf{x}|$  is the cardinal of  $\mathbf{x}$ 's membership function. This principle of vote transfer toward any subset  $\mathbf{w}$  is illustrated in Fig. 3.



**Figure 3.** Quasi-continuous histograms: illustration of vote transfer.

Deriving this vote transfer allows the localization of the modes of the probability density function underlying the local distribution. For more details, please refer to Comby and Strauss.<sup>7</sup>

Once the modes are detected, we generate the background scene by keeping for each pixel, the main mode of the local intensity distribution. Notice that the distribution may have several modes, but we keep only the one which maximizes the density function underlying the observations. It is worth noticing that the mode localization's precision as in influence on the quality of the generated background. In a classical histogram, this precision depends on the width of the histogram's bin. On the contrary, using quasi-continuous histograms

entails a kind of natural interpolation between the discrete values of the partition and improves the localization's precision.<sup>8</sup> We can thus, generate a clear background regardless the granularity of the histogram as illustrated by the results in Section 3.

## 2.2. Moving objects detection

Once the background model has been generated, it can be used to detect moving objects in the scene using the background subtraction method. Unlike probability-based approaches<sup>2-4</sup> which use costly floating point operations, we simply compute the difference between the current frame and the background model. Detection is thus straightforward and very fast.

## 3. EXPERIMENTAL RESULTS

We used several real image sequences to test the performance of our background generation method. The method is implemented and tested on a 1.7 GHz Pentium system. In all experiments, the images are of size 320x240 pixels. In the case of color images, the background generation method is applied in each color component separately.

### 3.1. Background generation

We tested our method's capability to generate a clear background from a sequence of images containing multiple moving objects. Fig. 4 shows some examples of generated backgrounds. The *Laboratory Hall* sequence, Fig. 4.a, was acquired in our laboratory, while the two other sequences, Fig. 4.b and Fig. 4.c, are selected from the PETS databases.<sup>9,10</sup> The total length of the sequences is 600 frames. For all sequences, we use the first 100 frames to generate the background. As it can be seen, the proposed method can generate a very clear background from a sequence with cluttered moving objects. The *Soccer Game* sequence for example, shown in Fig. 4.b, is taken from a video which always shows more than twelve moving players in each frame. From the last column of Fig. 4.b we can see that the generated background is very clear and all the moving objects have been successfully removed.

The only parameter that needed to be specified in the experiments is the number  $N$  of observations, i.e. the number of frames for training the background model. Theoretically, the larger the training set is, the better the background model is. But at the same time, if  $N$  is too large then the ability to adapt to changes will be very low. To measure the influence of this parameter, we use a sequence of frames for which a ground truth background is available. We then vary the number of training frames used to generate the background. Fig. 5 shows results obtained for different numbers of training frames. As it can be seen, with a small number of training frames, there are some noise points in the generated background, though the overall visual quality of the background is still acceptable. When the number of training frames is increased, we obtain a very clear background. In the example shown in Fig. 5, the noise points are eliminated from 80 to 90 training frames.

For a quantitative analysis, let define an error measure for a background generation method. Let  $B^*$  be a known ground truth background and  $B$  an estimated background image. We compute the absolute difference image  $D$  between  $B^*$  and  $B$  by:

$$\forall i, j \ D(i, j) = |B^*(i, j) - B(i, j)| \quad (6)$$

The error of the background estimation is then defined as:

$$\epsilon = \frac{\sum_{i=1}^N \sum_{j=1}^M D(i, j)}{N * M}, \quad (7)$$

where  $[N, M]$  is the size of the sequence's frames.

The smaller  $\epsilon$  is, the better the background estimation is. Fig. 6 shows the sensitivity of the method as a function of the number of training frames. We can see that the method is quite stable for a number of training frames greater than 80. Of course, for very few training frames, less than 40 frames, the estimated background shows important differences with the true background.

The implementation is quite fast. On average, it takes 25 s to generate a background from 100 training frames on our 1.7 GHz Pentium system. This includes the time needed for reading and storing the samples.

### 3.2. Background subtraction

Moving objects are detected using background subtraction method by computing the difference between the current frame and the generated background model. Fig. 7 shows examples of detection results using background



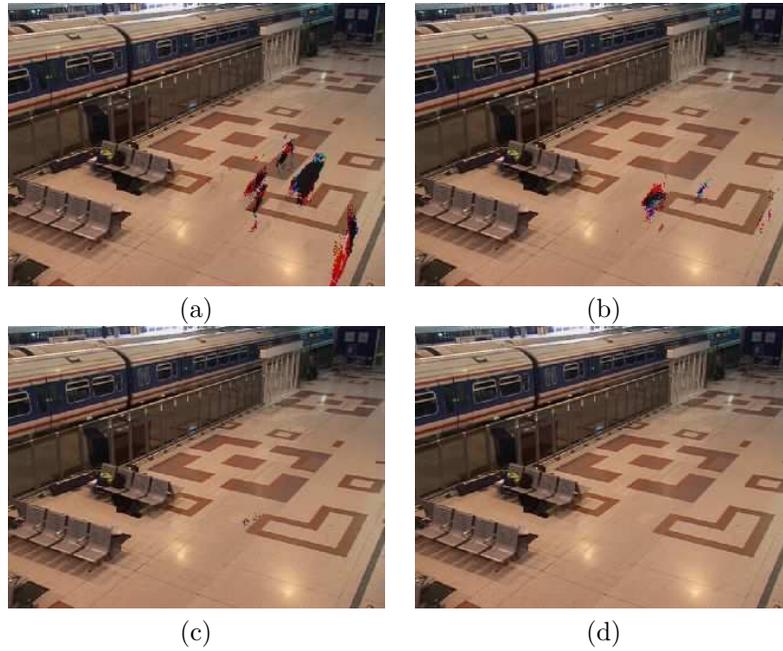
(a) Laboratory Hall

(b) Railway Station

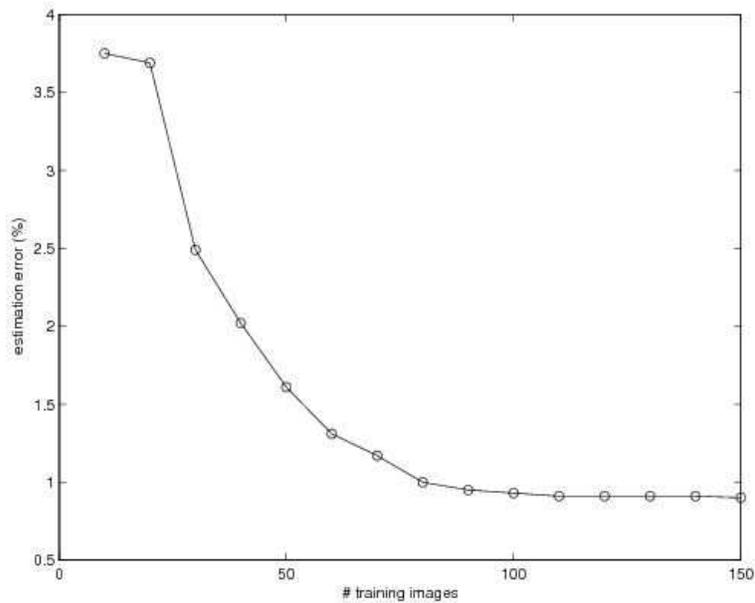
(c) Soccer Game

**Figure 4.** Background images generated by our method. The first three rows show frames #1, #20 and #50 of the sequences. The last row shows the generated background.

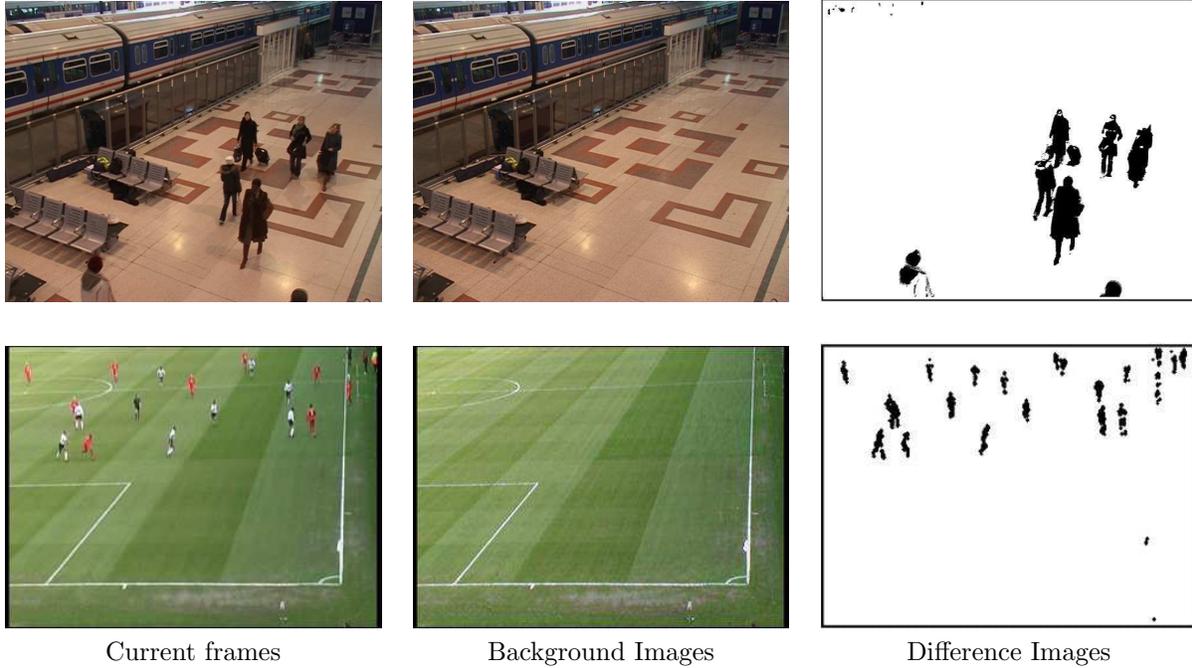
subtraction. We perform only background subtraction follows by an automatic threshold selection to obtain a binarized image. We used Otsu's<sup>11</sup> method for threshold selection. Because the generated background is very clear, moving objects are correctly segmented. The detection step is straightforward and fast, about 0.015 s per image in our experiments which allows a rate of 66 frames/s (fps).



**Figure 5.** Generated backgrounds for different numbers of training images: (a) With 40 frames, (b) With 60 frames, (c) With 80 frames, (d) With 100 frames.



**Figure 6.** Estimation error variation as a function of the numbers of training frames.



**Figure 7.** Background subtraction results.

#### 4. CONCLUSION

This paper introduces a new method for generating a background image from an image sequence with moving objects. Based on a robust mode estimation method, which employs the quasi-continuous histograms (QCH) framework, we are able to generate very clear backgrounds from a sequence with cluttered moving objects. This background model is then used for moving objects segmentation using background subtraction.

Nevertheless, some aspects should be considered to improve the robustness of the approach. One important point is how to deal with the case where the background is not static, for example in outdoor scenes with moving trees. In these cases, a multimodal approach is needed and the QCH framework could be used since it allows to detect all modes of a distribution. Considering multi-backgrounds is part of our future work. Another direction of future work could be extending the current pixel-based method to a region-based detection method, through incorporation of spatial or neighborhood information.

#### REFERENCES

1. R. Cucchiara, C. Grana, and M. Piccardi, "Detecting Moving Objects, Ghost and Shadows in Video Streams," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **25**(10), pp. 1337–1342, 2003.
2. C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland, "Pfinder: Real-time Tracking of the Human Body," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **19**(7), pp. 780–785, 1997.
3. C. Stauffer and W. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246–252, 1999.
4. A. Elgammal, D. Harwood, and L. Davis, "Non-parametric Model for Background Subtraction," in *European Conference on Computer Vision, Dublin, Ireland*, pp. 751–767, 2000.
5. K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time Foreground-Background Segmentation Using Codebook," *Real Time Imaging* **11**(3), pp. 172–185, 2005.
6. Y. Liu, H. Yao, W. Gao, X. Chen, and D. Zhao, "Nonparametric Background Generation," *Journal of Visual Communications and Image Representation* **18**(3), pp. 253–263, 2007.

7. F. Comby and O. Strauss, "Using Quasi-Continuous Histograms for Fuzzy Main Motion Estimation in Video Sequence," *Fuzzy Sets and Systems* **158**, pp. 475–495, 2007.
8. O. Strauss, F. Comby, and M. Aldon, "Rough Histograms for Robust Statistics," in *Proceedings of the 15th International Conference on Pattern Recognition*, pp. 684–687, 2000.
9. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2003. Available from '<ftp://ftp.pets.rdg.ac.uk/pub/VS-PETS>'.
10. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2006. Available from '<http://pets2006.net/data.html>'.
11. N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. on System, Man and Cybernetics* **9**(1), pp. 62–62, 1979.